ECHELON FORMS AND REFORMULATIONS OF PATHOLOGICAL SEMIDEFINITE PROGRAMS

Aleksandr I. Touzov

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
2022

Approved by:

Quoc Tran-Dinh

Gábor Pataki

Sayan Banerjee

Jan Hannig

Greg Blekherman

ABSTRACT

ALEKSANDR TOUZOV: Echelon forms and reformulations of pathological semidefinite programs
(Under the direction of Gábor Pataki)

This thesis seeks to better understand when and why various well-known pathologies arise in semidefinite programs (SDP). The first part of this thesis is concerned with the pathology of weak infeasibility. Unlike in linear programs, Farka's lemma may fail to identify infeasible SDPs. This pathology occurs precisely when an SDP has no feasible solution, but it has nearly feasible solutions that approximate the constraint set to arbitrary precision. These SDPs are ill-posed and numerically often unsolvable. They are also closely related to "bad" linear projections that map the cone of positive semidefinite matrices to a nonclosed set. We describe a simple echelon form of weakly infeasible SDPs with the following properties: it is obtained by elementary row operations and congruence transformations; it makes weak infeasibility evident; and using it we can construct any weakly infeasible SDP or bad linear projection by an elementary combinatorial algorithm. We also prove that some SDPs in the literature are in our echelon form, for example, the SDP from the sum-of-squares relaxation of minimizing the famous Motzkin polynomial.

The second part of this thesis deals with the pathology of exponentially sized solutions in SDPs. As a classic example of Khachiyan shows, some SDPs have solutions whose size - the number of bits necessary to describe it - is exponential in the size of the input. Although the common consent seems that large solutions in SDPs are rare, we prove that they are actually quite common: a linear change of variables transforms every strictly feasible SDP into a Khachiyan type SDP, in which the leading variables are large. As to how large, that depends on the singularity degree of a dual problem. We further show some SDPs in which large solutions appear naturally, without any change of variables. Along the way, we draw connections to continued fractions, Fourier-Motzkin elimination, and give positive progress to answering the long-standing open question: can we decide feasibility of SDPs in polynomial time?

To my beloved family

ACKNOWLEDGEMENTS

Beyond anything, I would like to thank the support and guidance of my advisor Gábor Pataki in the completion of this work and my time at UNC Chapel Hill. I have grown to learn a lot from him. Academically - Professor Pataki exemplifies everything that makes a great teacher great: a genuine care to see the growth and development of his students, an innate ability to break down and present complicated information in an easy to digest form, and, most importantly, a patience and perseverance that makes learning from him an opportunity and a privilege. Personally - he demonstrates that nothing is accomplished without hard work and and a steadfast resolve; it is often easy to let a challenging undertaking go by the wayside from fear of failure, but through example, Professor Pataki has shown on numerous occasions that we do indeed reap the rewards of our hard work. He is a great person that has made my experience at Chapel Hill a positive one, and I will surely miss working with him after my time here is done.

I would also like to thank the kind and supportive faculty and staff of our department. Prior to coming here, I have always heard that graduate school was a stressful undertaking that would take a psychological toll on anyone that went through it. After working with and learning from everyone, I must disagree. This program has been illuminating as much as it has been enjoyable to come to the department each day. The atmosphere created by the department and everyone working in it is one of a team, that I will always consider a pleasure to have been on.

Last, but absolutely not the least, I would like to thank my loving parents, sister, and my love Emily. You have always been my greatest support and those whom I could always lean on in times of need. I am lucky to have you in my life, and no matter where my road takes me in the future, or how far we may depart, you will always be by my side.

TABLE OF CONTENTS

LIST OF TABLES ... viii

LIST OF FIGURES ... ix

LIST OF ABBREVIATIONS AND SYMBOLS ... x

1 Introduction ... 1

1.1 Semidefinite programming ... 2

1.2 New contributions and techniques ... 4

1.3 Outline of dissertation ... 7

2 An echelon form of weakly infeasible semidefinite programs and bad projections of the psd cone ... 8

2.1 Introduction ... 8

2.2 Preliminaries ... 13

2.3 The main result, and the easy direction ... 16

2.4 How to construct any weakly infeasible SDP and bad projection of the psd cone ... 27

2.5 Proofs: certificates of infeasibility and not strong infeasibility separately ... 34

2.6 Proof of Theorem 1 ... 39

2.7 Our problem library and computational tests ... 51

2.8 Discussion and conclusion ... 53

3 How do exponential size solutions arise in semidefinite programming? ... 56

3.1 Introduction ... 56

3.1.1 Notation and preliminaries ... 62

3.2 Main results and proofs ... 64

3.2.1 Reformulating (P) and statement of Theorem 5 ... 64

3.2.2 Proof of Theorem 5 ... 70

3.2.3 Computing the exponents by Fourier-Motzkin elimination 85
3.3 When we do not even need a change of variables 86
3.4 Conclusion 91
BIBLIOGRAPHY 93

LIST OF TABLES

2.1 Parameters of our weakly infeasible SDPs: $k+1$ being the length of the infeasibility certificate, $\ell+1$ the length of the not-strong infeasibility certificate, n the matrix order, and m the number of constraints. 52

2.2 Number of infeasible instances correctly identified by MOSEK and SDPA-GMP 53

LIST OF FIGURES

2.1 A visualization of (ME) ... 9

2.2 The matrix of inner products of the A_i' and X_j in Theorem 1 ... 18

2.3 Proving that ($\mathrm{P}_{\mathrm{weak}}$) is infeasible ... 19

2.4 Proving that ($\mathrm{P}_{\mathrm{weak}}$) is not strongly infeasible ... 20

2.5 The A_i' and X_j obtained from reformulating the SDP (2.11) ... 22

2.6 Certificates of weak infeasibility in the Motzkin polynomial SDP ... 23

2.7 How to diagonalize $X(R_{1:t})$ in Lemma 7 ... 42

2.8 X_{j+1} before it is transformed ... 46

2.9 How step (2.45) changes X_{j+1} ... 47

2.10 How steps (*) and (**) in Algorithm 3 change X_s, where $s \leq j$. ... 48

2.11 How step (*) in Algorithm 4 changes A_i', when $q = i$ ... 49

2.12 How step (*) in Algorithm 4 changes A_i', when $q < i$ ... 50

2.13 How step (2.45) changes A_i', when $q > i$ ... 50

3.1 Feasible sets of (*Khachiyan*) (on the left) and of the quadratic inequalities (3.13) derived from (*Mild-SDP*) (on the right) ... 57

3.2 Verifying that $x_1, \ldots, x_k$ exist, without computing them ... 71

3.3 The tail-index of A_{j+1}' ... 72

3.4 Feasible sets of (*Khachiyan*) (on left) and of inequalities derived from the perturbed Khachiyan SDP (3.29) (on the right) ... 78

3.5 Shifting x_2 to the right increases α_2 ... 83

LIST OF ABBREVIATIONS AND SYMBOLS

$\mathbb{R}^n$	Set of n dimensional real valued vectors
$\mathbb{R}^n_+$	The positive orthant cone
$\mathbb{R}^n_{++}$	The interior of the positive orthant cone
$\mathcal{S}^n$	Set of $n \times n$ real symmetric matrices
$\mathcal{S}^n_+$	Cone of positive semidefinite symmetric matrices
$\mathcal{S}^n_{++}$	Cone of positive definite symmetric matrices
A^*	The adjoint of a matrix or linear operator A
$\mathcal{R}(A)$	Range space of a matrix or linear operator A
$\mathcal{N}(A)$	Null space of a matrix or linear operator A
$A \bullet B$	The inner product between matrices A and B, defined $\mathrm{Tr}(A^\top B)$
$A \oplus B$	Diagonal concatenation of square matrices A and B
$A \succeq B$	The difference $A - B$ is positive semidefinite
$A \succ B$	The difference $A - B$ is positive definite

CHAPTER 1

Introduction

Over the past several decades we have seen drastic growth in computing power for solving mathematical and real world problems. Problems which were previously ignored due to their large scope and intractability are now approachable, and moreover, frequently central to the study of many modern mathematical fields. One salient examples of such a problem is mathematical programming.

Although not a new concept, with origins dating back to Kantorovich and Dantzig in the 1940s [13], mathematical programming was largely limited in scope to linear programming (LP) by simplex method until the first polynomial-time LP algorithms discovered by Khachiyan in 1979 [24]. Since then, interior point methods for conic optimization have taken off alongside the computer revolution of the 21st century. With the advent of semidefinite programming (SDP), often referred to as linear programming for the 2000's, we are now able to solve previously impossible problems with a high degree of accuracy and precision. In particular, SDPs are now crucial, in many modern industries, for solving problems in control theory, probability theory, robust optimization, facility planning, PDEs, coding theory, statistical estimation, geometric optimization, and much more. For an in-depth survey of SDPs we refer to [64]. However, even after decades of study, SDPs are not without their own mysteries and anomalous behaviors. Despite being a rather straightforward extension of linear programming, many of the nice behaviors we take for granted in LPs, such as strong duality, polynomial time solvability, exact certificates of infeasibility, etc. fail in SDPs in what we call pathological semidefinite programs.

To understand why these problematic SDPs arise, we leverage seemingly innocuous tools from basic convex analysis and linear algebra to effectively "diagnose" these pathological instances. When trying to understand pathologies such as exponentially size SDP solutions or weakly infeasible semidefinite programs, these "diagnostics" often express themselves as a reformulation of the original problem in which the problematic behavior becomes trivial to see.

1.1 Semidefinite programming

One of the most important sub-fields of mathematical programming is that of semidefinite programming. A semidefinite program (SDP) in standard primal form is a mathematical optimization problem of the form

$$\begin{aligned} \inf \quad & B \bullet X \\ s.t. \quad & A_i \bullet X = c_i, \quad i = 1, \dots, m \\ & X \succeq 0 \end{aligned} \tag{P}$$

where $B, A_1, \dots, A_m$ are real symmetric matrices, $X \succeq 0$ means that X is symmetric and positive semidefinite, and for two symmetric matrices A, B we let $A \bullet B = \mathrm{Tr}(AB)$. If we look closely at this problem and the natural dual problem

$$\begin{aligned} \sup \quad & c^\top y \\ s.t. \quad & B - \textstyle\sum_{i=1}^m A_i y_i \succeq 0 \end{aligned} \tag{D}$$

it may not be surprising why SDPs are often considered the natural generalization of the linear program. After all, if we replace each matrix with a vector, $\bullet$ with the canonical euclidean inner product, and each instance of $\succeq$ with an element-wise $\geq$, then we precisely recover the standard primal-dual linear program.

However, it is not by accident that the SDP is expressed with "inf" and "sup" as opposed to "min" and "max". This difference highlights one of the many symptoms that pathological SDPs may exhibit, which linear programs do not. In this case, while weak duality between the above primal-dual pair always holds, strong duality and more so attainment of optimal values does not always follow.

While we often assume a constraint qualification, such as existence of an interior point (i.e. Slater's condition), to enforce strong duality when studying SDPs, this is simply avoiding the pathological behaviors rather than diagnosing and solving them. To rectify these pathological problems, two tools are often leveraged: Ramana's extended dual formulation [55] and facial reduction. Ramana's dual extends the original SDP instance with a polynomial number of constraints and variables into a new SDP which satisfies strong duality and attainment of optimal solutions analogous to the nice properties we enjoy in linear programming. Facial reduction, on the other hand, is a more general concept that can in fact be used to show correctness of Ramana's dual [56].

The idea of facial reduction is simple: if we can find the largest face of the semidefinite cone which has a nonempty intersection of its interior with the affine space of solutions $H := \{X \mid A_i \bullet X = c_i\}$ for (P), then reducing the problem to this minimal face produces a problem which automatically satisfies Slater's constraint qualification. Originally proposed by Borwein and Wolkowicz in [10] to strengthen the conditioning of conic-convex programs, facial reduction algorithms have become a powerful practical and theoretical tool for studying SDPs and other conic-convex optimization problems. For a comprehensive survey of facial reduction algorithms, where they are useful, and how they are leveraged to rectify loss of Slater's condition, we refer to [14].

It would appear then, that facial reduction algorithms are key to diagnosing and solving SDP pathologies when they arise. In fact, this idea is not a new one. Recent work by Liu and Pataki [31] used facial reduction to construct exact duals for more general conic linear programs, effectively generalizing the SDP dual of Ramana. The same work also examined pathological infeasible conic linear programs. It is well known that, beyond linear programs, both (P) and the corresponding Farka's alternative system may be infeasible; identifying these pathological infeasible problems is difficult. In their work, Liu and Pataki used the facial reduction toolbox to construct strong alternative systems for both the primal and dual conic program for the case that this "weak" infeasibility presents itself: producing a result akin to a "strong" Farka's Lemma.

Similarly, we study the pathology of weak infeasible in SDPs in more detail and explore previously unexamined bad behaviors related to the long standing open question: can we determine SDP feasibility in polynomial time? Using facial reduction and basic results from convex analysis and linear algebra, we make positive progress towards finding answers and solving these problems.

1.2 New contributions and techniques

The first type of pathology that this thesis examines is the pathology of weakly infeasible semidefinite programs. Unlike linear programs, Farka's alternative system for (P)

$$\begin{array}{rcl} \sum_{i=1}^{m} A_i x_i & \succeq & 0 \\ c^\top x & = & -1 \end{array}$$

may be infeasible, even when the primal standard form SDP is infeasible. This is problematic as the alternative system is one of the best tools for providing certificates of infeasibility for an SDP. The issue with semidefinite programs that gives rise to this pathology stems from a fundamental question in convex analysis: when is the linear image of a closed convex set closed? In particular, for SDPs, the convex set of interest is the semidefinite cone while the linear map is the operator

$$\mathcal{A} : X \mapsto (A_1 \bullet X, \ldots, A_m \bullet X)^\top$$

When the mapping of the semidefinite cone is not closed, we can find some vector $c \in \mathbb{R}^m$ for which the primal standard form is infeasible, yet there exist matrices which satisfy $\mathcal{A}(X) = c$ and are arbitrarily close to the semidefinite cone. These instances are referred to as being weakly infeasible. Geometrically, weakly infeasible SDPs arise precisely when the solution space H is an asymptote of the semidefinite cone such that the classical separating hyperplane theorem fails to hold. However, in modern literature this pathology is even more interesting. Among weak infeasibility's many guises, a few notable cases in which we see this behavior are:

1. in difficult to solve SDPs which state-of-the-art interior point solvers such as MOSEK or SDPA-GMP fail to correctly identify infeasibility, even for instances as small as $n = 3$ with MOSEK and $n = 7$ with the general precision solver SDPA-GMP which can carry out calculations with precision 10^{-200}.

2. in cases for which the operator $\mathcal{A}$ maps the semidefinite cone to a non-closed set. The space of such $\mathcal{A}$ was recently studied from the perspective of convex algebraic geometry by Jiang and Sturmfels [21] where membership to this variety can explicitly be checked under appropriate parameterization.

3. in the study of condition for the SDP feasibility problem; these problems are *ill-posed*, that is, their distance to the set of feasible instances is zero and so interior point methods whose complexity bounds depend on Renegar's distance to ill-posedness have no convergence guarantees [51].

4. in certain classes of polynomial optimization problems related closely to minimizing non-negative polynomials which are not a sum-of-squares [67].

In this regard, weak infeasibility appears to be a topic well-worth examining and understanding. In our study of these instances, we develop an echelon form for weakly infeasible SDPs: a form that any such instance can be "untangled" into using only elementary reformulations akin to those used in gaussian elimination. As a biproduct of our work, the echelon form developed for weakly infeasible SDPs also characterizes orbits of non-closed projections of the semidefinite cone.

Moreover, our work in developing this characterization answers an important question: how do we generate any weakly infeasible semidefinite program and bad projection of the semidefinite cone? While previous works by Liu and Pataki [31] and Waki [67] provide partial answers to this question, we see that our echelon form allows us to solve the problem entirely and we produce a library of weakly infeasible SDPs with instances which have unique geometric structure not present in either of the previous works.

The second pathological behavior we study is the existence of exponentially large solutions in SDPs relative to the problem's size: the number of bits necessary to encode the problem data. By a classical argument using Cramer's rule, the size of any basic feasible solution to a linear program is polynomial in the size of the problem instance. This is not the case in SDPs. A classic example by Khachiyan shows that even simple SDPs can exhibit solutions which would require exponential size relative to the problem data to encode. These exponentially sized solutions appear to be the main obstacle in answering the long standing open question: can we decide SDP feasibility in polynomial time?

It would seem that if we cannot even write down a solution in polynomial time, then what hope do we have in claiming the SDPs feasibility status is decidable as such? If we look more closely at Khachiyan's example,

$$x_1 \geq x_2^2, \quad x_2 \geq x_3^2, \quad \ldots \quad x_{k-1} \geq x_k^2, \quad x_k \geq 2$$

it is not too difficult to convince ourselves that $x_1 = 2^{2^{k-1}}$ is feasible using simple symbolic manipulation. It seems then, that writing down the solutions to an SDP may not necessarily be required to decide the problem's feasibility. This gives us hope and in fact, we show that any strictly feasible SDP can be reformulated into a Khachiyan type instance: where deciding feasibility of the problem no longer requires computing values for the k largest leading variables. As to how large is k? That depends on the singularity degree of a dual type problem. Along the way, we show that when generalizing Khachiyan's example to Khachiyan-type SDPs, the size of leading variables depends tightly on certain combinatorial properties of the SDP: properties which algebraically manifest through connections with continued fractions and Fourier-Motzkin elimination.

The assumptions of our work are minimal. In fact, the main idea that we leverage is the idea of elementary reformulation which we used in the first part of this thesis to develop an echelon form for weakly infeasible SDPs, and it appears to be necessary once again here, since even Khachiyan's simple example can be reformulated into an instance with variables which no longer need be large by replacing $x \leftarrow Gx$ for a random dense matrix G. Moreover, we show that even reformulation is unnecessary in many cases. In fact, for any unconstrained polynomial minimization problem relaxed into a sum-of-squares SDP using the standard monomial basis, we see that exponentially sized solutions appear naturally. Additionally, we see that Khachiyan's example is not as artificial as it may originally seem: [40] proposed an example of certifying polynomial non-negativity with a sum-of-squares proof system which we show is essentially equivalent to Khachiyan's original example.

We conclude this section by partially answering the question: how do we represent exponentially sized solutions to SDPs in polynomial space?

1.3 Outline of dissertation

1. In Chapter 2 we study the pathology of weak infeasibility in semidefinite programs. After reviewing preliminary material, Section 2.3 presents Theorem 1, our main result: an echelon form which makes weak infeasibility of an SDP evident with readily available certificates for SDP infeasibility and not-strong infeasibility. In this section, we prove the "easy" direction of the theorem to build some intuition and relegate the more difficult direction to a later section. Section 2.4 describes an algorithm for constructing weakly infeasible SDPs, and we show that any weakly infeasible SDP and non-closed projection of the semidefinite cone is among its outputs. The main work for developing this result is presented in Section 2.6, where we prove the difficult direction of Theorem 1. Afterwards, in Section 2.7, we describe our problem library and computational tests. We finish this chapter with Section 2.8 in which we reinterpret Theorem 1 in two ways: as a "sandwich" theorem and as a "factorization" theorem.

2. In Chapter 3 we tackle the pathology of exponentially sized solution in SDPs. To begin this section, we review some related material and outline Khachiyan's example: a motivating SDP which exhibits exponentially large solutions, yet verifying feasibility can trivially be done in polynomial time with simple symbolic algebra. We go on to present the main result, Theorem 5 in Section 3.2.1 and illustrate it via two extreme examples; Theorem 5 shows that, in fact, any strictly feasible SDP is equivalent to a Khachiyan-type SDP where existence of exponentially sized solutions is easy to verify without computing the values of large variables. In Section 3.2.2 we go on to prove Theorem 5 through a sequence of Lemmas. Along the way, we show explicit connections between this work and continued fractions as well as Fourier-Motzkin elimination for linear inequalities: an interesting contrast with the non-linearity of SDPs. In Section 3.3 we turn our attention to implications of our result to the study of polynomial optimization problems. Here, we show that exponentially large solutions appear naturally in polynomial minimization problems and present Theorem 6. We conclude this section with a discussion.

CHAPTER 2

An echelon form of weakly infeasible semidefinite programs and bad projections of the psd cone

2.1 Introduction

Semidefinite programming (SDP) feasibility problems of the form

$$\begin{aligned} \mathcal{A}X &= b \\ X &\in \mathcal{S}^n_+, \end{aligned} \tag{P}$$

are fundamental in many areas of applied mathematics, including combinatorial optimization, polynomial optimization, control theory, and machine learning. Here $\mathcal{A}$ is a linear map from $n \times n$ symmetric matrices to $\mathbb{R}^m$, b is in $\mathbb{R}^m$, and $\mathcal{S}^n_+$ is the set of symmetric positive semidefinite (psd) matrices.

Semidefinite programs – either in the feasibility form, or in an optimization form, with an objective function attached – are often pathological and this work focuses on their pathological kind of infeasibility, called *weak infeasibility.* Precisely, we say that (P) is weakly infeasible when it has no feasible solution, but the set of solutions of the linear system of equations $\mathcal{A}X = b$ has zero distance to $\mathcal{S}^n_+$.

Example 2.1. An enlightening, classical, and minimal example is

$$\begin{array}{rcl} x_{11} & = & 0 \\ x_{12} = x_{21} & = & 1 \\ X & \in & \mathcal{S}^2_+, \end{array} \tag{ME}$$

where the (i, j)th element of X is denoted by x_{ij}. If X satisfies the equality constraints of (ME), then

$$X = \begin{pmatrix} 0 & 1 \\ 1 & x_{22} \end{pmatrix}$$

for some x_{22} real number, so X cannot be positive semidefinite. Hence (ME) is infeasible. However, such X matrices converge to $\mathcal{S}^2_+$ if we choose $x_{22} > 0$ to be large: then to make X psd, we must only slightly change its 0 entry to $1/x_{22}$. So we conclude that (ME) is weakly infeasible.

We visualize this example in Figure 2.1. The solid blue set bordered by a hyperbola is

$$S = \left\{ (x_{11}, x_{22}) \in \mathbb{R}^2_+ \ : \ x_{11}x_{22} \geq 1 \right\},$$

the set of diagonals of 2×2 psd matrices whose offdiagonal elements are 1. We approach S arbitrarily closely if we fix $x_{11} = 0$ and make x_{22} large, moving towards infinity on the x_{22} axis.

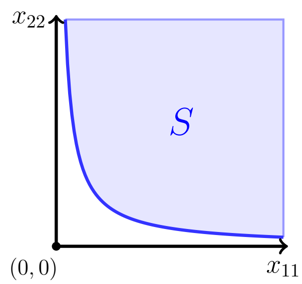


Figure 2.1: A visualization of (ME)

$\Diamond$

Weakly infeasible SDPs appear in the literature in many guises, some of which are modern and some others classic:

- they are difficult SDPs that are often mistaken for feasible ones by even the best solvers.
- they are closely related to linear maps under which the image of $\mathcal{S}^n_+$ is not closed. Precisely, (P) is weakly infeasible, if it is infeasible, but a sequence $\{X_i\} \subseteq \mathcal{S}^n_+$ satisfies

$$\mathcal{A}X_i \to b, \text{ as } i \to +\infty,$$

in other words, when b is in the closure of $\mathcal{A}\mathcal{S}_+^n$, but not in $\mathcal{A}\mathcal{S}_+^n$ itself. Such linear maps cause other pathologies in SDPs, such as unattained optimal values and positive duality gaps [49, Lemma 2]. They are also intriguing from the viewpoint of pure mathematics: they were recently christened *bad projections of the psd cone* by Jiang and Sturmfels [21] and explored from the perspective of algebraic geometry.

  More broadly, bad projections are a good example of linear maps that carry a closed set into a nonclosed one, because $\mathcal{S}_+^n$ is one of the simplest sets for which such maps even exist. The "(non)closedness of the linear image" question appears in several equivalent forms, for example, we may ask whether the sum of closed convex cones is closed. (Non)closedness of the linear image can be ensured by many diverse conditions: some of the key ones are Jameson's property (G) (Jameson [20] and Bauschke et al. [6]) and existence of certain tangent directions [46, Theorem 1.1, Theorem 5.1].

- they are *ill-posed*, i.e., their distance to the set of feasible instances is zero. Hence their infeasibility cannot be detected by interior point methods whose complexity depends on the distance to feasibility; the best we can do is compute solutions of nearby feasible instances, see Peña and Renegar [51, Theorem 13]. Nor can we detect their infeasibility by the algorithm of Nesterov et al. in [39], though this algorithm can detect *near* ill-posedness.

  For a sample of the thriving literature on algorithm analysis based on distance to (in)feasibility, we refer to Renegar [59, 60]; Peña [50]; and the comprehensive book by Bürgisser and Cucker [12].

- according to a classic viewpoint of Klee [25], when (P) is weakly infeasible, the affine subspace $\{ X : \mathcal{A}X = b \}$ is an *asymptote* of $\mathcal{S}_+^n$. The asymptotic behavior is indeed apparent on Figure 2.1. [1]

Although the infeasibility of weakly infeasible SDPs cannot be reliably detected in general, several algorithmic approaches are available:

[1] To be precise, although Klee introduced the notion of asymptotes, he did not specifically mention asymptotes of the psd cone.

- Facial reduction algorithms can handle pathological SDPs, at least in theory, since they must be implemented in exact arithmetic. Facial reduction originated in a paper by Borwein and Wolkowicz [11], then simpler variants were proposed, for example in [44, 47]. We will use facial reduction as a theoretical tool to develop our echelon form. In particular, we will use Parts (1) and (2) of Theorem 5 in [31]; an earlier version of the latter appeared in Lourenço et al. [34].

- In contrast to the previous points, approximate, or robust solutions to SDPs in the Lasserre hierarchy of polynomial optimization can be found by SDP solvers, even when exact solutions are impossible to compute: see Henrion and Lasserre [19] and Lasserre and Magron [29]. Some of these SDPs are weakly infeasible and one of them comes from minimizing the famous Motzkin polynomial. We closely examine this SDP in Example 2.4. In other related work, the Douglas-Rachford method presented in Liu et al. [32] successfully identified infeasibility of the weakly infeasible SDPs from [31].

To sketch our contributions, we revisit Example 2.1, where we naturally describe the affine subspace

$$H = \{X \mid X \text{ is } n \text{ by } n \text{ symmetric}, \ \mathcal{A}X = b\} \tag{2.1}$$

in two ways. First, with equations $A_1 \bullet X = 0$ and $A_2 \bullet X = 2,$ where

$$A_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \text{ and } A_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{2.2}$$

and the $\bullet$ product of symmetric matrices is the trace of their regular product. By the argument in Example 2.1, this representation certifies that (ME) is infeasible.

Besides, $H = \{\lambda X_1 + X_2 \ : \ \lambda \in \mathbb{R}\}$ where

$$X_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \text{ and } X_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{2.3}$$

This generator representation proves that H is an asymptote of $\mathcal{S}^2_+,$ since $\lambda X_1 + X_2$ approaches $\mathcal{S}^2_+$ as $\lambda \to +\infty$.

We see that A_1, A_2 and X_1, X_2 share a common "echelon" structure and we may wonder whether such a structure appears in every weakly infeasible SDP. The answer is naturally no, since we can easily ruin this structure even in (ME). For example, we may take linear combinations of the equations and perform congruence transformations, in other words, replace both A_i by $T^\top A_i T$ for some invertible T.

However, it turns out that the same operations can untangle any weakly infeasible SDP. More precisely, in Theorem 1 we develop an echelon form of weakly infeasible SDPs with the following features: i) it is constructed using elementary row operations and congruence transformations; ii) it makes weak infeasibility evident, since the matrices both in the equality and in the generator representation of the underlying affine subspace have the same echelon structure; and iii) it permits us to construct any weakly infeasible SDP.

Let us explain the last point by an analogy with basic linear algebra. We know that any infeasible linear system of equations $Ax = b$ can be brought to a normal form

$$\begin{array}{rcl} A'x & = & b' \\ 0^\top x & = & 1 \end{array} \tag{2.4}$$

using elementary row operations. Thus we can verify infeasibility of a linear system using the normal form (2.4). Further, we can construct any infeasible linear system as follows: we choose A' and b' in (2.4) arbitrarily, then perform elementary row operations. This basic algorithm always succeeds and every infeasible linear system is among its outputs.

This work shows that a similar scheme works for a more involved pathology – weak infeasibility – in a much more involved problem – an SDP.

Further, in Example 2.4 we present an SDP that is naturally in our echelon form, without ever having to perform elementary row operations or congruence transformations. This SDP arises from a sum-of-squares (SOS) relaxation of minimizing the famous Motzkin polynomial; we thus hope that our work will be of interest to the sum-of-squares optimization community.

The plan of the paper is as follows. In Section 2.2 we review preliminaries consisting of basic linear algebra and SDP duality. In Section 2.3 we present and illustrate our main result, Theorem 1, and to build intuition, we prove the "easy" direction. In Section 2.4 we describe our algorithm to construct weakly infeasible SDPs and show that any weakly infeasible SDP is among its outputs. Our algorithm also constructs any bad projection of the psd cone. For the reader's convenience, some of the proofs are postponed to Section 2.5 and 2.6. The most difficult proof is the "hard" direction in Theorem 1, which we give in Section 2.6. Section 2.7 describes our problem library and computational tests. In Section 2.8 we reinterpret Theorem 1 in two ways: as a "sandwich" theorem and as a "factorization" theorem. Here we also discuss open research directions.

To make the paper's results accessible to a broad audience, we prove them using only basic results in SDP duality and linear algebra, all of which we summarize in Section 2.2. This work has some unavoidable overlap with [31], where the lemmas of Section 2.5 were already proved. On the other hand, here we prove these lemmas in a more elementary fashion. Reference [31] also gave a scheme to construct weakly infeasible SDPs in a certain restricted class; however, that scheme does not capture even some weakly infeasible SDPs with 3×3 matrices. We comment in detail on these points in Section 2.5 and Section 2.6.

2.2 Preliminaries

Operators and matrices For a linear operator (or matrix) $\mathcal{M}$, we denote its rangespace by $\mathcal{R}(\mathcal{M})$, its nullspace by $\mathcal{N}(\mathcal{M})$, and its adjoint by $\mathcal{M}^*$.

We denote the set of $n \times n$ symmetric matrices by $\mathcal{S}^n$. Further, N stands for the set $\{1, \dots, n\}$.

Given a matrix $M \in \mathbb{R}^{n \times n}$ and $R, S \subseteq N$, we denote the submatrix of M corresponding to rows in R and columns in S by $M(R, S)$. When $R = \{r\}$ is a singleton, we simply write $M(r, S)$ for $M(\{r\}, S)$. For brevity, we let $M(R) := M(R, R)$.

We denote the concatenation of matrices A and B along the diagonal by $A \oplus B$,

$$A \oplus B := \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}.$$

Thus, $M \oplus 0$ is the matrix obtained by attaching zero rows and colums to M. The dimensions of $M \oplus 0$ will be clear from the context.

Further, $X \succeq 0$ means that the matrix X is symmetric and positive semidefinite and $X \succ 0$ means it is symmetric and positive definite.

Basics of SDP duality Consider the pair of SDPs

$$
\text{(P-opt)} \quad \begin{array}{ll} \inf & C \bullet X \\ s.t. & X \text{ is feasible in (P)} \end{array} \qquad\qquad \begin{array}{ll} \sup & b^\top y \\ s.t. & \mathcal{A}^* y \preceq C, \end{array} \quad \text{(D)}
$$

where $C \in \mathcal{S}^n$, and for $T, S \in \mathcal{S}^n$ we write $T \preceq S$ to say $S - T \succeq 0$. We say that (D) is the dual of (P-opt) and vice versa, (P-opt) is the dual of (D).

When both are feasible, the optimal value of (P-opt) is at least as large as the optimal value of (D). These optimal values agree and the optimal value of (P-opt) is attained when (D) satisfies *Slater's condition*, i.e., when there is $y \in \mathbb{R}^m$ such that $C - \mathcal{A}^* y \succ 0$.

We say that (P) is *strongly infeasible*, if the distance of the affine subspace H (see (2.1)) from $\mathcal{S}^n_+$ is positive. By this definition, we see that every infeasible SDP is either strongly or weakly infeasible.

We know that (P) is strongly infeasible exactly when its *alternative system*

$$
\begin{array}{rcl} \mathcal{A}^* y & \succeq & 0 \\ b^\top y & = & -1 \end{array} \tag{P-alt}
$$

is feasible. In other words, strong infeasibility of (P) is certified by (P-alt).

We will use the above results as building blocks, since they can be proven in a few pages, with real analysis and elementary linear algebra as sole prerequisites; see Renegar [61, Chapter 3].

Reformulations In the sequel we represent the operator $\mathcal{A}$ via symmetric matrices $A_1, \dots, A_m$ as

$$
\mathcal{A}X \;=\; (A_1 \bullet X, \dots, A_m \bullet X)^\top.^2 \tag{2.5}
$$

The following definition will be used throughout the paper.

Definition 2.1. We say that we reformulate (P) if we apply to it some of the following operations (in any order):

[2]Then the adjoint $\mathcal{A}^*$ is given as $\mathcal{A}^* y = \sum_{i=1}^m y_i A_i$ for $y \in \mathbb{R}^m$.

1. Exchange (A_i, b_i) and (A_j, b_j), where i and j are distinct indices in $\{1, \dots, m\}$.

2. Replace (A_i, b_i) by $\lambda(A_i, b_i) + \mu(A_j, b_j)$, where λ and μ are reals and $\lambda \neq 0$.

3. Replace all A_i by $T^\top A_i T$, where T is a suitably chosen invertible matrix.

We also say that by reformulating (P) we obtain a reformulation; and that we reformulate the map $\mathcal{A} : \mathcal{S}^n \to \mathbb{R}^m$ if we reformulate (P) with some $b \in \mathbb{R}^m$.

We make two observations that will be useful later. First, reformulating (P) preserves its status: it is feasible (infeasible, weakly infeasible) if and only if has the same status after reformulating it. Second, in Definition 2.1 operations 1 and 2 can be naturally viewed as elementary row operations performed on the constraints of system $\mathcal{A}X = b$.

Semidefinite echelon form To motivate our next definition, suppose that a square matrix M is in row echelon, i.e., upper triangular form

$$M = \begin{pmatrix} * & * & \dots & * \\ & * & \dots & * \\ & & \ddots & \\ & & & * \end{pmatrix}$$

where the empty cells are all zeroes. Assuming the diagonal entries of M are nonzero, this form serves two purposes. First, it makes it clear that the columns of M span the whole space; and second, it shows that the nullspace of M contains only 0.

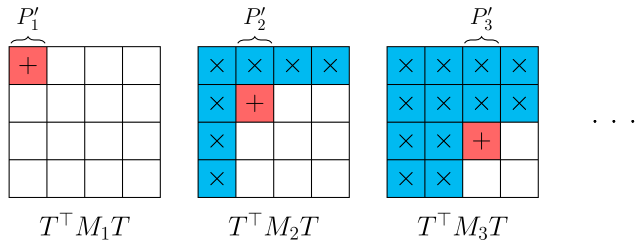
Analogously, we define an echelon form of a sequence of symmetric matrices:

Definition 2.2. We say that the sequence of symmetric $n \times n$ matrices $(M_1, \dots, M_k)$ is in semidefinite echelon form with structure $\{P_1, \dots, P_k\}$ if the following three conditions hold: i) the P_i are disjoint subsets of N, ii) for $i = 1, \dots, k$

$$\begin{array}{rcl} M_i(P_i) & \text{is} & \text{diagonal with positive diagonal entries, and} \\ M_i(P_1 \cup \dots \cup P_{i-1}, N) & \text{is} & \text{arbitrary,} \end{array} \tag{2.6}$$

and iii) the remaining elements of all M_i are zero. (Note that by symmetry $M_i(N, P_1 \cup \dots \cup P_{i-1}) = M_i(P_1 \cup \dots \cup P_{i-1}, N)$.)

Thus, for a suitable permutation matrix T the M_i look like



where the columns of M_i with indices in P_i were permuted into columns of $T^\top M_i T$ with indices in P'_i.

Here the $+$ red blocks are positive definite and diagonal, the $\times$ blue blocks may have arbitrary elements, and the white blocks are zero.

To highlight the analogy with the row echelon form, suppose $(M_1, \ldots, M_k)$ is in semidefinite echelon form with structure $\{P_1, \ldots, P_k\}$. In Section 2.3 we will see that this special form serves two purposes. First, suppose that $X \succeq 0$ satisfies $M_i \bullet X = 0$ for all i. Then by an elementary argument the rows (and columns) of X indexed by $P_1 \cup \cdots \cup P_k$ are zero. Second, suppose X is a symmetric matrix whose $N \setminus (P_1 \cup \cdots \cup P_k)$ diagonal block is positive definite. Then by another elementary argument $\sum_{i=1}^k \lambda_i M_i + X$ is positive definite for suitable $\lambda_1, \ldots, \lambda_k$ reals.

Remark 2.2.1. A sequence $(M_1, \ldots, M_k)$ in semidefinite echelon form is a type of *facial reduction sequence* [11, 44, 47]. Precisely, $(M_1, \ldots, M_k)$ certifies that any $X \succeq 0$ such that $M_i \bullet X = 0$ for all i belongs to a *face* of $\mathcal{S}^n_+$, namely the set of of psd matrices with certain rows and columns equal to zero.

Let us consider a special case when the structure of $(M_1, \ldots, M_k)$ is $\{P_1, \ldots, P_k\}$, P_1 contains the first $|P_1|$ indices of N, P_2 contains the next $|P_2|$ indices, and so on, and the positive definite blocks in all M_i are identities. These sequences were defined in [31], and baptized as *regularized facial reduction sequences.*

2.3 The main result, and the easy direction

The main result of the paper is the following.

Theorem 1. *The problem (P) is weakly infeasible if and only if it has a reformulation*

$$\begin{array}{rcl} \mathcal{A}'X & = & b' \\ X & \succeq & 0 \end{array} \tag{$\mathrm{P_{weak}}$}$$

with the following properties:

1. $(A'_1, \ldots, A'_{k+1})$ *is in semidefinite echelon form and* $(b'_1, \ldots, b'_k, b'_{k+1}) = (0, \ldots, 0, -1)$ *for some* $k \geq 1$;

2. *there is* $(X_1, \ldots, X_{\ell+1})$ *in semidefinite echelon form such that* $\ell \geq 1$ *and*

$$\begin{array}{rcl} \mathcal{A}'X_i & = & 0 \quad \textit{for } i = 1, \ldots, \ell \\ \mathcal{A}'X_{\ell+1} & = & b'. \end{array} \tag{2.7}$$

$\square$

Here we understand that $\mathcal{A}'$ is represented by symmetric matrices A'_i as

$$\mathcal{A}'X = (A'_1 \bullet X, \ldots, A'_m \bullet X)^\top. \tag{2.8}$$

If (P) is weakly infeasible, then we can choose the reformulation ($\mathrm{P_{weak}}$) so the positive definite blocks in $A'_1, \ldots, A'_k$ and $X_1, \ldots, X_\ell$ are all nonempty. Indeed, in the proof of the "only if" direction we will construct the reformulation ($\mathrm{P_{weak}}$) and X_j sequence precisely in this manner.

Example 2.2. (Example 2.1 continued) As a quick check, the problem (ME) needs only a minimal reformulation. To put it into the echelon form of ($\mathrm{P_{weak}}$), we set

$$A'_1 := A_1,\ A'_2 := -\frac{1}{2}A_2, \tag{2.9}$$

where A_1 and A_2 are given in (2.2), and use (X_1, X_2) from equation (2.3). In this SDP we have $k = \ell = 1$. $\Diamond$

It is useful to visualize Theorem 1 via the matrix of inner products of the A'_i and X_j in Figure 2.2.

$$
(A_i' \bullet X_j)_{i=1,j=1}^{m,\ell+1} = \begin{array}{cc}
\overbrace{\hphantom{0 \quad \dots \quad 0 \quad -1}}^{\ell+1} & \\
\left(\begin{array}{ccc|c}
0 & \dots & 0 & 0 \\
0 & \ddots & 0 & 0 \\
0 & \dots & 0 & 0 \\
\hline
0 & \dots & 0 & -1 \\
\hline
0 & \dots & 0 & b'_{k+2} \\
 & \ddots & & \\
0 & \dots & 0 & b'_m
\end{array}\right) & \begin{array}{l} \left.\vphantom{\begin{array}{c}0\\0\\0\\0\end{array}}\right\} k+1 \\ \vphantom{\begin{array}{c}0\\0\\0\end{array}} \end{array}
\end{array}
$$

Figure 2.2: The matrix of inner products of the A_i' and X_j in Theorem 1

The proof of the "only if" direction of Theorem 1 is technical and deferred to Section 2.6. However, the proof of the "if" direction is elementary, and we provide it below.

Proof (of "if" in Theorem 1). It suffices to prove that ($\mathrm{P_{weak}}$) is weakly infeasible. To that end, we first prove it is infeasible, so to obtain a contradiction we assume that X is feasible in it. We also assume that $(A_1', \dots, A_{k+1}')$ has structure $\{P_1, \dots, P_{k+1}\}$ (see Definition 2.2).

Since $A_1' \bullet X = 0,$ a positively weighted linear combination of the diagonal elements of $X(P_1)$ is zero. Since $X \succeq 0,$ these elements are zero, hence the rows (and columns) of X indexed by P_1 are zero.

Continuing, $A_2' \bullet X = 0, \dots, A_k' \bullet X = 0$ implies that the rows (and columns) of X indexed by $P_2 \cup \dots \cup P_k$ are zero. Hence $A_{k+1}' \bullet X$ is a positively weighted linear combination of the diagonal elements of $X(P_{k+1})$, so

$$A_{k+1}' \bullet X \geq 0.$$

This contradiction proves that ($\mathrm{P_{weak}}$) is indeed infeasible.

This process is illustrated on Figure 2.3, where the submatrices marked by $\oplus$ are positive semidefinite. For convenience we assume in this figure that the columns of all matrices indexed by P_1 come first; the columns indexed by P_2 come next; etc.

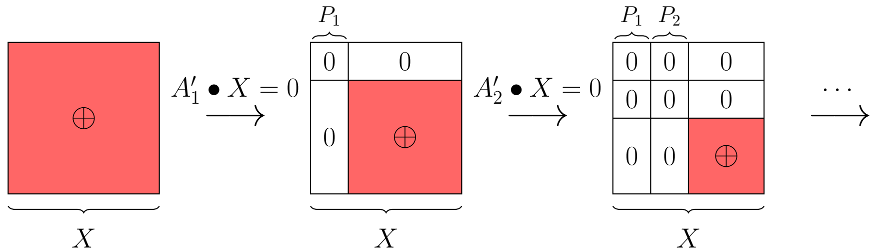


Figure 2.3: Proving that ($\mathrm{P}_{\mathrm{weak}}$) is infeasible

Next we prove that ($\mathrm{P}_{\mathrm{weak}}$) is not strongly infeasible. For that, let

$$H' = \{ X \in \mathcal{S}^n \ : \ \mathcal{A}'X = b' \},$$

and fix $\epsilon > 0$. We will construct a psd matrix which is ϵ close to H'. Suppose the structure of $(X_1, \ldots, X_{\ell+1})$ is $\{Q_1, \ldots, Q_{\ell+1}\}$ and for brevity, let $Q_{\ell+2} = N \setminus (Q_1 \cup \cdots \cup Q_{\ell+1})$.

First we define $X_\delta \in \mathcal{S}^n$ so that $X_\delta(Q_{\ell+2}) = \delta I$ and the other elements of X_δ are zero. Here $\delta > 0$ is chosen so the norm of X_δ is at most ϵ. See the leftmost picture in Figure 2.4.

Second, we define $X'_{\ell+1} := X_{\ell+1} + X_\delta$. Then the $(Q_{\ell+1} \cup Q_{\ell+2})$ diagonal block of $X'_{\ell+1}$ is positive definite and $X'_{\ell+1}$ is within ϵ distance of H' (since $X_{\ell+1} \in H'$). See the middle picture in Figure 2.4.

Next we let $X'_\ell := \gamma_\ell X_\ell + X'_{\ell+1}$ where γ_ℓ is a positive real. The definition of positive definiteness ($G \succ 0$ if $x^\top G x > 0$ for all nonzero x) implies that

$$X'_\ell(Q_\ell \cup Q_{\ell+1} \cup Q_{\ell+2}) \succ 0$$

if γ_ℓ is sufficiently large. Further, X'_ℓ is still within ϵ distance of H' (since $\mathcal{A}'X_\ell = 0$). We refer to the rightmost picture in Figure 2.4.

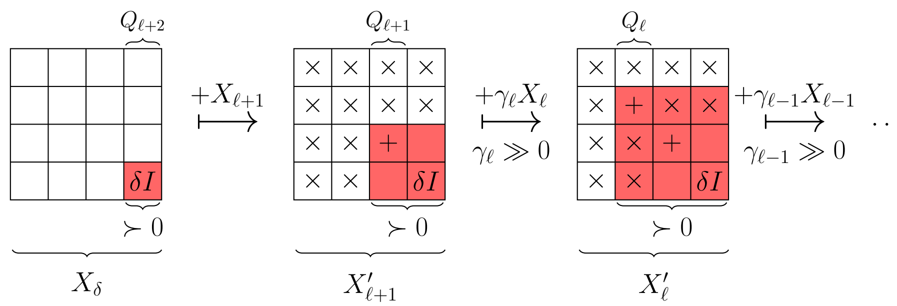


Figure 2.4: Proving that ($\mathrm{P}_{\mathrm{weak}}$) is not strongly infeasible

Continuing in this fashion we add $\gamma_{\ell-1}X_{\ell-1}$ to X'_ℓ for some large $\gamma_{\ell-1}$ and so on. Eventually we obtain a positive definite matrix, within ϵ distance of H', and conclude that ($\mathrm{P}_{\mathrm{weak}}$) is not strongly infeasible. The proof is complete.

$\square$

Remark 2.3.1. Suppose (P) is weakly infeasible. Based on Theorem 1 we can prove this to a "third party" by the following data:

1. The original problem (P) and the reformulation ($\mathrm{P}_{\mathrm{weak}}$);

2. The sequence of matrices $(X_1, \ldots, X_{\ell+1})$;

3. The operations needed to reformulate (P) into ($\mathrm{P}_{\mathrm{weak}}$). These can be encoded in a very compact manner, just by two matrices: the elementary row operations by an $m \times m$ matrix $G = (g_{ij})$ and the congruence transformations by an $n \times n$ matrix T. Then the equations

$$\begin{array}{rcll} A'_i & = & T^\top \left(\sum_{j=1}^m g_{ij} A_j\right) T & \text{for} \quad i = 1, \ldots, m \\ b' & = & Gb & \end{array} \tag{2.10}$$

hold.

So, to verify that (P) is weakly infeasible, we check that ($\mathrm{P}_{\mathrm{weak}}$) and $(X_1, \ldots, X_{\ell+1})$ are in the required form, and equations (2.7) and (2.10) hold. All these computations must be done over real numbers and in exact arithmetic. This discussion implies that the problem "is (P) weakly infeasible?" is in $\mathcal{NP}$ in the real number model of computing [3].

[3] It is also in co-$\mathcal{NP}$, since if (P) is not weakly infeasible, we can verify this by exhibiting either a feasible solution of (P), or a feasible solution of (P-alt).

This result already follows from previous works [55, 26]. Precisely, these papers described certificates to verify infeasibility of any infeasible SDP, regardless of whether it is strongly or weakly infeasible. For example, Ramana's infeasibility certificate of an SDP is a semidefinite system which is feasible exactly when the SDP in question is infeasible. Note that ($\mathrm{P}_{\mathrm{infeas}}$) in Lemma 5 also certifies infeasibility of (P). Thus, using any of these certificates, we obtain a certificate that (P) is *weakly* infeasible, if we verify that (P) and its alternative system (P-alt) are *both* infeasible.

On the other hand, our echelon form ($\mathrm{P}_{\mathrm{weak}}$) does more than just verify weak infeasibility. It makes weak infeasibility evident to see; and we can use it to conveniently construct any weakly infeasible SDP, a feature that previously known certificates do not have.

Note that the following related question: "Can we decide the feasibility status (feasibility, or weak/strong infeasibility) of (P) in polynomial time?" is open in the real number model, and in the Turing model as well.

Example 2.3. The SDP in the form (P) with data

$$
\begin{aligned}
A_1 &= \begin{pmatrix} 8 & -1 & -9 & -2 \\ -1 & -26 & 3 & 39 \\ -9 & 3 & 10 & 3 \\ -2 & 39 & 3 & -16 \end{pmatrix}, A_2 = \begin{pmatrix} 5 & -3 & -6 & -2 \\ -3 & -6 & 5 & 21 \\ -6 & 5 & 7 & 2 \\ -2 & 21 & 2 & -11 \end{pmatrix} \\
A_3 &= \begin{pmatrix} -6 & -3 & 7 & 4 \\ -3 & 34 & 1 & -43 \\ 7 & 1 & -8 & -5 \\ 4 & -43 & -5 & 18 \end{pmatrix}, A_4 = \begin{pmatrix} 5 & 4 & -9 & -6 \\ 4 & -28 & 6 & 48 \\ -9 & 6 & 13 & 5 \\ -6 & 48 & 5 & -21 \end{pmatrix} \qquad (2.11) \\
b &= (-44, -22, 44, -68)^\top
\end{aligned}
$$

is weakly infeasible, but from this form this would be very difficult to tell.

However, once we reformulate (2.11) by the formulas in (2.10) and the G and T matrices

$$G = \frac{1}{2}\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 1 & 0 \\ 1 & 1 & 3 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \quad T = \begin{pmatrix} -1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix},$$

it is brought into the form ($\mathrm{P}_{\mathrm{weak}}$) with the A_i' and X_j shown on Figure 2.5.

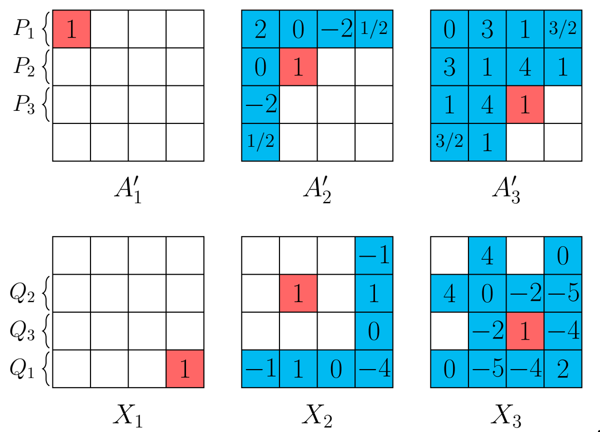


Figure 2.5: The A_i' and X_j obtained from reformulating the SDP (2.11)

In the reformulation equations $A_1' \bullet X = A_2' \bullet X = 0$, $A_3' \bullet X = -1$ certify infeasibility and (X_1, X_2, X_3) certify not strong infeasibility. The matrix A_4' is omitted (and is straightforward to compute from the formulas in (2.10)). Note that now $k = \ell = 2$. $\Diamond$

The reader may ask whether some SDPs are naturally in the echelon form of ($\mathrm{P}_{\mathrm{weak}}$) *without even having to reformulate them.* We next present such an SDP from a prominent application of semidefinite programming, polynomial optimization.

We first recall a definition. Given a multivariate polynomial $f = f(x_1, \dots, x_n)$, we say that f is a sum of squares (SOS) if $f = \sum_{i=1}^{t} f_i^2$ for some t positive integer and f_i polynomials. An SOS polynomial is of course nonnegative. On the other hand, the first example of a nonnegative, but not SOS polynomial was given by Motzkin in [35] and there are many more nonnegative polynomials than SOS polynomials: see Blekherman [9].

Example 2.4. Given the famous Motzkin polynomial

$$f(x, y) = 1 - 3x^2y^2 + x^2y^4 + x^4y^2, \tag{2.12}$$

we can find its infimum over $\mathbb{R}^2$ by solving the problem

$$\begin{array}{lll} \sup & \lambda & \\ s.t. & f(x,y) - \lambda & \geq \quad 0. \end{array} \tag{2.13}$$

In the SOS relaxation of (2.13) proposed in [28] and [43] we solve the following problem instead:

$$\begin{array}{ll} \sup & \lambda \\ s.t. & f - \lambda \quad \text{is} \quad \text{SOS}. \end{array} \tag{2.14}$$

In turn, we formulate (2.14) as an SDP as follows. We define a vector of monomials [4]

$$z = (x^2, y^2, x, y, xy, xy^2, x^2y, 1)^\top,$$

then we know that $f - \lambda$ is SOS if and only if $f - \lambda = X \bullet zz^\top$ for some $X \succeq 0$.

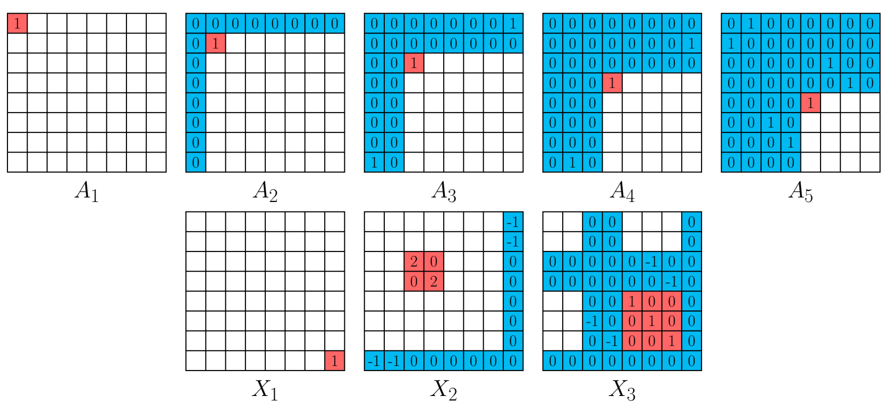


Figure 2.6: Certificates of weak infeasibility in the Motzkin polynomial SDP

We then match the coefficients of monomials in $f - \lambda$ and $X \bullet zz^\top$ and obtain the SDP

[4] To strictly follow the SOS recipe we should also include in z the monomials x^3 and y^3. We omitted these for simplicity, but it is straightforward to check that even if we do include them, the resulting SDP is still in the echelon form of ($\mathrm{P}_{\mathrm{weak}}$).

$$
\begin{array}{lrcll}
\sup & -E_{88} \bullet X & & & \\
s.t. & E_{11} \bullet X & = & 0 & (x^4) \\
 & E_{22} \bullet X & = & 0 & (y^4) \\
 & (E_{33} + E_{18}) \bullet X & = & 0 & (x^2) \\
 & (E_{44} + E_{28}) \bullet X & = & 0 & (y^2) \\
 & (E_{55} + E_{12} + E_{36} + E_{47}) \bullet X & = & -3 & (x^2y^2) \\
 & & \vdots & & \\
 & X & \succeq & 0. &
\end{array} \tag{2.15}
$$

In (2.15) the E_{ij} are unit matrices in $\mathcal{S}^8$ whose elements in the (i,j) and (j,i) position are 1 and the rest zero. For each equation we show the corresponding monomial in parentheses. For example, $E_{11} \bullet X = 0$ because $f - \lambda$ has no x^4 term. Note that in (2.15) we indicated the equation corresponding to x^2y and several other equations only by vertical dots.

Of course, we know that $f - \lambda$ is not SOS for any $\lambda,$ hence (2.15) is infeasible. We next verify that it is weakly infeasible, and in the echelon form ($\mathrm{P}_{\mathrm{weak}}$) without ever having to reformulate it.

We see that $A_1 := E_{11}, A_2 := E_{22}, \ldots, A_5 := E_{55} + E_{12} + E_{36} + E_{47}$ is in semidefinite echelon form, hence the equations in (2.15) prove it is infeasible (the last right hand side is -3, not -1 as in Theorem 1, but this does not matter).

On the other hand, let

$$
\begin{array}{lcl}
X_1 & := & E_{88} \\
X_2 & := & 2E_{33} + 2E_{44} - E_{18} - E_{28} \\
X_3 & := & E_{55} + E_{66} + E_{77} - E_{47} - E_{36}.
\end{array} \tag{2.16}
$$

Then (X_1, X_2, X_3) is in semidefinite echelon form and proves that (2.15) is not strongly infeasible. To see why, we write the equations in (2.15) as $\mathcal{A}X = b,$ then we can check that $\mathcal{A}X_1 = \mathcal{A}X_2 = 0$ and $\mathcal{A}X_3 = b.$

In Figure 2.6 we visualize the certificates of infeasibility (on the top) and the certificates of not-strong-infeasibility (on the bottom).

$\Diamond$

To better explain Example 2.4, we make three remarks.

First, SDPs that come from polynomial optimization problems are widely known to be difficult, both due to their often pathological behavior, and also due to their size. On the one hand, some remedies to address the difficult behaviors are available. For example, if such an SDP is feasible, we can ensure strong duality by adding a redundant ball constraint, see Henrion and Josz [22]. See also references [19, 29] mentioned in the introduction. One may also entirely do away with the SDP based approach, and either optimize directly over SOS polynomials, see Papp and Yildiz [41]; or use a second order conic programming, or linear programming relaxation, which is a bit weaker, but much more scalable, see Ahmadi and Majumdar [1]. Example 2.4 complements these works: it gives a combinatorial insight into why some of the pathologies arise in the first place.

Second, Waki in [67] constructed a library of weakly infeasible SDPs from the SOS relaxation of polynomial optimization problems; on the other hand [67] did not provide certificates of the kind we study in this work.

Third, suppose we just wish to decide whether $f - \lambda$ is SOS for some *fixed* λ. For that, we set up an SDP feasibility problem with the constraints of (2.15), and add the constraint $E_{88} \bullet X = 1 - \lambda$. Interestingly, this SDP turns out to be strongly infeasible, as it was proved by Henrion [18].

We now move on, and in Corollary 1 characterize the underlying operators in weakly infeasible SDPs. The discussion in the introduction shows that these are linear operators that map $\mathcal{S}^n_+$ to a nonclosed set. These operators were recently baptized as "bad projections of the psd cone," and explored through the lens of algebraic geometry [21].

Corollary 1. *Suppose $\mathcal{A} : \mathcal{S}^n \to \mathbb{R}^m$ is a linear map. Then $\mathcal{A}\mathcal{S}^n_+$ is not closed if and only if $\mathcal{A}$ has a reformulation $\mathcal{A}'$ with the following properties:*

1. *$(A'_1, \ldots, A'_{k+1})$ is in semidefinite echelon form, where $k \geq 1$;*
2. *There is $(X_1, \ldots, X_{\ell+1})$, in semidefinite echelon form, where $\ell \geq 1$ and the matrix of inner products of the A'_i and X_j matrices looks like*

$$
(A_i' \bullet X_j)_{i=1,j=1}^{m,\ell+1} = \begin{array}{c} \overbrace{\qquad\qquad\qquad}^{\ell+1} \\ \left(\begin{array}{ccc|c} 0 & \dots & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & \dots & 0 & 0 \\ \hline 0 & \dots & 0 & -1 \\ \hline 0 & \dots & 0 & \times \\ & \ddots & & \\ 0 & \dots & 0 & \times \end{array}\right) \end{array} \begin{array}{l} \left.\begin{array}{c} \\ \\ \\ \\ \\ \end{array}\right\} k+1 \\ \\ \\ \\ \end{array}
$$

where the $\times$ symbols denote arbitrary elements.

$\square$

Proof. To show the "only if" direction, suppose $\mathcal{A}\mathcal{S}_+^n$ is not closed, and suppose $b \in \mathbb{R}^m$ is in the closure of $\mathcal{A}\mathcal{S}_+^n$ but $b \notin \mathcal{A}\mathcal{S}_+^n$. Then (P) is weakly infeasible, so we appeal to Theorem 1 and construct $\mathcal{A}', b',$ and $X_1, \dots, X_{\ell+1}$ therein. Then the matrix $(A_i' \bullet X_j)_{i=1,j=1}^{m,\ell+1}$ is in the form given in Figure 2.2, so items 1 and 2 in our corollary hold.

For the "if" direction, suppose $\mathcal{A}'$ and $(X_1, \dots, X_{\ell+1})$ are as in the statement of Corollary 1, and let $b' = \mathcal{A}' X_{\ell+1}$. Then $(\mathcal{A}', b')$ and $(X_1, \dots, X_{\ell+1})$ satisfy items 1 and 2 in Theorem 1. Hence the system ($\mathrm{P_{weak}}$) therein is weakly infeasible, so b' is in the closure of $\mathcal{A}'\mathcal{S}_+^n$ but $b' \notin \mathcal{A}'\mathcal{S}_+^n$. Thus $\mathcal{A}'\mathcal{S}_+^n$ is not closed, hence neither is $\mathcal{A}\mathcal{S}_+^n$, as required. $\square$

We next contrast Corollary 1 with an equivalent characterization of nonclosedness of $\mathcal{A}\mathcal{S}_+^n$ that we recap below in Theorem 2. Theorem 2 is obtained from [48, Theorem 1] by setting $B = 0$.

Theorem 2. *Suppose that Z is a maximum rank psd matrix in $\mathcal{R}(\mathcal{A}^*)$, the linear span of $A_1, \dots, A_m$. Assume without loss of generality that Z is of the form*

$$
Z = \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix} \tag{2.17}
$$

for some $r \in \{0, \ldots, n\}$. Then $\mathcal{A}\mathcal{S}^n_+$ is not closed if and only if there is a matrix $V \in \mathcal{R}(\mathcal{A}^*)$ of the form

$$V = \begin{pmatrix} V_{11} & V_{12} \\ V_{12}^\top & V_{22} \end{pmatrix}, \tag{2.18}$$

where $V_{22} \in \mathcal{S}^{n-r}_+$ and $\mathcal{R}(V_{12}^\top) \not\subseteq \mathcal{R}(V_{22})$. $\square$

Next we argue that Corollary 1 is more useful than Theorem 2, although the latter is more compact. In particular, Corollary 1 can be used to construct maps under which $\mathcal{S}^n_+$ is not closed, as we show in Section 2.4.

On the other hand, Theorem 2 cannot be used for this purpose in a straightforward manner. Suppose indeed that we try to construct such an $\mathcal{A}$ and choose $A_1 := Z$ as in (2.17) (with $0 < r < n$) and $A_2 := V$ as in (2.18) as elements of $\mathcal{R}(\mathcal{A}^*)$. However, we cannot guarantee that Z remains a maximum rank psd matrix in $\mathcal{R}(\mathcal{A}^*)$ after we chose the other A_i.

2.4 How to construct any weakly infeasible SDP and bad projection of the psd cone

We now build on Theorem 1, and present a combinatorial algorithm, Algorithm 1, to construct any weakly infeasible SDP of the form (P). Algorithm 1 (with Algorithm 2 as a subroutine) also constructs any bad projection of the psd cone, and provides a vector b in the closure of $\mathcal{A}\mathcal{S}^n_+$ such that $b \notin \mathcal{A}\mathcal{S}^n_+$.

Algorithm 1 first chooses positive integers m, k and ℓ, such that $k+1 \leq m$, then constructs a weakly infeasible SDP with just $k+1$ constraints in the echelon form of ($\mathrm{P_{weak}}$). It also chooses a sequence $(X_1, \ldots, X_{\ell+1})$ in semidefinite echelon form that certifies not strong infeasibility of this "small" SDP. Finally, it chooses the remaining $m-(k+1)$ equality constraints and reformulates the SDP.

Algorithms 1 and 2 rely on Theorem 1, but to simplify notation, everywhere we write A_i in place of A_i' and b_i in place of b_i'.

Algorithm 1 Construct Weakly Infeasible SDP

1: Choose m, k and ℓ positive integers such that $k+1 \leq m$. Also choose $(A_1, \dots, A_{k+1})$ and $(X_1, \dots, X_{\ell+1})$ in semidefinite echelon form, which satisfy the following *base equations:*

$$\boxed{A_i \bullet X_j = \begin{cases} 0 & \text{if } (i,j) \neq (k+1, \ell+1) \\ -1 & \text{if } (i,j) = (k+1, \ell+1) \end{cases}} \tag{BASE}$$

2: Let $(b_1, \dots, b_k, b_{k+1}) = (0, \dots, 0, -1)$.

3: Choose $A_{k+2}, \dots, A_m$ so they have zero $\bullet$ product with $X_1, \dots, X_\ell$.

4: Set $b_i := A_i \bullet X_{\ell+1}$ for $i = k+2, \dots, m$.

5: Reformulate (P).

Observe that step 1 ensures that the first $k+1$ rows of the $(A_i \bullet X_j)_{i,j=1}^{m,\ell+1}$ matrix are as required in Figure 2.2. Steps 3 and 4 ensure that the rest of the matrix looks like as required in the same figure.

The only nontrivial step in Algorithm 1 is step 1, so the question is, how to carry out this step?

The main idea is that we have many potentially nonzero blocks in the A_i and the X_j, and only a small number of equations to satisfy. Precisely, by a straightforward count the A_i altogether have at least constant times k^3 potentially nonzero blocks of the form $A_i(P_s, P_t)$, where $\{P_1, \dots, P_{k+1}\}$ is the structure of $(A_1, \dots, A_{k+1})$. Similarly, the X_j have at least constant times ℓ^3 potentially nonzero blocks. So if we set these blocks in the right order, then the A_i and X_j will satisfy the (BASE) equations, of which there are only $(k+1)(\ell+1)$.

To carry out this plan, we need two lemmas.

Lemma 1. *Suppose that $(A_1, \dots, A_{k+1})$ is in semidefinite echelon form with structure $\{P_1, \dots, P_{k+1}\}$ and $(X_1, \dots, X_{\ell+1})$ is in semidefinite echelon form with structure $\{Q_1, \dots, Q_{\ell+1}\}$.*

Also suppose that $(A_1, \dots, A_{k+1})$ and $(X_1, \dots, X_{\ell+1})$ satisfy the base equations (BASE). *Then*

$$P_1 \cap (Q_1 \cup \dots \cup Q_{\ell+1}) = \emptyset \tag{2.19}$$

$$Q_1 \cap (P_1 \cup \dots \cup P_{k+1}) = \emptyset. \tag{2.20}$$

Proof. We prove (2.20), the proof of (2.19) is analogous. Since $(A_1, \ldots, A_{k+1})$ is in semidefinite echelon form, and $X_1 \succeq 0$, an argument like in the "if" direction in Theorem 1 proves $X_1(P_1 \cup \cdots \cup P_{k+1}, N) = 0$. Since the only nonzero entries of X_1 are in $X_1(Q_1)$, the statement follows.

$\square$

Lemma 2 shows how to solve a linear system of equations in an unusual setup, in which only the right hand side is fixed. In Lemma 2 we index the b_j reals and Y_j matrices from 2 to $\ell + 1$ for convenience. We also define the inner product of possibly nonsymmetric matrices M and Y as the trace of $M^\top Y$.

Lemma 2. *Given positive integers p, q and ℓ, and real numbers $b_2, \ldots, b_{\ell+1}$ there is a polynomial time algorithm to find $M, Y_2, \ldots, Y_{\ell+1}$ in $\mathbb{R}^{p \times q}$ such that*

$$\begin{aligned} M \bullet Y_2 &= b_2 \\ &\vdots \\ M \bullet Y_{\ell+1} &= b_{\ell+1}. \end{aligned} \tag{2.21}$$

Further, any solution to (2.21) is a possible outcome of this algorithm.

Proof. If all b_j are zero, we first choose an arbitrary M, then choose $Y_2, \ldots, Y_{\ell+1}$ to solve the system (2.21). If not all b_j are zero, we do the same, but we make sure to pick $M \neq 0$.

$\square$

Algorithm 2, which is used as a subroutine in Algorithm 1, constructs sequences of A_i and X_j that satisfy the (BASE) equations.

Algorithm 2 Base Equations Algorithm

1: Choose $(A_i)_{i=1}^{k+1}$ and $(X_j)_{j=1}^{\ell+1}$ in semidefinite echelon form with structure $\{P_i\}_{i=1}^{k+1}$ and $\{Q_j\}_{j=1}^{\ell+1}$ respectively, which satisfy

$$P_1 \neq \emptyset, \ldots, P_k \neq \emptyset,\ Q_1 \neq \emptyset, \ldots, Q_\ell \neq \emptyset,\ (2.19),\ \text{and}\ (2.20).$$

2: **for** $i = 2 : k+1$ **do**

3: Set $A_i(P_{i-1}, Q_1), X_2(P_{i-1}, Q_1), \ldots, X_{\ell+1}(P_{i-1}, Q_1)$ to satisfy the base equations (BASE) with left hand side $A_i \bullet X_2, \ldots, A_i \bullet X_{\ell+1}$.

4: **end for**

Lemma 3. *We can implement step 3 of Algorithm 2 so the algorithm is correct.*

Proof. For brevity, we define $P_{k+2} := N \setminus (P_1 \cup \cdots \cup P_{k+1})$ and we fix $i \in \{2, \ldots, k+1\}$. To implement step 3 of Algorithm 2 we first set the (P_{i-1}, Q_1) block of $A_i, X_2, \ldots, X_{\ell+1}$ to zero, then introduce a target vector $(b_2, \ldots, b_{\ell+1})^\top$:

$$b_j = \begin{cases} -\frac{1}{2} A_i \bullet X_j & \text{if } (i,j) \neq (k+1, \ell+1) \\ -\frac{1}{2}\big(A_i \bullet X_j + 1\big) & \text{if } (i,j) = (k+1, \ell+1). \end{cases}$$

Next we invoke Lemma 2 with $A_i(P_{i-1}, Q_1)$ in place of M and $X_j(P_{i-1}, Q_1)$ in place of Y_j for $j = 2, \ldots, \ell+1$. Finally we symmetrize A_i and the X_j, namely we set

$$A_i(Q_1, P_{i-1}) := A_i(P_{i-1}, Q_1)^\top \text{and } X_j(Q_1, P_{i-1}) := X_j(P_{i-1}, Q_1)^\top$$

for $j = 2, \ldots, \ell+1$.

Suppose we perform step 3 with a certain $i \in \{2, \ldots, k+1\}$ to satisfy the base equations (BASE) with left hand side $A_i \bullet X_j$ for $j = 2, \ldots, \ell+1$. We next show that the previously satisfied equations remain true.

If $i = 2$ then there is nothing to show, so assume $i \geq 3$. Let us fix $t \in \{2, \ldots, i-1\}$ and $j \in \{2, \ldots, \ell+1\}$. We will show that the equation with left hand side $A_t \bullet X_j$ remains satisfied.

For that, we note that the support of A_t is contained in the blocks

$$\begin{aligned} \mathcal{I}_1 &:= (P_1 \cup \cdots \cup P_t, P_1 \cup \cdots \cup P_t) \\ \mathcal{I}_2 &:= (P_1 \cup \cdots \cup P_{t-1}, P_{t+1} \cup \cdots \cup P_{k+2}) \\ \mathcal{I}_3 &:= (P_{t+1} \cup \cdots \cup P_{k+2}, P_1 \cup \cdots \cup P_{t-1}) \end{aligned} \tag{2.22}$$

cf. Definition 2.1. In iteration i we change $X_j(P_{i-1}, Q_1)$ and $X_j(Q_1, P_{i-1})$ for $j = 2, \ldots, \ell + 1$. So it suffices to show

$$(P_{i-1}, Q_1) \cap \mathcal{I}_1 = \emptyset \tag{2.23}$$

$$(P_{i-1}, Q_1) \cap \mathcal{I}_2 = \emptyset \tag{2.24}$$

$$(P_{i-1}, Q_1) \cap \mathcal{I}_3 = \emptyset. \tag{2.25}$$

Indeed, (2.23) and (2.25) follow by (2.20). Further, (2.24) follows from $t - 1 < i - 1$ and the proof is complete.

□

In summary, we have the following theorem.

Theorem 3. *Algorithm 1 always correctly constructs a weakly infeasible SDP and any weakly infeasible SDP of the form* (P) *is among its outputs.*

Proof. Lemmas 1–3 imply that Algorithm 1 always correctly outputs a weakly infeasible SDP. On the other hand, suppose (P) is weakly infeasible. Then (P) has a reformulation ($\mathrm{P}_{\mathrm{weak}}$) as presented in Theorem 1. For simplicity, let us denote the operator in ($\mathrm{P}_{\mathrm{weak}}$) by $\mathcal{A}$ and represent $\mathcal{A}$ with matrices $A_1, \ldots, A_m$. Also let us denote the right hand side in ($\mathrm{P}_{\mathrm{weak}}$) by b.

Assume the first $k + 1 \geq 2$ equations in ($\mathrm{P}_{\mathrm{weak}}$) prove it is infeasible. We know that ($\mathrm{P}_{\mathrm{weak}}$) is not strongly infeasible, and we let $(X_1, \ldots, X_{\ell+1})$ be the sequence that certifies this as presented in Theorem 1. Recall that $\ell \geq 1$.

Suppose that $(A_1, \ldots, A_{k+1})$ has structure $\{P_1, \ldots, P_{k+1}\}$ and $(X_1, \ldots, X_{\ell+1})$ has structure $\{Q_1, \ldots, Q_{\ell+1}\}$. By the remark following Theorem 1 we can assume that $P_1, \ldots, P_k$ and $Q_1, \ldots, Q_\ell$ are nonempty. Hence by Lemma 2 we have that $(A_1, \ldots, A_{k+1})$ and $(X_1, \ldots, X_{\ell+1})$ are possible outputs of Algorithm 2. Besides, $A_{k+2}, \ldots, A_m$ and $b_{k+2}, \ldots, b_m$ are possible outputs of steps 3 and 4 in Algorithm 1. Because of the reformulation step 5 we see that (P) is a possible output of Algorithm 1, and the proof is complete.

□

As a quick check, Algorithm 1 constructs a variant of (ME) as follows. First it sets $m = 2$, $k = \ell = 1$, $P_1 = \{1\}$, $Q_1 = \{2\}$, $P_2 = Q_2 = \emptyset$ and

$$A_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & \alpha \\ \alpha & 0 \end{pmatrix}, X_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, X_2 = \begin{pmatrix} 0 & \beta \\ \beta & 0 \end{pmatrix},$$

where α and β are arbitrary. Then the subroutine Algorithm 2 sets α and β to satisfy $2\alpha\beta = -1$. Algorithm 1 in step 2 sets $b = (0, -1)^\top$. Then it skips steps 3 and 4 (since $m = k + 1$) and also skips the reformulation of step 5.

Example 2.5. Let $k = \ell = 1$, $P_1 = \{1\}$, $P_2 = \{2\}$, $Q_1 = \{3\}, Q_2 = \{2\}$, α and β be arbitrary reals, and

$$A_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & 0 & \alpha \\ 0 & 1 & 0 \\ \alpha & 0 & 0 \end{pmatrix}, X_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, X_2 = \begin{pmatrix} 0 & 0 & \beta \\ 0 & 1 & 0 \\ \beta & 0 & 0 \end{pmatrix}.$$

Our algorithms construct a weakly infeasible SDP from this data as follows. Algorithm 2 sets α and β so that $\alpha\beta = -1$. After this the A_i and X_j satisfy the (BASE) equations. Then Algorithm 1 in step 2 sets $(b_1, b_2) = (0, -1)$ and in step 3 it chooses

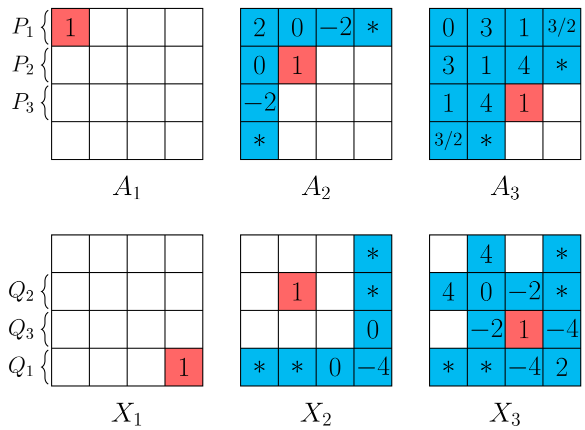
$$A_3 := \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix},$$

as it has zero inner product with X_1. Finally, in step 4 it sets $b_3 := A_3 \bullet X_2 = 0$ and skips the reformulation of step 5. $\Diamond$

We note that a scheme to construct weakly infeasible SDPs was given in [31]. That scheme is somewhat similar to the one presented here, as it uses a sequence of A_i matrices to certify infeasibility and a sequence of X_j matrices to certify not strong infeasibility. However, the scheme in [31] assumes that the positive definite blocks of the A_i and X_j matrices do not overlap, so the variety of weakly infeasible SDPs it can construct is limited. In particular, it cannot even construct the small SDP of Example 2.5.

On the other hand, our Algorithm 1, that generates *any* weakly infeasible SDP, relies on Theorem 1; that result is simple to state, but somewhat technical to prove.

Example 2.6. (Example 2.3 continued) We now show how Algorithm 1 constructs the SDP in Example 2.3. We first set $m = 4, k = \ell = 2,$ and select (A_1, A_2, A_3) and (X_1, X_2, X_3) in semidefinite echelon form shown below:



Here at the start the entries marked by $*$ are arbitrary. Algorithm 2 first ensures $A_2 \bullet X_2 = A_2 \bullet X_3 = 0$ by setting

$$A_2(1,4) = 1/2, \quad X_2(1,4) = -1, \quad X_3(1,4) = 0.$$

Then it ensures $A_3 \bullet X_2 = 0, A_3 \bullet X_3 = -1$ by setting

$$A_3(2,4) = 1, \quad X_2(2,4) = 1, \quad X_3(2,4) = -5.$$

(It also sets the symmetric blocks, for example, it sets $A_2(4,1) = 1/2$ and so on.)

Next, Algorithm 1 in step 2 sets $(b_1, b_2, b_3) = (0, 0, -1)$, then in step 3 it selects

$$A_4 := \frac{1}{2} \begin{pmatrix} -1 & -2 & -1 & 3 \\ -2 & 2 & -1 & 2 \\ -1 & -1 & 0 & -1 \\ 3 & 2 & -1 & 0 \end{pmatrix}$$

as it is orthogonal to X_1 and X_2. Then in step 4 it sets $b_4 = A_4 \bullet X_3 = -12$. We thus get the weakly infeasible instance of Example 2.3 with the A_i and X_j shown above and
$b = (0, 0, -1, -12)^\top$.

$\Diamond$

2.5 Proofs: certificates of infeasibility and not strong infeasibility separately

In this section we construct two distinct reformulations of (P): one to certify it is infeasible, and the second to certify it is not strongly infeasible. Lemma 5 already appeared as part (1) of Theorem 5 in [31] and Lemma 6 as part (2) of Theorem 5 in [31]. An earlier version of the latter result appeared in [34]. Here we give shorter and more elementary proofs.

We first state a necessary condition for a semidefinite system to be infeasible. Lemma 4 is a slightly stronger version of Lemma 3.2 in Waki and Muramatsu [68].

Lemma 4. *Suppose $B \in \mathcal{S}^n$ and $L \subseteq \mathcal{S}^n$ is a subspace such that $(B + L) \cap \mathcal{S}^n_+ = \emptyset$. Then the following hold:*

1. *The system* (2.26) *is feasible:*

$$\begin{aligned} B \bullet Y &\leq 0 \\ Y &\in L^\perp \cap \left(\mathcal{S}^n_+ \setminus \{0\}\right). \end{aligned} \tag{2.26}$$

2. *If* (2.26) *has a positive definite solution, then it has a positive definite solution Y such that $B \bullet Y < 0$.*

Proof. To prove item 1, let $\mathcal{B} : \mathbb{R}^m \to \mathcal{S}^n$ be a linear map whose rangespace is L. We claim that the optimal value of the SDP

$$\begin{array}{ll} \sup & y_0 \\ s.t. & -By_0 - \mathcal{B}y \preceq 0 \end{array} \tag{2.27}$$

is zero. Indeed, its optimal value is nonnegative, since $(y, y_0) = (0, 0)$ is feasible in it. On the other hand, the optimal value cannot be positive: if $y_0 > 0$ were feasible with some y, then we would get the contradiction $B + \frac{1}{y_0}\mathcal{B}y \succeq 0$.

First assume that (2.27) satisfies Slater's condition. Then its dual (which is of the form (P-opt)), is feasible. Any Y feasible in the dual of (2.27) satisfies

$$B \bullet Y = -1, \quad \mathcal{B}^* Y = 0,$$

as required.

Second, assume that (2.27) does not satisfy Slater's condition. We claim that the optimal value of the SDP

$$\begin{array}{ll} \sup & t \\ s.t. & tI - By_0 - \mathcal{B}y \preceq 0 \end{array} \tag{2.28}$$

is zero. Indeed, it is nonnegative since setting all variables to zero we obtain a feasible solution. On the other hand if $t > 0$ were feasible with some y_0 and y, then the contradiction $By_0 + \mathcal{B}y \succeq tI \succ 0$ would follow.

Note that (2.28) does satisfy Slater's condition with $t = -1$, $y = 0$, and $y_0 = 0$. Thus there is a Y feasible in the dual of (2.28), which satisfies

$$B \bullet Y = 0, \quad \mathcal{B}^* Y = 0, \quad I \bullet Y = 1,$$

as required.

To prove item 2, we observe that $(B + L) \cap \mathcal{S}^n_+ = \emptyset$ implies $B \notin L$. So there is $Y' \in L^\perp$ such that

$$B \bullet Y' < 0.$$

Suppose (2.26) has a positive definite solution. Then we add a sufficiently small positive multiple of Y' to it and obtain a Y positive definite feasible solution such that $B \bullet Y < 0$. The proof is now complete.

$\square$

Lemma 5. *The SDP* (P) *is infeasible if and only if it has a reformulation*

$$\begin{array}{rcl} \mathcal{A}'X & = & b' \\ X & \succeq & 0 \end{array} \tag{$\mathrm{P_{infeas}}$}$$

in which $(A'_1, \ldots, A'_{k+1})$ *is in semidefinite echelon form and* $(b'_1, \ldots, b'_k, b'_{k+1}) = (0, \ldots, 0, -1)$ *for some* $k \geq 0$.

Proof. The "if" direction is given verbatim in the proof of the "if" direction in Theorem 1. For the "only if" direction, by elementary row operations (operations 1 and 2 in Definition 2.1) we will first achieve the following:

$$A_1 \succeq 0, \; b_1 \in \{0, -1\} \tag{2.29}$$

and

$$b_1 = 0 \Rightarrow 0 < \operatorname{rank} A_1 < n. \tag{2.30}$$

For that, we distinguish two cases.

Case 1 (*The linear system* $\mathcal{A}X = b$ *is infeasible*)**.** Then by elementary linear algebra there is y such that

$$\mathcal{A}^* y = 0, \quad b^\top y = -1.$$

We have $y \neq 0$ so after permuting the equations in $\mathcal{A}X = b$ we assume $y_1 \neq 0$ without loss of generality. Next, using elementary row operations we replace A_1 by $\mathcal{A}^* y$ and b_1 by $b^\top y$. As a result, (2.29) and (2.30) hold.

Case 2 (*The linear system* $\mathcal{A}X = b$ *is feasible*)**.** We first fix $X_0 \in \mathcal{S}^n$ such that $\mathcal{A}X_0 = b$. Then we apply Lemma 4 with $L := \mathcal{N}(\mathcal{A})$ and $B := X_0$ and find Y feasible in the system (2.26). Further, using item 2 in Lemma 4, if Y is positive definite, we ensure $X_0 \bullet Y < 0$. We have

$Y \in L^{\perp},$ and $L^{\perp} = \mathcal{R}(\mathcal{A}^*),$ so we write $Y = \mathcal{A}^* y$ for some $y \in \mathbb{R}^m$ and deduce

$$0 \, \geq \, X_0 \bullet Y \, = \, X_0 \bullet \mathcal{A}^* y \, = \, (\mathcal{A} X_0)^\top y = b^\top y.$$

We then proceed as in Case 1: we permute the equations in $\mathcal{A}X = b$, if needed, and replace A_1 by $\mathcal{A}^* y$ and b_1 by $b^\top y$. Afterwards, if $b_1 < 0$ then we rescale A_1 and b_1 so that $b_1 = -1$. Again, (2.29) and (2.30) hold.

Now that we have satisfied (2.29) and (2.30), we choose an invertible T matrix such that $T^\top A_1 T = I_r \oplus 0$ for some $r \geq 0,$ and let

$$A_i' := T^\top A_i T, \; b_i' := b_i \; \text{ for } i = 1, \ldots, m. \tag{2.31}$$

If $b_1' = -1,$ we set $k = 0$ and stop.

If $b_1' = 0,$ then we must have $0 < r < n$. We must also have $m > 1,$ otherwise the all zero matrix would be feasible in (P). We then delete the equation $A_1' \bullet X = 0$ and also delete the first r rows and columns from the other A_i'. We thus obtain a smaller SDP, say (P$'$), with $m - 1$ equations and order $n - r$ matrices. We see that (P$'$) is infeasible: if X' were feasible in it, then $X := 0 \oplus X'$ would be feasible in (P). So we proceed by induction, as a reformulation of (P$'$) into the form of (P$_{\text{infeas}}$) yields a reformulation of (P) into the same form.

□

As a quick sanity check, we consider the SDP in (ME), and the reformulation given in Example 2.2 (see equation (2.9)). This reformulation is in the form (P$_{\text{infeas}}$). Note that now $k = 1$.

The proof of Lemma 5 implies that the positive definite blocks in $A_1', \ldots, A_k'$ are nonempty, and can be chosen as identity matrices. However, the positive definite block in A_{k+1}' may be empty, as it is in the reformulated version of (ME) given in the previous paragraph.

We next present Lemma 6 to construct a certificate that (P) is not strongly infeasible. Lemma 6 builds on two ideas. First, if (P) is not strongly infeasible, then the alternative system (P-alt) is infeasible. In turn, if (P-alt) is infeasible, then using Lemma 5 we will reformulate it to make its infeasibility evident.

Lemma 6. *The SDP (P) is not strongly infeasible if and only if it has a reformulation*

$$\begin{array}{rcl} \mathcal{A}''X & = & b'' \\ X & \succeq & 0, \end{array} \tag{$\mathrm{P}_{\mathrm{notstrong}}$}$$

such that for some $(X_1, \ldots, X_{\ell+1})$ *in semidefinite echelon form with* $\ell \geq 0$ *the following holds:*

$$\begin{array}{rcl} \mathcal{A}''X_i & = & 0 \quad \textit{for} \;\; i = 1, \ldots, \ell \;\; \textit{and} \\ \mathcal{A}''X_{\ell+1} & = & b''. \end{array} \tag{2.32}$$

Proof. The proof of the "if" direction is given in the proof of the "if" direction in Theorem 1.

For the "only if" direction, we assume that (P) is not strongly infeasible, and choose an operator $\mathcal{B} : \mathcal{S}^n \to \mathbb{R}^m$ such that $\mathcal{R}(\mathcal{A}^*) = \mathcal{N}(\mathcal{B})$. We also choose $X_0 \in \mathcal{S}^n$ such that $\mathcal{A}X_0 = b$ (such an X_0 must exist, otherwise (P) would be strongly infeasible). Since (P) is not strongly infeasible, the alternative system (P-alt) is infeasible. We claim that (P-alt) is equivalent to

$$\begin{array}{rcl} \mathcal{B}Y & = & 0 \\ X_0 \bullet Y & = & -1 \\ Y & \succeq & 0. \end{array} \tag{2.33}$$

Indeed, by the choice of $\mathcal{B}$ we have that $\mathcal{B}Y = 0$ for some $Y \in \mathcal{S}^n$ iff $Y = \mathcal{A}^*y$ for some $y \in \mathbb{R}^m$. For any such Y and y we see that

$$b^\top y \;=\; (\mathcal{A}X_0)^\top y \;=\; X_0 \bullet Y,$$

and this proves that (P-alt) and (2.33) are equivalent.

Thus, by Lemma 5, the system (2.33) has a reformulation of the form ($\mathrm{P}_{\mathrm{infeas}}$), in which for some $\ell \geq 0$ the first $\ell + 1$ equations prove the infeasibility. These equations are of the form

$$\begin{array}{rcl} X_j \bullet Y & = & 0 \quad (j = 1, \ldots, \ell) \\ X_{\ell+1} \bullet Y & = & -1, \end{array} \tag{2.34}$$

where $(X_1, \ldots, X_{\ell+1})$ is in semidefinite echelon form.

Note that in (2.33) the only equation with nonzero right hand side is $X_0 \bullet Y = -1$. Given that from (2.33) we derived equations (2.34) by elementary row operations and by congruence transformations, we see that

$$
\begin{array}{rcll}
X_j & \in & T^\top \mathcal{R}(\mathcal{B}^*) T & \text{for } j = 1, \dots, \ell \\
X_{\ell+1} & \in & T^\top \big(\mathcal{R}(\mathcal{B}^*) + X_0 \big) T &
\end{array}
\tag{2.35}
$$

for some invertible matrix T.

Observe that $\mathcal{R}(\mathcal{B}^*) = \mathcal{N}(\mathcal{A})$. Then from (2.35) we deduce that for $i = 1, \dots, m$

$$
A_i \bullet T^{-\top} X_j T^{-1} = \begin{cases} 0 & \text{if} \quad j \in \{1, \dots, \ell\} \\ b_i & \text{if} \quad j = \ell + 1 \end{cases}
$$

holds. We have $A_i \bullet T^{-\top} X_j T^{-1} = T^{-1} A_i T^{-\top} \bullet X_j$ for all i. We define the operator $\mathcal{A}''$ as

$$
\mathcal{A}'' X = (A_1'' \bullet X, \dots, A_m'' \bullet X)^\top,
$$

where $A_i'' = T^{-1} A_i T^{-\top}$ for all i and let $b'' = b$. We see that $\mathcal{A}'', b'',$ and the $X_1, \dots, X_{\ell+1}$ that we already defined satisfy the requirements of our lemma.

$\square$

Yet again, consider the reformulation of (ME) given in Example 2.2 and set $A_1'' = A_1'$, and $A_2'' = A_2'$, and X_1 and X_2 as in (2.3). Then (A_1'', A_2'') and (X_1, X_2) with $b'' = (0, -1)^\top$ satisfy the conclusions of Lemma 6. Note that now $\ell = 1$.

By the proof of Lemma 6 the positive definite blocks in $X_1, \dots, X_\ell$ are nonempty, and can be chosen as identity matrices. However, the positive definite block in $X_{\ell+1}$ may be empty, as it is in the reformulated version of (ME) that we gave in the previous paragraph.

2.6 Proof of Theorem 1

Section 2.5 showed how to produce a reformulation ($\mathrm{P}_{\mathrm{infeas}}$) to prove that (P) is infeasible; and another reformulation ($\mathrm{P}_{\mathrm{notstrong}}$) to prove it is not strongly infeasible. In this section we show that a single reformulation can accomplish both. This common reformulation was fairly straightforward to produce when we started with a simple problem like (ME). In the general case we need a technical proof.

We first define operators that transform a certain targeted block of a matrix. To absorb Definition 2.3 we need to recall the notation $M(R,S)$ and $M(R)$ for blocks of a matrix M from the start of Section 2.2.

Definition 2.3. Suppose $R \subseteq N$ and G is matrix of order $|R|$. The matrix $I_{R,G}$ is obtained from the $n \times n$ identity by replacing $I(R)$ by G, i.e., by performing the following two steps:

$$I_{R,G} := I,$$
$$I_{R,G}(R) := G.$$

For example, if $n = 4$, $R = \{1, 4\}$, and $G = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix}$, then

$$I_{R,G} = \begin{pmatrix} 2 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 4 & 0 & 0 & 5 \end{pmatrix}.$$

Suppose $M \in \mathbb{R}^{n \times n}$ and $R \subseteq N$. Then the operation

$$M := M \cdot I_{R,G}$$

right multiplies by G the columns of M indexed by R and leaves the rest of M unchanged.

Given subsets $R_1, \ldots, R_t$ of N and indices i and j such that $1 \leq i \leq j \leq t$ we will use the following shorthand:

$$R_{i:j} := R_i \cup R_{i+1} \cup \cdots \cup R_j. \tag{2.36}$$

We will often use a congruence transformation to put matrices into a convenient block diagonal form, bringing us to the following lemma:

Lemma 7. *Suppose $X \in \mathcal{S}^n$ and $R_1, \ldots, R_t$ are disjoint subsets of N such that*

$$X(R_{1:t}) \succeq 0.$$

Then there is an invertible matrix T such that

$$(T^\top XT)(R_{1:t}) \text{ is nonnegative diagonal,}$$

and T can be chosen as the product of $n \times n$ invertible matrices

$$T = I_{R_1,U_1} W_1 \ldots I_{R_t,U_t} W_t,$$

where

1. *the U_i are orthonormal matrices for all i.*
2. *right multiplying an $n \times n$ matrix, say M, by W_i adds multiples of columns in $M(N, R_i)$ to columns of $M(N, R_j)$ for some j indices in $\{i+1, \ldots, t\}$.*

□

Suppose W_i is a matrix in the statement of Lemma 7. We can describe W_i algebraically as follows: i) it has all 1 entries on the main diagonal; ii) the block $W_i(R_i, R_j)$ is nonzero for some j indices in $\{i+1, \ldots, t\}$; iii) all other blocks of W_i are zero.

Proof (of Lemma 7). Let U_1 be a matrix of orthonormal eigenvectors of $X(R_1)$ and define $T := I_{R_1,U_1}$. Then $(T^\top XT)(R_{1:t})$ looks like on the first picture of Figure 2.7, where the $\times$ symbols represent arbitrary elements.

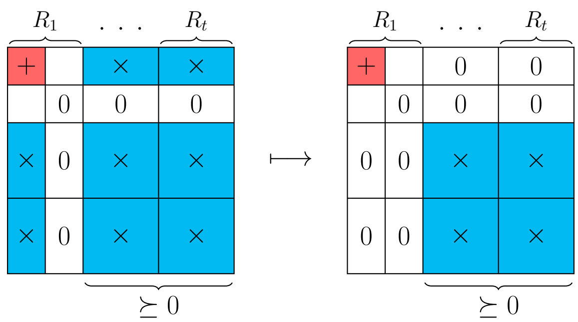


Figure 2.7: How to diagonalize $X(R_{1:t})$ in Lemma 7

Next we let W_1 be a matrix such that right multiplying $T^\top XT$ by W_1 adds columns of $T^\top XT$ indexed by R_1 to columns indexed by R_j to zero out the $T^\top XT(R_1, R_j)$ block for all $j > 1$. Then the $R_{1:t}$ region of $W_1^\top T^\top XTW_1$ looks like in the right picture on Figure 2.7.

We then redefine $T := TW_1$ and $X := T^\top XT$, and continue in like fashion with the $R_{2:t}$ diagonal block of X.

$\square$

The next definition is from the theory of facial reduction [11, 47].

Definition 2.4. We say that a sequence of symmetric matrices $X_1, \dots, X_t$ is a facial reduction sequence for $\mathcal{S}^n_+$ if

$$X_1 \in \mathcal{S}^n_+, \quad \text{and} \quad X_{i+1} \in \left(\mathcal{S}^n_+ \cap X_1^\perp \cap \cdots \cap X_i^\perp\right)^* \quad \text{for} \;\; i = 1, \dots, t-1.$$

Here, for a set $C \subseteq \mathcal{S}^n$ we write $C^* = \{Y : X \bullet Y \geq 0 \text{ for all } X \in C\}$ for its dual cone.

Evidently, if $(X_1, \dots, X_t)$ is in semidefinite echelon form, then it is a facial reduction sequence, but the converse is not true in general.

Lemma 8 below follows from Lemma 1 in [31]; however, below we give a simpler proof.

Lemma 8. *Suppose that $(X_1, \dots, X_t)$ is a facial reduction sequence, and V is an invertible matrix. Then $(V^\top X_1 V, \dots, V^\top X_t V)$ is also a facial reduction sequence.*

Proof. Let $(X_1, \dots, X_t)$ be as stated. For brevity, define the map $\mathcal{V} : \mathcal{S}^n \to \mathcal{S}^n$ as $\mathcal{V}X = V^\top XV$ for $X \in \mathcal{S}^n$. Then the conjugate of $\mathcal{V}$ is computed as $\mathcal{V}^*Y = VYV^\top$ for $Y \in \mathcal{S}^n$.

Let us fix $i \in \{1, \ldots, t-1\}$ and let $Y \in \mathcal{S}^n_+ \cap (\mathcal{V}X_1)^\perp \cap \cdots \cap (\mathcal{V}X_i)^\perp$. We will show

$$\mathcal{V}X_{i+1} \bullet Y \geq 0, \tag{2.37}$$

and this will prove our claim. From the definition of the conjugate we deduce

$$\mathcal{V}^*Y \in \mathcal{S}^n_+ \cap X_1^\perp \cap \cdots \cap X_i^\perp. \tag{2.38}$$

Hence $\mathcal{V}X_{i+1} \bullet Y = X_{i+1} \bullet \mathcal{V}^*Y \geq 0$, where the inequality follows from (2.38) and from $(X_1, \ldots, X_t)$ being a facial reduction sequence. Hence (2.37) follows and the proof is complete.

□

We can now prove the difficult direction in Theorem 1.

Proof (of "only if" in Theorem 1). Suppose that (P) is weakly infeasible and Lemma 5 produced the reformulation ($\mathrm{P}_{\mathrm{infeas}}$) with operator $\mathcal{A}'$ and right hand side b'. We claim that $k \geq 1$, so to obtain a contradiction, suppose $k = 0$. Then the alternative system of ($\mathrm{P}_{\mathrm{infeas}}$) (namely the system (P-alt) with $(\mathcal{A}', b')$ in place of $(\mathcal{A}, b)$) is feasible, in particular, $y = (1, 0, \ldots, 0)^\top$ is feasible in it. Hence ($\mathrm{P}_{\mathrm{infeas}}$) is strongly infeasible. Thus (P) is also strongly infeasible, which is the desired contradiction.

Also suppose that Lemma 6 produced the reformulation ($\mathrm{P}_{\mathrm{notstrong}}$) with operator $\mathcal{A}''$ and right hand side b''; and it produced the sequence $(X_1, \ldots, X_{\ell+1})$ which is in semidefinite echelon form and certifies that ($\mathrm{P}_{\mathrm{notstrong}}$) is not strongly infeasible. We claim that $\ell \geq 1$. Indeed, if ℓ were 0, then X_1 would be feasible in ($\mathrm{P}_{\mathrm{notstrong}}$), hence (P) would also be feasible, which would be a contradiction.

As usual, we represent the operator $\mathcal{A}'$ with matrices A_i' and the operator $\mathcal{A}''$ with matrices A_i'' for $i = 1, \ldots, m$. Further, following the proof of Lemma 5 we assume without loss of generality that the positive definite blocks in the A_i' are identities.

If ($\mathrm{P}_{\mathrm{notstrong}}$) is the same as ($\mathrm{P}_{\mathrm{infeas}}$), then there is nothing to do. Otherwise, since both are reformulations of (P), we can transform ($\mathrm{P}_{\mathrm{notstrong}}$) into ($\mathrm{P}_{\mathrm{infeas}}$) if we

1. perform a sequence of elementary row operations on the equations $A_i'' \bullet X = b_i''$; then

2. replace all A_i'' by $V^\top A_i'' V$ for some invertible matrix V.

Suppose we perform only the elementary row operations, and for simplicity we still call the resulting reformulation ($\mathrm{P}_{\mathrm{notstrong}}$) with operator $\mathcal{A}''$ (represented by matrices A_i'') and right hand side b''. Of course, now $b' = b''$. Afterwards equations (2.32) still hold. At this point we have

$$\begin{aligned} A_i' &= V^\top A_i'' V && \text{for } \ i = 1, \dots, m, \\ A_i'' \bullet X_s &= V^\top A_i'' V \bullet V^{-1} X_s V^{-\top} && \text{for } \ i = 1, \dots, m; \ \text{for } \ s = 1, \dots, \ell + 1, \end{aligned} \tag{2.39}$$

where the second set of equations follows from the properties of the $\bullet$ product. We next perform the following operations:

$$X_s := V^{-1} X_s V^{-\top} \quad \text{for } \ s = 1, \dots, \ell + 1. \tag{2.40}$$

Let us consider the following *invariant conditions*, where $j \in \{0, \dots, \ell + 1\}$:

(INV-1): The semidefinite system ($\mathrm{P}_{\mathrm{infeas}}$) is a reformulation of (P) with properties given in Lemma 5. In particular, $(A_1', \dots, A_{k+1}')$ is in semidefinite echelon form and $(b_1', \dots, b_k', b_{k+1}') = (0, \dots, 0, -1)$.

(INV-2):

$$\begin{aligned} \mathcal{A}' X_s &= 0 \ \text{ for } \ s = 1, \dots, \ell \\ \mathcal{A}' X_{\ell+1} &= b'. \end{aligned} \tag{2.41}$$

(INV-3): $(X_1, \dots, X_{\ell+1})$ is a facial reduction sequence.

(INV-4): $(X_1, \dots, X_j)$ is in semidefinite echelon form.

We claim that all these conditions hold when $j = 0$. Indeed, (**INV-1**) holds since ($\mathrm{P}_{\mathrm{infeas}}$) was constructed in Lemma 5. Condition (**INV-4**) holds vacuously. Condition (**INV-2**) holds by (2.39), by $b' = b''$, and since we performed the operations in (2.40). Finally, condition (**INV-3**) holds by Lemma 8, since we started with $(X_1, \dots, X_{\ell+1})$ being in semidefinite echelon form, then we performed operations (2.40).

The goal is to have the invariant conditions satisfied with $j = \ell + 1$. Once that is done, the proof is complete, since we can set $(\mathrm{P}_{\mathrm{weak}})$ equal to $(\mathrm{P}_{\mathrm{infeas}})$. So let us assume that $j \in \{0, \ldots, \ell\}$ is an integer, and all the invariant conditions hold with j.

We will perform Step j below, which diagonalizes a certain block of X_{j+1} and in the process also transforms $\mathcal{A}'$ and the other X_j. We will prove that afterwards the invariant conditions hold with $j + 1$. Recall the notation (2.36).

Step j We assume that $(A'_1, \ldots, A'_{k+1})$ has structure $\{P_1, \ldots, P_{k+1}\}$ and set $P_{k+2} := N \setminus P_{1:(k+1)}$. We also assume that $(X_1, \ldots, X_j)$ has structure $\{Q_1, \ldots, Q_j\}$.

By condition (**INV-4**) and using an argument like in the proof of the "if" direction in Theorem 1 we see that $\mathcal{S}^n_+ \cap X_1^\perp \cap \cdots \cap X_j^\perp$ is the set of psd matrices whose rows and columns corresponding to $Q_{1:j}$ are zero. By condition (**INV-3**) we have

$$X_{j+1} \in \left(\mathcal{S}^n_+ \cap X_1^\perp \cap \cdots \cap X_j^\perp\right)^*,$$

hence

$$X_{j+1}(N \setminus Q_{1:j}) \succeq 0. \tag{2.42}$$

Let us define $R_i = P_i \setminus Q_{1:j}$ for $i = 1, \ldots, k+2$. Then we rewrite (2.42) as

$$X_{j+1}(R_1 \cup \cdots \cup R_{k+2}) \succeq 0. \tag{2.43}$$

So X_{j+1} looks like on Figure 2.8, where the red submatrix is positive semidefinite, and the blue and white submatrices are arbitrary.

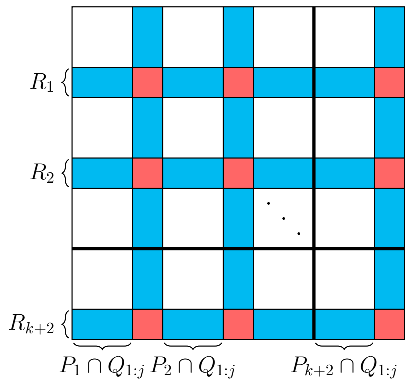


Figure 2.8: X_{j+1} before it is transformed

We now apply Lemma 7 with $X := X_{j+1}$, $t := k+2$ and the R_i just defined. The goal is to diagonalize $X_{j+1}(R_{1:t})$. Let T be the transformation matrix produced by Lemma 7, then

$$T = I_{R_1,U_1} W_1 \dots I_{R_{k+2},U_{k+2}} W_{k+2}, \tag{2.44}$$

where

(a) the U_i are orthonormal matrices for all i.

(b) right multiplying an $n \times n$ matrix, say M, by W_i adds multiples of columns in $M(N, R_i)$ to columns of $M(N, R_j)$ where $j \in \{i+1, \dots, k+2\}$.

We perform the operations

$$\begin{aligned} X_s &:= T^\top X_s T \quad \text{for} \quad s = 1, \dots, \ell+1, \\ A_i' &:= T^{-1} A_i' T^{-\top} \quad \text{for} \quad i = 1, \dots, m, \end{aligned} \tag{2.45}$$

and set

$$Q_{j+1} := \{t \mid \text{the } (t,t) \text{ element of } X_{j+1}(N \setminus Q_{1:j}) \text{ is positive}\}.$$

We claim that the invariant conditions now hold for $j+1$. Indeed, (**INV-2**) holds by how we redefined the X_s and A_i' in (2.45). Condition (**INV-3**) holds by Lemma 8.

We next look at (**INV-4**). We first show on Figure 2.9 how X_{j+1} looks before and after step (2.45). To better see what happened, we permuted the rows and columns of X_{j+1}, so that rows (and columns) indexed by $Q_{1:j}$ come first. The $\oplus$ block stands for a positive semidefinite block.

Step (2.45) transforms $X_{j+1}(N \setminus Q_{1:j})$ to be nonnegative diagonal, and in this process $X_{j+1}(Q_{1:j}, N \setminus Q_{1:j})$ (and the symmetric counterpart) is also transformed. We show the changed block in red in Figure 2.9.

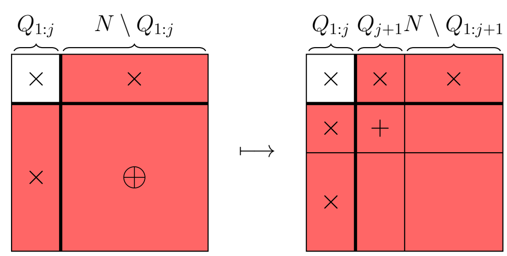


Figure 2.9: How step (2.45) changes X_{j+1}

Next we look at how $X_1, \ldots, X_j$ change due to step (2.45), so we fix $s \in \{1, \ldots, j\}$. Given the factorization of T in (2.44), replacing X_s by $T^\top X_s T$ amounts to running Algorithm 3 below:

Algorithm 3 Transforming X_s

for $t = 1 : (k+2)$ **do**

(*) $X_s := I_{R_t,U_t}^\top X_s I_{R_t,U_t}$;

(**) $X_s := W_t^\top X_s W_t$;

end for

We claim that after Algorithm 3 is run, the matrix X_s remains in the same shape it was in before. Suppose this is true after we performed steps (*) and (**) for $t = 1, \ldots, q-1$, where $q \geq 1$.

We next perform step (*) with $t = q$. This amounts to first multiplying $X_s(N, R_q)$ from the right by U_q, then multiplying $X_s(R_q, N)$ from the left by $U_q^\top$. Since $R_1 \cup \cdots \cup R_{k+2} = N \setminus Q_{1:j}$, we see that

$$R_q \subseteq N \setminus Q_{1:j} \subseteq N \setminus Q_{1:s}.$$

We depict X_s on Figure 2.10, with the affected parts shaded in red and conclude that X_s remains in the shape it was in before step (2.45). Note that on Figure 2.10 we permuted the rows and columns of X_s so that $X_s(Q_{1:s})$ is in the upper left corner.

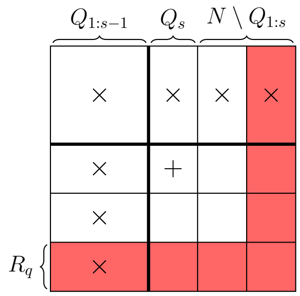


Figure 2.10: How steps (*) and (**) in Algorithm 3 change X_s, where $s \leq j$.

We next perform step (**) with $t = q$. Multiplying X_s from the right by W_q adds multiples of columns in $X_s(N, R_q)$ to columns in $X_s(N, R_{q'})$ where $q' \in \{q+1, \ldots, k+2\}$. Then we perform the analogous row operations. Thus Figure 2.10 again tells us that X_s remains in the same shape.

We thus conclude that condition (**INV-4**) holds after step (2.45) with $j+1$ instead of j.

We next prove that condition (**INV-1**) remains unchanged after step (2.45) is executed, so we look at how the A'_i change. Let us fix $i \in \{1, \ldots, m\}$.

Given the decomposition (2.44), we have

$$\begin{aligned} T^{-\top} &= I^{-\top}_{R_1 U_1} W_1^{-\top} \ldots I^{-\top}_{R_{k+2}, U_{k+2}} W^{-\top}_{k+2}, \\ T^{-1} &= W^{-1}_{k+2} I^{-1}_{R_{k+2} U_{k+2}} \ldots W_1^{-1} I^{-1}_{R_1, U_1}. \end{aligned} \tag{2.46}$$

We also know that for all $t \in \{1, \ldots, k+2\}$ by the the definition of $I_{R,G}$ and by $U_t^\top = U_t^{-1}$ the following hold:

$$\begin{aligned} I^{-1}_{R_t, U_t} &= I_{R_t, U_t^\top} \\ I^{-\top}_{R_t, U_t} &= I_{R_t, U_t}. \end{aligned} \tag{2.47}$$

Thus, given the decomposition (2.46), performing step (2.45) on A'_i amounts to running Algorithm 4 below.

Algorithm 4 Transforming A'_i

---

for $t = 1 : (k+2)$ **do**

(*) $A'_i := I_{R_t, U_t^\top} A'_i I_{R_t, U_t}$;

(**) $A'_i := W_t^{-1} A'_i W_t^{-\top}$;

end for

We claim that after Algorithm 4 is run, the matrix A_i' remains in the shape it was in before.

Suppose this is true after we performed steps (*) and (**) for $t = 1, \dots, q-1$, where $q \geq 1$. Next we perform (*) with $t = q$. We need to keep in mind that $q \in \{1, \dots, k+2\}$ and $R_q \subseteq P_q$.

We distinguish three cases.

q = i We show on Figure 2.11 the A_i' matrix before and after step (*). The changed portion is in red.

First the submatrix $A_i'(N, R_q)$ is multiplied from the right by U_q, then the submatrix $A_i'(R_q, N)$ is multiplied from the left by $U_q^\top$.

Thus $A_i'(R_q) = I$ is replaced by $U_q^\top I U_q = I$, so it remains an identity. In summary, A_i' has the same form before and after step (*).

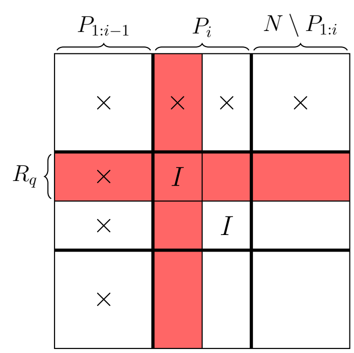


Figure 2.11: How step (*) in Algorithm 4 changes A_i', when $q = i$

q < i We show on Figure 2.12 the A_i' matrix, before and after step (*). The changed portion is in red.

First the submatrix $A_i'(N, R_q)$ is multiplied from the right by U_q, then the submatrix $A_i'(R_q, N)$ is multiplied from the left by $U_q^\top$.

Again, A_i' has the same form before and after step (*).

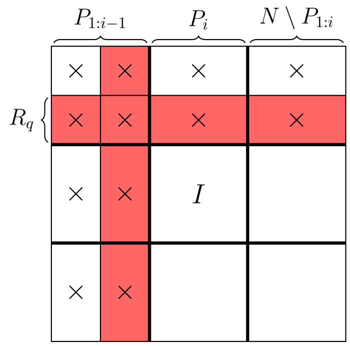


Figure 2.12: How step (*) in Algorithm 4 changes A'_i, when $q < i$

$\mathbf{q > i}$ We show on Figure 2.13 the A'_i matrix, before and after step (*). The changed portion is again in red.

First the submatrix $A'_i(N, R_q)$ is multiplied from the right by U_q, then the submatrix $A'_i(R_q, N)$ is multiplied from the left by $U_q^\top$.

Yet again, A'_i has the same form before and after step (*).

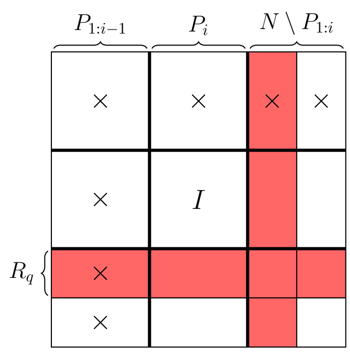


Figure 2.13: How step (2.45) changes A'_i, when $q > i$

Next we perform step (**) with $t = q$. We recall that right multiplying A'_i by W_q adds columns of $A'_i(N, R_q)$ to columns in $A'_i(N, R_{q'})$ where $q < q'$. By the algebraic description of W_q (after the statement of Lemma 7) we see that multiplying A'_i from the right by $W_q^{-\top}$ adds columns of $A'_i(N, R_{q'})$ to columns in $A'_i(N, R_q)$ where $q < q'$. (Multiplying A'_i from the left by W_q^{-1} works analogously on the rows of A'_i.)

Recall that $R_q \subseteq P_q$ and $R_{q'} \subseteq P_{q'}$. Thus Figures 2.11, 2.12, and 2.13 tell us that A_i' remains in the same shape as it was in before step (**). Thus condition (**INV-1**) holds, and the proof is complete.

$\square$

2.7 Our problem library and computational tests

Using the results of the previous sections, we now show how to generate a library of weakly infeasible SDPs. We accompany our SDPs with an intuitive visualization and examine whether their infeasibility can be recognized by the prominent SDP solvers MOSEK [4] and SDPA-GMP [15].

Libraries of weakly infeasible SDPs are available [31, 67] the latter of these was generated using the Lasserre relaxation of polynomial optimization problems. On the other hand, any weakly infeasible SDP is a possible output of our Algorithm 1, so our current library includes instances that are unlikely to appear in any previous collection.

The instances We constructed all 80 instances using Algorithm 1 and split them into two classes: clean and messy.

- We constructed our clean instances by steps 1–4 of Algorithm 1, without using the reformulation step of step 5, so our clean SDPs are in the echelon form ($\mathrm{P}_{\mathrm{weak}}$).
- From each clean instance we created a corresponding messy instance as follows. We applied elementary row operations that we represent by an $m \times m$ integral matrix $F = (f_{ij})$, then a congruence transformation that we represent by an $n \times n$ integral matrix T.

  That is, if (P) is a clean instance, then in the corresponding messy instance constraint i is

$$T^\top (\sum_{j=1}^{m} f_{ij} A_j) T \bullet X \;=\; \sum_{j=1}^{m} f_{ij} b_j.$$

We further categorize the instances as "miniature", "small", "medium", and "large", with parameters given in Table 2.1. In each category the clean and messy instances are in one-to-one correspondence. For example, from the "mini, clean, 9" instance we constructed the "mini, messy,

	Miniature	Small	Medium	Large
k	1	3	3	4
ℓ	1	1–3	2–4	2–4
n	3	5–15	25–40	120–240
m	2–5	4–7	4–7	5–8

Table 2.1: Parameters of our weakly infeasible SDPs: $k+1$ being the length of the infeasibility certificate, $\ell+1$ the length of the not-strong infeasibility certificate, n the matrix order, and m the number of constraints.

9" instance. In all instances all data is integral, so their weak infeasibility can be verified by hand, in exact arithmetic.

Data storage For convenience we give our instances in three formats.

1. In the ".mat" files (in Matlab format) the $A_1, \ldots, A_m$ are stored as rows of a matrix A and the $X_1, \ldots, X_{\ell+1}$ are stored as rows of a matrix X. These files also contain the right hand side b.

   For each messy instance the files also include the matrices F and T that were used to create it from the corresponding clean instance.

2. The ".cbf" and ".dat-s" files contain the same SDPs. The ".cbf" files can be directly read by MOSEK and the ".dat-s" files can be directly read by SDPA-GMP.

The ".jpg" files in the "image" subdirectories contain visualizations of the matrices A_i, X_j and F and T for each problem. Matrices $A_1, \ldots, A_{k+1}$ and $X_1, \ldots, X_{\ell+1}$ are color coded, just like in Figure 2.5.

Computational testing For computational testing we selected the SDP solvers MOSEK and SDPA-GMP as representative industry standards. MOSEK is currently the only commercially available SDP solver, it is fast and accurate on most industrial problems, however, it has limited numerical precision. On the other hand, SDPA-GMP can carry out calculations with precision 10^{-200}.

The results are in Table 2.2, where we reported a solver's output as "correct" if it marked an instance as infeasible.

	Miniature		Small		Medium/Large	
	Clean	Messy	Clean	Messy	Clean	Messy
MOSEK	0	0	0	0	0	0
SDPA-GMP	10	10	0	2	0	0
Total correct	10	10	10	10	20	20

Table 2.2: Number of infeasible instances correctly identified by MOSEK and SDPA-GMP

We see that while MOSEK failed to identify infeasibility of any of the SDP instances, SDPA-GMP correctly identified the infeasibility of all miniature and of some small instances. However, both solvers failed on the the larger instances.

Besides standalone SDP solvers we also tested two implementations of facial reduction. The first is the Sieve-SDP algorithm [69], which works very well if the constraint matrices of an SDP are in semidefinite echelon form. Not surprisingly, Sieve-SDP quickly proved infeasibility of all clean instances, but failed on all messy instances. The second implementation is the facial reduction method of Permenter and Parrilo [52] which performed very similarly.

The problem instances are available from the webpage of the first author.

2.8 Discussion and conclusion

We presented an echelon form of weakly infeasible SDPs that permits us to construct any weakly infeasible SDP and any bad projection of the psd cone by a combinatorial algorithm. We conclude with a discussion.

First we recall normal forms of other types of SDPs and linear maps; these normal forms are similar in spirit, but much easier to derive. For example, in [49] and in Section 2.2.1 in [48] we produced a normal form of *well-behaved* semidefinite systems of the form

$$\sum_{i=1}^{m} x_i A_i \preceq B, \tag{2.48}$$

and called the normal form a *good reformulation.* The system (2.48) with $B = 0$ is well behaved iff $\mathcal{A}\mathcal{S}_+^n$ is closed, hence we obtain a normal form of linear maps that carry $\mathcal{S}_+^n$ to a *closed* set.

This normal form also permits us to construct any such linear map. In contrast, to derive such a normal (echelon) form of weakly infeasible SDPs and of maps that carry $\mathcal{S}_+^n$ to a *nonclosed* set, we needed a much more technical proof.

Next, we reinterpret Theorem 1 in two equivalent forms.

The first interpretation is a "sandwich theorem" which we state in terms of weak infeasibility of $H \cap \mathcal{S}_+^n$ where the affine subspace H is defined in (2.1).

Theorem 4. *The semidefinite program $H \cap \mathcal{S}_+^n$ is weakly infeasible if and only if there are positive integers k and ℓ, an invertible matrix T, and sequences $(A_1', \dots, A_{k+1}')$ and $(X_1, \dots, X_{\ell+1})$, both in semidefinite echelon form such that*

$$\begin{aligned} X_{\ell+1} + \operatorname{lin}\{ X_1, \dots, X_\ell \} &\subseteq T^\top H T \\ &\subseteq \{X \ : \ A_1' \bullet X = \dots = A_k' \bullet X = 0, \ A_{k+1}' \bullet X = -1\}. \end{aligned}$$

$\square$

Here $\operatorname{lin}\{ X_1, \dots, X_\ell \}$ stands for the linear span of $X_1, \dots, X_\ell,$ and $T^\top H T$ for the set $\{ T^\top X T \ : \ X \in H \}$.

The second interpretation is a "factorization" result. Suppose that ($\mathrm{P}_{\mathrm{weak}}$) in Theorem 1 was obtained by elementary row operations that we represent by an $m \times m$ matrix G and by congruence transformations that we represent by an $n \times n$ matrix T, see the discussion in Remark 2.3.1. We define the map $\mathcal{T} : \mathcal{S}^n \to \mathcal{S}^n$ as $\mathcal{T}(X) = TXT^\top$ for $X \in \mathcal{S}^n$. Then in Theorem 1 the operators $\mathcal{A}$ and $\mathcal{A}'$ and vectors b and b' are related as

$$\mathcal{A}' = G\mathcal{A}\mathcal{T}, \, b' = Gb,$$

so $(\mathcal{A}, b)$ can be "factorized" into $(\mathcal{A}', b')$ (and vice versa).

Next we comment on computing the echelon form ($\mathrm{P}_{\mathrm{weak}}$). In general, we do not have an efficient or stable method to do that, since we would need to solve the SDPs that arise in Lemma 4 in exact arithmetic. However, the complexity of even finding a feasible solution to an SDP is unknown (see Remark 2.3.1). Thus the echelon form ($\mathrm{P}_{\mathrm{weak}}$) is similar in spirit to the Jordan normal form of a matrix, which also must be computed in exact arithmetic [16]. At the same time, even though they are nontrivial to compute, both our echelon form, and the Jordan normal form yield both theoretical and practical insights.

We finally mention some intriguing research questions. First, it may be of interest to interpret our results from the viewpont of projective geometry, in the spirit of Naldi and Sinn's recent paper [36]. Second, recall again Example 2.4: here the SDP from minimizing the sum-of-squares (SOS) relaxation of the Motzkin polynomial is weakly infeasible, and is in the echelon form of $(\mathrm{P}_{\mathrm{weak}})$ without having to reformulate it. It would be interesting to see whether the same is true of other sum-of-squares SDPs.

CHAPTER 3

How do exponential size solutions arise in semidefinite programming?

3.1 Introduction

Linear programs and polynomial size solutions The classical linear programming (LP) feasibility problem asks whether a system of linear inequalities

$$Ax \geq b$$

has a solution, where the matrix A and the vector b both have integer entries. When the answer is "yes", then by a classical argument a feasible rational x has size at most $2n^2 \log n$ times the size of (A, b), where n is the number of variables. When the answer is "no", there is a certificate of infeasibility whose size is similarly bounded. Here and in the sequel by "size" of a matrix or vector we mean the number of bits necessary to describe it.

Semidefinite programs and exponential size solutions Semidefinite programs (SDPs) are a far reaching generalization of linear programs, and they have attracted widespread attention in the last few decades. An SDP feasibility problem can be formulated as

$$x_1 A_1 + \cdots + x_m A_m + B \succeq 0, \tag{P}$$

where the A_i and B are symmetric matrices with integer entries and $S \succeq 0$ means that the symmetric matrix S is positive semidefinite.

In a stark contrast to a linear program, the solutions of (P) may have exponential size in the size of the input. This surprising fact is illustrated by a classical example of Khachiyan:

$$x_1 \geq x_2^2,\ x_2 \geq x_3^2, \ldots, x_{m-1} \geq x_m^2,\ x_m \geq 2. \tag{\textit{Khachiyan}}$$

We can formulate (*Khachiyan*) as an SDP, if we write its quadratic constraints as

$$\begin{pmatrix} x_i & x_{i+1} \\ x_{i+1} & 1 \end{pmatrix} \succeq 0 \text{ for } i = 1, \ldots, m-1. \tag{3.1}$$

We see that $x_1 \geq 2^{2^{m-1}}$, hence the size of x_1, and of any feasible solution of (*Khachiyan*), is at least 2^{m-1}.

We show the feasible set of (*Khachiyan*) with $m = 3$ on the left in Figure 3.1. For better visibility, we replaced the constraint $x_3 \geq 2$ by $2 \geq x_3 \geq 0$ and made x_3 increase from right to left.

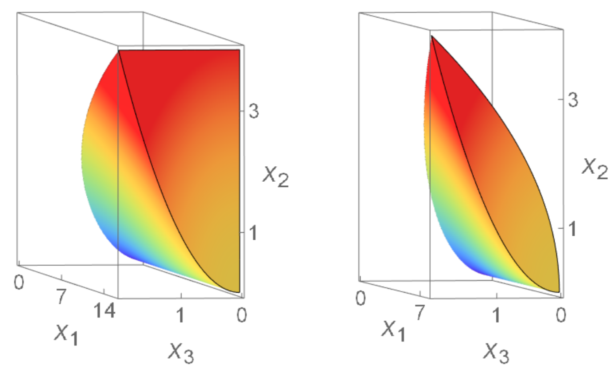


Figure 3.1: Feasible sets of (*Khachiyan*) (on the left) and of the quadratic inequalities (3.13) derived from (*Mild-SDP*) (on the right)

Exponential size solutions in SDPs are mathematically intriguing, and greatly complicate approaches to the following fundamental open problem:

Can we decide feasibility of (P) in polynomial time?

Indeed, algorithms that decide feasibility of (P) in polynomial time must assume that a polynomial size solution exists (if there is one): see a detailed exposition in [17]. In contrast, the algorithm in [53] that achieves the best known complexity bound for SDP feasibility uses a fundamental result from the first order theory of reals [58], and is polynomial only in fixed dimension.

We know of few papers that deal directly with the complexity of SDP. However, several works study the complexity of a related problem, optimizing a polynomial subject to polynomial inequality constraints. On the positive side, some polynomial optimization problems are polynomial time solvable when the dimension is fixed: see [58, 7, 8, 5, 66]. Further, polynomial size solutions exist in special cases [65]. On the other hand, several fundamental problems in polynomial optimization are NP-hard, see for example, [8, 42, 2, 3].

Khachiyan's example naturally leads us to the following questions:

$$\text{Are exponential size solutions common in SDPs?} \tag{3.2}$$

$$\text{Can we represent them in polynomial space?} \tag{3.3}$$

The answer to (3.2) seems to be a definite "no", for some of the following reasons:

- Exponential size solutions do not appear in typical SDPs in the literature.
- They can be eliminated even in (*Khachiyan*) by a fairly simple change. For example, let us add a new variable $x_{m+1},$ and change the last constraint to $x_m \geq 2 + x_{m+1}$; afterwards x_1 does not have to be large anymore.

  Or, let us replace x by $Gx,$ where G is a random, dense matrix; afterwards (*Khachiyan*) will be quite messy, and will have no variables that are obviously larger than others.

To question (3.3) we have hope to get a "yes" answer. After all, to convince ourselves that $x_1 := 2^{2^{m-1}}$ (with a suitable $x_2, \ldots, x_m$) is feasible in (*Khachiyan*), we do not need to explicity write it down, a symbolic computation suffices. Still, question (3.3) seems to be open.

Contributions It turns out that, despite these obstacles, we can still answer "yes" to question (3.2). One of the underlying techniques we use is facial reduction [11, 45, 47, 14, 68] that was originally introduced to induce strong duality in conic optimization problems.

We assume that (P) has a *strictly feasible* solution x for which $\sum_{i=1}^m x_i A_i + B$ is positive definite. We fix a nonnegative integer parameter $k,$ the *singularity degree* of a dual problem. We precisely define k soon, but for now we only need to know that $k \leq 1$ holds when (P) is a linear program. An informal version of our main result follows.

Informal Theorem 5 *After a linear change of variables $x \leftarrow Mx$, where M is a suitable invertible matrix, the leading k variables in strictly feasible solutions of (P) for which x_k is sufficiently large obey a Khachiyan type hierarchy. Namely, the inequalities*

$$x_1 \geq d_2 x_2^{\alpha_2},\, x_2 \geq d_3 x_3^{\alpha_3}, \ldots, x_{k-1} \geq d_k x_k^{\alpha_k} \tag{3.4}$$

hold, where

$$2 \geq \alpha_j \geq 1 + \frac{1}{k-j+1} \text{ for } j = 2, \ldots, k. \tag{3.5}$$

Here the d_j and α_j are positive constants that depend on the A_i, on B, and the last $m-k$ variables, that we consider fixed. $\square$

Hence, if k is large and x_k is large then x_1 is larger than x_k.

How much larger? In the worst case, when $\alpha_j = 2$ for all j, like in (*Khachiyan*), x_1 is at least constant times $x_k^{2^{k-1}}$. In the best case, when $\alpha_j = 1 + \frac{1}{k-j+1}$ for all j, by an elementary calculation x_1 is at least constant times x_k^k. So even in this best case the magnitude of x_1 is exponentially larger than that of x_k.

Our assumptions are minimal. We assumed that (P) has a strictly feasible solution, and indeed there are semidefinite programs without strictly feasible solutions, with large singularity degree, and without large solutions: we discuss such an SDP after Example 3.1. Further, we need to focus on just a subset of variables and allow a linear change of variables.[1] Nevertheless, we show that in SDPs coming from minimizing a univariate polynomial, large variables appear naturally, without any change of variables. The same is true of an SDP published in [40] that proves nonnegativity of a linear function over a set described by quadratic constraints.

We also partially answer the representation question (3.3). We show that in strictly feasible SDPs, after the change of variables $x \leftarrow Mx$, we can verify that a strictly feasible x exists, without even computing the values of the "large" variables $x_1, \ldots, x_k$. The same is true of SDPs coming from minimizing a univariate polynomial; in the latter SDPs we do not even need a change of variables.

[1] Since a change of variables, say $x \leftarrow Gx$ ruins the structure even of the nicely structured (*Khachiyan*), we may need to perform the inverse operation $x \leftarrow G^{-1}x$.

Related work Linear programs can be solved in polynomial time, as it was first proved by Khachiyan [24]; see Grötschel, Lovász, and Schrijver [17] for an exposition that handles important details like the necessary accuracy. Other landmark polynomial time algorithms for linear programming were given by Karmarkar [23], Renegar [57], and Kojima et al [27].

On the other hand, to decide SDP feasibility in polynomial time, we must assume that there is a polynomial size solution (should there be a solution). We refer to [17] for such an algorithm based on the ellipsoid method. The algorithm of Porkolab and Khachiyan [53] is the fastest known algorithm to decide SDP feasibility; however, it runs in polynomial time only for fixed n and m. The algorithm of [53] uses a foundational result of Renegar [58], which decides in polynomial time the feasibility of a system of polynomial inequalities in fixed dimension. We further refer to Nesterov and Nemirovskii [38] for foundational interior point methods to solve SDPs with an objective function. We also refer to Renegar [61] for a very clean treatment of interior point methods for convex optimization.

The complexity of SDP is closely related to the complexity of optimizing a polynomial subject to polynomial inequality constraints. To explain how, first consider a system of convex quadratic inequalities

$$x^\top Q_i x + b_i^\top x + c_i \leq 0 \ (i = 1, \dots, m) \tag{3.6}$$

where the Q_i are fixed symmetric psd matrices, and $x \in \mathbb{R}^n$ is the vector of variables. The question whether we can decide feasibility of (3.6) in polynomial time is also fundamental, open, and, in a sense, easier than the question of deciding feasibility of (P) in polynomial time. The reason is that (3.6) can be represented as an instance of (P) by choosing suitable A_i and B matrices. On the other hand, we can formulate semidefiniteness of a symmetric matrix variable by requiring the principal minors (which are polynomials) to be nonnegative.

Among positive results in polynomial optimization, we already mentioned Renegar's paper [58]. Bienstock [7] proved that such problems can be solved in polynomial time, if the number of constraints is fixed, the constraints and objective are quadratic, and at least one constraint is strictly convex. The work of [7] builds on Barvinok's fundamental result [5] that proved we can test in polynomial time whether a system of a fixed number of quadratic equations is feasible. It also builds on early work of Vavasis [65] which proved that a system with linear constraints and one quadratic constraint has a solution of polynomial size. In other important early work, Vavasis and Zippel [66] proved we can solve indefinite quadratic optimization problems with a ball constraint, in polynomial time.

On the flip side, there are many hardness results. For example, Bienstock, del Pia, and Hildebrand [8] proved it is NP-hard to test whether a system of quadratic inequalities has a polynomial size rational solution, even if we know that the system has a rational solution. Pardalos and Vavasis [42] proved the fundamental problem of minimizing a (nonconvex) quadratic function subject to linear constraints is also NP-hard. The following problem is also classical, and was proven to be NP-hard only in 2013, by Ahmadi, Olshevsky, Parrilo, and Tsitsiklis [2]: can we test convexity of a polynomial? It is also NP-hard to test whether a polynomial optimization problem attains its optimal value, see Ahmadi and Zhang [3].

One of the tools we use is an elementary facial reduction algorithm. These algorithms were originally designed to ensure strong duality in conic optimization problems. They originated in the paper of Borwein and Wolkowicz [11], then simpler variants were given, for example, by Waki and Muramatsu [68] and in [45, 47]. For a recent comprehensive survey of facial reduction and its applications, see Drusvyatskiy and Wolkowicz [14].

In other related work, O' Donnell [40] presented an SDP that certifies nonnegativity of a polynomial via the sum-of-squares proof system, and is essentially equivalent to (*Khachiyan*). Previously it was thought that sum-of-squares proofs, a popular tool in theoretical computer science, can be found in polynomial time. However, due to this work, it is now clear that this is not obviously the case. Precisely, the complexity of finding SOS proofs is just as open as the complexity of deciding feasibility of SDPs.

The plan of the paper In Subsection 3.1.1 we review preliminaries. In Subsection 3.2.1 we formally state Theorem 5 and illustrate it via two extreme examples. In Subsection 3.2.2 we prove it in a sequence of lemmas. In particular, in Lemma 13 we give a recursive formula, akin to a continued fractions formula, to compute the α_j exponents in (3.4). As an alternative, in Subsection 3.2.3 we show how to compute the α_j using the classical Fourier-Motzkin elimination for linear inequalities; this is an interesting contrast with SDPs being highly nonlinear. In Section

3.3 we cover the case of SDPs coming from polynomial optimization and also revisit the example from [40]. Section 3.4 concludes with a discusion.

Our proofs are fairly elementary. We use Proposition 1, a convex analysis argument about positive semidefinite matrices and linear subspaces, but other than that, we only rely on basic linear algebra, and on manipulating quadratic polynomials.

3.1.1 Notation and preliminaries

Matrices Given a matrix $M \in \mathbb{R}^{n\times n}$ and $R, S \subseteq \{1,\ldots,n\}$ we denote the submatrix of M corresponding to rows in R and columns in S by $M(R,S)$. We write $M(R)$ to abbreviate $M(R,R)$.

We let $\mathcal{S}^n$ be the set of $n\times n$ symmetric matrices and $\mathcal{S}^n_+$ be the set of $n\times n$ symmetric positive semidefinite (psd) matrices. The notation $S \succ 0$ means that the symmetric matrix S is positive definite. The inner product of symmetric matrices S and T is defined as $S \bullet T := \operatorname{trace}(ST)$.

Definition 3.1. We say that $(C_1,\ldots,C_\ell)$ is a *regular facial reduction sequence for $\mathcal{S}^n_+$* if each C_i is in $\mathcal{S}^n$ and of the form

$$C_1 = \begin{array}{c} \begin{array}{cc} \overbrace{}^{r_1} & \overbrace{}^{n-r_1} \end{array} \\ \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \end{array}, \ldots, C_i = \begin{array}{c} \begin{array}{ccc} \overbrace{}^{r_1+\ldots+r_{i-1}} & \overbrace{}^{r_i} & \overbrace{}^{n-r_1-\ldots-r_i} \end{array} \\ \begin{pmatrix} \times & \times & \times \\ \times & I & 0 \\ \times & 0 & 0 \end{pmatrix} \end{array}$$

for $i = 1,\ldots,\ell,$ where the r_i are nonnegative integers, and the $\times$ symbols correspond to blocks with arbitrary elements.

These sequences appear in facial reduction algorithms. The term "facial reduction" reflects the following: given a psd matrix Y which has zero $\bullet$ product with $C_1,\ldots,C_\ell$ we see that the first r_1 diagonal elements of Y are zero (by $C_1 \bullet Y = 0$), hence the first r_1 rows and columns are zero.

Continuing, we deduce that the first $r_1 + \cdots + r_\ell$ rows and columns of Y are zero, so overall Y is reduced to live in a face of $\mathcal{S}^n_+$. [2]

Next we formalize what we mean by "replacing x by Mx for some invertible matrix M in (P)."

Definition 3.2. We say that we *reformulate* (P) if we apply to it some of the following operations (in any order):

1. Exchange A_i and A_j, where i and j are distinct indices in $\{1, \ldots, m\}$.

2. Replace A_i by $\lambda A_i + \mu A_j$, where λ and μ are reals, and $\lambda \neq 0$.

3. Replace all A_i by $T^\top A_i T$ and B by $T^\top B T$, where T is a suitably chosen invertible matrix.

We also say that by reformulating (P) we obtain a *reformulation.*

We see that operations (1) and (2) amount to performing elementary row operations on a dual type system, say, on

$$A_i \bullet Y = 0 \text{ for } i = 1, \ldots, m.$$

Since operations (1) and (2) can be encoded by an invertible matrix, they amount to replacing the variable x by Mx in (P), where M is some invertible matrix. As to operation (3), it does not influence the magnitude of the x_i and we only use it to put (P) into a more convenient looking form.

Reformulations were previously used to study various pathologies in SDPs, for example, unattained optimal values and duality gaps [48]; and infeasibility [30]. In this work we show that they help us understand another classical pathology, exponential size solutions.

We will rely on the following statement about the connection of $\mathcal{S}^n_+$ and a linear subspace.

Proposition 1. *Suppose L is a linear subspace of $\mathcal{S}^n$. Then exactly one of the followig two alternatives is true:*

1. *There is a nonzero positive semidefinite matrix in L.*

2. *There is a positive definite matrix in $L^\perp$.*

[2] A convex subset F of $\mathcal{S}^n_+$ is a face, if for any $X, Y \in \mathcal{S}^n_+$ if the open line segment $\{\lambda X + (1-\lambda) Y : 0 < \lambda < 1\}$ intersects F, then both X and Y must be in F.

3.2 Main results and proofs

3.2.1 Reformulating (P) and statement of Theorem 5

In our first lemma we present an algorithm to reformulate (P) into a more convenient looking form. The algorithm is a simplified version of the algorithm in [30]; the latter, in turn, is a specialized facial reduction algorithm.

Lemma 9. *The problem (P) has a reformulation*

$$x_1 A_1' + \cdots + x_k A_k' + x_{k+1} A_{k+1}' + \cdots + x_m A_m' + B' \succeq 0 \tag{P'}$$

with the following properties:

- *k is a nonnegative integer, and $(A_1', \ldots, A_k')$ is a regular facial reduction sequence.*
- *If $r_1, \ldots, r_k$ is the size of the identity block in $A_1', \ldots, A_k'$, respectively, then $n - r_1 - \cdots - r_k$ is the maximum rank of a matrix in*

$$\{\, Y \succeq 0 \,|\, A_i \bullet Y = 0 \; for \; i = 1, \ldots, m \,\}. \tag{3.7}$$

Proof. Let L be the linear span of $A_1, \ldots, A_m$ and apply Proposition 1. If item (2) holds we let $k = 0$, $A_i' = A_i$ for all i, $B' = B$ and stop.

If item (1) holds, we choose a nonzero psd matrix $V = \sum_{i=1}^m \lambda_i A_i$ in L and assume $\lambda_1 \neq 0$ without loss of generality. We then choose a T invertible matrix so that

$$T^\top V T = \begin{pmatrix} I_{r_1} & 0 \\ 0 & 0 \end{pmatrix},$$

where r_1 is the rank of V. We let $A_1' := T^\top V T, A_i' := T^\top A_i T$ for $i \geq 2$, and $B' = T^\top B T$.

Let r be the maximum rank of a psd matrix in $L^\perp$ (i.e., in (3.7)). Also, let $L_{\rm new}$ be the linear span of $A_1', \ldots, A_m'$. We claim that r is also the maximum rank of a psd matrix in $L_{new}^\perp$. For that, suppose $Y \succeq 0$ is in $L^\perp$. Then

$$A_i \bullet Y \;=\; T^\top A_i T \bullet T^{-1} Y T^{-\top} \;=\; 0,$$

so $T^{-1}YT^{-\top}$ is in $L_{new}^\perp$ and has the same rank as Y. Similarly, from any psd matrix in $L_{\rm new}^\perp$ we can construct a psd matrix in $L^\perp$ with the same rank. This proves our claim.

We see that if $Y \in L_{\rm new}^\perp \cap \mathcal{S}_+^n$ then $A_1' \bullet Y = 0$ so the sum of the first r_1 diagonal elements of Y is zero, hence the first r_1 rows and columns of Y are zero.

We next construct an SDP

$$\sum_{i=2}^m x_i F_i + G \succeq 0,$$

where F_i is obtained from A_i' by deleting the first r_1 rows and columns for $i = 2, \dots, m$ and G is obtained from B' in the same manner. By the above argument the maximum rank of a matrix in $\{Z \succeq 0 : F_i \bullet Z = 0 \, (i = 2, \dots, m)\}$ is also r, so we can proceed in a similar manner with this smaller SDP. When our process stops, we have the required reformulation. $\square$

From now on we assume that

$$\boxed{k \text{ is the smallest integer that satisfies the requirements of Lemma 9.}}$$

Using the terminology of facial reduction, k is the *singularity degree of* the system (3.7). This concept was originally introduced by Sturm in [63] and used to derive error bounds, namely, bounds on the distance of a point from the feasible set of an SDP. For a broad generalization of Sturm's result to conic systems over so-called amenable cones, see a recent result by Lourenço [33].

Definition 3.3. We say that $(\bar{x}_{k+1}, \dots, \bar{x}_m)$ is *partially strictly feasible* in (P') if there is $(x_1, \dots, x_k)$ such that $(x_1, \dots, x_k, \bar{x}_{k+1}, \dots, \bar{x}_m)$ is strictly feasible in it.

For the rest of the paper we fix

$$\boxed{(\bar{x}_{k+1}, \dots, \bar{x}_m) \text{ a partially strictly feasible solution in } (P').}$$

From now on we will say that a number is a *constant*, if it depends only on the $\bar{x}_i$, A_i and B. Theorem 5 will rely on such constants.

We now formally state our main result.

Theorem 5. *There is* $(x_1, \dots, x_k)$ *such that* $(x_1, \dots, x_k, \bar{x}_{k+1}, \dots, \bar{x}_m)$ *is strictly feasible in* (P') *and* x_k *is arbitrarily large.*

Further, if x_k *is sufficiently large and* $(x_1, \dots, x_k, \bar{x}_{k+1}, \dots, \bar{x}_m)$ *is strictly feasible in* (P') *then*

$$x_j \quad \geq \quad d_{j+1} x_{j+1}^{\alpha_{j+1}} \, \textit{for} \ j = 1, \dots, k-1, \tag{3.8}$$

where

$$2 \geq \alpha_{j+1} \geq 1 + \frac{1}{k-j} \ \textit{for}\, j = 1, \dots, k-1. \tag{3.9}$$

Here the d_j *and* α_j *are positive constants.*

$\square$

The proof of Theorem 5 has three main parts. First, in Lemma 10 we prove the first statement, that in strictly feasible solutions of (P') we can have arbitrarily large x_k. Lemma 11 is a technical statement about a certain parameter, the *tail-index* of the A_j'.

In the second part, Lemma 12 deduces from (P') a set of quadratic polynomial inequalities. These are typically "messy", namely they look like

$$(x_1 + x_2 + x_3)(x_4 + 10x_5) > (x_2 - 3x_4)^2.$$

Third, in Lemma 13 from these messy inequalities we first derive "cleaned up" versions, such as

$$x_1 x_4 > \text{constant}\, x_2^2,$$

then from these cleaned up inequalities we deduce the inequalities (3.8) and a recursive formula to compute the α_j. Next, Lemma 14 proves that the α_j exponents are a monotone function of the tail-index of the A_j'. Finally, Lemma 15 shows that a minimal tail-index gives the smallest possible exponent α_j. Combining all lemmas gives us Theorem 5.

Before we get to the proof, we illlustrate Theorem 5 via two extreme examples. Although Theorem 5 is about strictly feasible solutions, the examples are simple, and in all of them we just look at feasible solutions.

Example 3.1. (Khachiyan SDP) Consider the SDP

$$\begin{pmatrix} \boldsymbol{x_1} & & & & \boldsymbol{x_2} \\ & \boldsymbol{x_2} & & & \boldsymbol{x_3} \\ & & \boldsymbol{x_3} & & \boldsymbol{x_4} \\ & & & x_4 & \\ \boldsymbol{x_2} & \boldsymbol{x_3} & \boldsymbol{x_4} & & 1 \end{pmatrix} \succeq 0, \tag{Kh-SDP}$$

which can be written in the form of (P') with the A_i' matrices given below and B' the matrix whose lower right corner is 1 and the remaining elements are zero:

$$\underbrace{\begin{pmatrix} 1 & & & & \\ & 0 & & & \\ & & 0 & & \\ & & & 0 & \\ & & & & 0 \end{pmatrix}}_{A_1'}, \underbrace{\begin{pmatrix} 0 & & & & 1 \\ & 1 & & & \\ & & 0 & & \\ & & & 0 & \\ 1 & & & & 0 \end{pmatrix}}_{A_2'}, \underbrace{\begin{pmatrix} 0 & & & & \\ & 0 & & & 1 \\ & & 1 & & \\ & & & 0 & \\ & 1 & & & 0 \end{pmatrix}}_{A_3'}, \underbrace{\begin{pmatrix} 0 & & & & \\ & 0 & & & \\ & & 0 & & 1 \\ & & & 1 & \\ & & 1 & & 0 \end{pmatrix}}_{A_4'}.$$

The subdeterminants in (Kh-SDP) with three red, three blue, and three green corners, respectively, give the inequalities

$$x_1 \geq x_2^2,\ x_2 \geq x_3^2,\ x_3 \geq x_4^2 \tag{3.10}$$

that appear in (*Khachiyan*). (For simplicity we left out the inequality $x_4 \geq 2$). $\Diamond$

We note in passing that the feasible sets of (Kh-SDP) and of the derived quadratic inequalities (3.10) are not equal. For example $x = (256, 16, 4, 2)$ is not feasible in (Kh-SDP), but is feasible in (3.10). However, we can easily construct an SDP that exactly represents (*Khachiyan*), as follows. We define a $(2m-1) \times (2m-1)$ matrix whose first $m-1$ order two principal minors are of the form (3.1) and the last order one principal minor represents the constraint $x_m \geq 2$. Permuting rows and columns puts this exact SDP into the regular form of (P').

We next note that the singularity degree of $\{Y \succeq 0 : A_i' \bullet Y = 0 \,\text{for}\; i = 1, \dots, 4\}$ is four. Indeed, consider a regular facial reduction sequence, say $\hat{A}_1, \hat{A}_2, \dots$ whose members are in the linear span of the A_i'. Suppose without loss of generality that $\hat{A}_1 \neq 0$. Then $\hat{A}_1 = A_1'$, since A_1' is the only nonzero psd matrix in the linear span of the A_i'. Similarly, asume without loss of generality that the lower right 3×3 block of $\hat{A}_2$ is nonzero. Then $\hat{A}_2 = A_2'$, and so on.

We finally discuss whether we need to assume that a strictly feasible solution exists, in order to derive Theorem 5. On the one hand, there are semidefinite programs which have no strictly feasible solutions, nor do they exhibit the hierarchy among the leading variables seen in (3.5). Suppose indeed that in (Kh-SDP) we change x_1 to $x_1 + 1$ and the 1 entry in the bottom right corner to 0. This change does not affect the parameter k. Further, the new SDP is no longer strictly feasible, and $x_2 = x_3 = x_4 = 0$ holds in any feasible solution, but x_1 can be -1 [3].

On the other hand, there are SDPs with no strictly feasible solution, which, however, have large size solutions: to produce such a problem, we simply take any SDP that has large solutions, and add all-zero rows and columns.

Example 3.2. (Mild SDP) As a counterpoint to (Kh-SDP) we next consider a mild SDP (we will see soon why we call it "mild")

$$\begin{pmatrix} \boldsymbol{x_1} & & \boldsymbol{x_2} & & \\ & \boldsymbol{x_2} & & \boldsymbol{x_3} & \\ \boldsymbol{x_2} & & \boldsymbol{x_3} & & \boldsymbol{x_4} \\ & \boldsymbol{x_3} & & \mathbf{x_4} & \\ & & \boldsymbol{x_4} & & \mathbf{1} \end{pmatrix} \succeq 0. \qquad (Mild\text{-}SDP)$$

[3] We can similarly create such an SDP with any number of variables.

We write $(Mild\text{-}SDP)$ in the form of (P') with the A_i' matrices shown below and B' the matrix whose lower right corner is 1 and the remaining elements are zero:

$$\underbrace{\begin{pmatrix} 1 & & & & \\ & 0 & & & \\ & & 0 & & \\ & & & 0 & \\ & & & & 0 \end{pmatrix}}_{A_1'}, \underbrace{\begin{pmatrix} 0 & & 1 & & \\ & 1 & & & \\ 1 & & 0 & & \\ & & & 0 & \\ & & & & 0 \end{pmatrix}}_{A_2'}, \underbrace{\begin{pmatrix} 0 & & & & \\ & 0 & & 1 & \\ & & 1 & & \\ & 1 & & 0 & \\ & & & & 0 \end{pmatrix}}_{A_3'}, \underbrace{\begin{pmatrix} 0 & & & & \\ & 0 & & & \\ & & 0 & & 1 \\ & & & 1 & \\ & & 1 & & 0 \end{pmatrix}}_{A_4'}.$$

In $(Mild\text{-}SDP)$ the subdeterminants with three red, three blue, and three green corners, respectively, yield the inequalities

$$x_1 x_3 \geq x_2^2,\ x_2 x_4 \geq x_3^2,\ x_3 \geq x_4^2. \tag{3.11}$$

Next from (3.11) we derive the inequalities

$$x_1 \geq x_2^{4/3},\ x_2 \geq x_3^{3/2},\ x_3 \geq x_4^2 \tag{3.12}$$

as follows. The last inequality $x_3 \geq x_4^2$ is copied from (3.11) to (3.12) only for completeness. Next we plug $x_3^{1/2} \geq x_4$ into the middle inequality in (3.11) to get $x_2 \geq x_3^{3/2}$. We finally use this last inequality in the first one in (3.11) and deduce $x_1 \geq x_2^{4/3}$.

To summarize, the exponents in the derived inequalities (3.12) are the smallest permitted by our bounds (3.9).

$\Diamond$

To illustrate the difference between (*Khachiyan*) and the inequalities derived from $(Mild\text{-}SDP)$, we show the set defined by the inequalities

$$x_1 x_3 \geq x_2^2,\ x_2 \geq x_3^2,\ 2 \geq x_3 \geq 0 \tag{3.13}$$

on the right in Figure 3.1. Note that the set defined by (3.13) is a three dimensional version of the set given in (3.11), normalized by adding upper and lower bounds on x_3.

3.2.2 Proof of Theorem 5

In Lemmas 10–12 we will use the following notation:

$$\begin{aligned}
r_j &= \text{size of the identity block in } A'_j \text{ for } j = 1, \ldots, k, \\
\mathcal{I}_1 &:= \{1, \ldots, r_1\}, \\
\mathcal{I}_2 &:= \{r_1 + 1, \ldots, r_1 + r_2\}, \\
&\vdots \\
\mathcal{I}_k &:= \{r_1 + \cdots + r_{k-1} + 1, \ldots, r_1 + \cdots + r_k\}, \\
\mathcal{I}_{k+1} &:= \{r_1 + \cdots + r_k + 1, \ldots, n\}.
\end{aligned} \tag{3.14}$$

Lemma 10. *There is $(x_1, \ldots, x_k)$ such that x_k is arbitrarily large, and $(x_1, \ldots, x_k, \bar{x}_{k+1}, \ldots, \bar{x}_m)$ is strictly feasible in (P').*

Proof. Let

$$Z := \sum_{i=k+1}^{m} \bar{x}_i A'_i + B'.$$

Since there is $x_1, \ldots, x_k$ such that $\sum_{i=1}^{k} x_i A'_i + Z \succ 0,$ and $A'_i(\mathcal{I}_{k+1}) = 0$ for $i = 1, \ldots, k$ we see that

$$Z(\mathcal{I}_{k+1}) \succ 0.$$

By the definition of positive definiteness (G is positive definite if $x^\top G x > 0$ for all nonzero x), and by the shape of A'_k, we see that the $\mathcal{I}_k \cup \mathcal{I}_{k+1}$ diagonal block of $x_k A'_k + Z$ is positive definite when x_k is large enough. For any such x_k there is x_{k-1} so the $\mathcal{I}_{k-1} \cup \mathcal{I}_k \cup \mathcal{I}_{k+1}$ diagonal block of $x_{k-1} A'_{k-1} + x_k A'_k + Z$ is positive definite. We construct $x_{k-2}, \ldots, x_1$ in a similar manner. $\square$

The proof of Lemma 10 partially answers the representation question (3.3). In particular, for the moment let us ignore the requirement that we need to choose x_k to be large and just focus on completing $(\bar{x}_{k+1}, \ldots, \bar{x}_m)$ to a strictly feasible solution. The proof that the required $(x_1, \ldots, x_k)$

could be computed is fairly simple, and it is illustrated on Figure 3.2, where the red blocks stand for the larger and larger blocks that we make positive definite. So we can convince ourselves that $(x_1, \ldots, x_k)$ *exist*, even without computing their actual values.

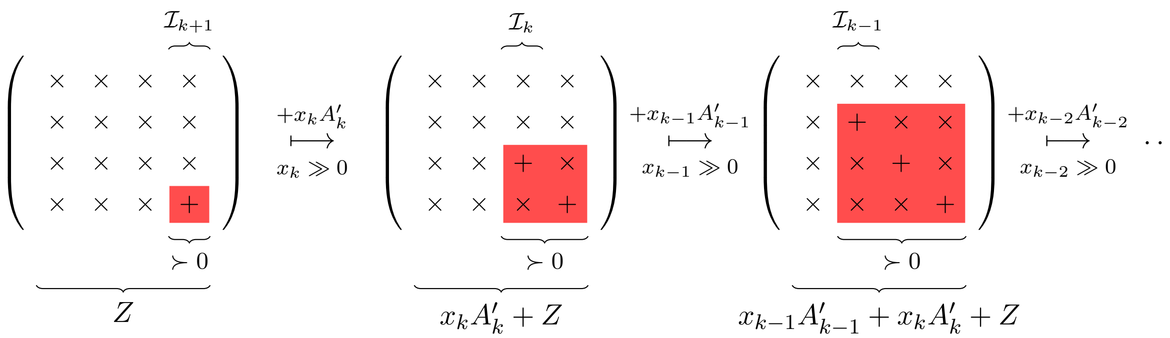


Figure 3.2: Verifying that $x_1, \ldots, x_k$ exist, without computing them

From now on we will assume

$$r_1 + \cdots + r_k < n, \tag{3.15}$$

and we claim that we can do so without loss of generality. Indeed, suppose that the sum of the r_j is n. Then an argument like in the proof of Lemma 10 proves that $A'_1, \ldots, A'_k$ have a positive definite linear combination. Hence the singularity degree of (3.7) is actually just 1, so Theorem 5 holds vacuously.

By (3.15) we see that $\mathcal{I}_{k+1} \neq \emptyset$.

To motivate our next definition we compare our two extreme examples from two viewpoints. From the first viewpoint we see that in (Kh-SDP) the x_j variables in the offdiagonal positions are more to the right than in ($Mild$-SDP). From the second viewpoint, in the inequalities (3.10) derived from (Kh-SDP) the exponents are larger than in the inequalities (3.12) derived from ($Mild$-SDP). We will see that these two facts are closely connected, so in the next definition we capture "how far to the right the x_j are in off-diagonal positions."

Definition 3.4. The *tail-index* of A'_{j+1} is

$$t_{j+1} := \max \{ t \, : \, A'_{j+1}(\mathcal{I}_j, \mathcal{I}_t) \neq 0 \} \text{ for } j = 1, \ldots, k-1. \tag{3.16}$$

$$
\begin{array}{rl}
 & \begin{array}{ccccccc} & \mathcal{I}_j & \mathcal{I}_{j+1} & & \mathcal{I}_{t_{j+1}} & & \mathcal{I}_{k+1} \end{array} \\
A'_{j+1} = & \left(\begin{array}{c|c|c|c|c|c||c}
\times & \times & \times & \times & \times & \times & \times \\ \hline
\times & \times & \times & \times & \bullet & & \\ \hline
\times & \times & I & & & & \\ \hline
\times & \times & & & & & \\ \hline
\times & \bullet & & & & & \\ \hline
\times & & & & & & \\ \hline\hline
\times & & & & & &
\end{array}\right).
\end{array}
$$

Figure 3.3: The tail-index of A'_{j+1}

In words, t_{j+1} is the index of the rightmost nonzero block of columns "directly above" the identity block in A'_{j+1}. We illustrate the tail-index on Figure 3.3. Here and in later figures the $\bullet$ blocks are nonzero, and we separate the columns indexed by $\mathcal{I}_{k+1}$ from the other columns by double vertical lines.

Continuing our examples, we see that $t_2 = t_3 = t_4 = 5$ in (Kh-SDP), whereas $t_2 = 3$, $t_3 = 4$, $t_4 = 5$ in ($Mild$-SDP).

Lemma 11.

$$t_{j+1} > j + 1 \textit{ for } j = 1, \ldots, k - 1.$$

Proof. We will use the following notation: for $r, s \in \{1, \ldots, k+1\}$ such that $r \leq s$ we let

$$\mathcal{I}_{r:s} := \mathcal{I}_r \cup \cdots \cup \mathcal{I}_s. \tag{3.17}$$

Let $j \in \{1, \ldots, k-1\}$ be arbitrary. To help with the proof, we picture A'_j and A'_{j+1} in equation (3.18). As always, the empty blocks are zero, and the $\times$ blocks are arbitrary. The blocks marked by $\otimes$ are $A'_{j+1}(\mathcal{I}_j, \mathcal{I}_{(j+2):(k+1)})$ and its symmetric counterpart. We will prove that these blocks are nonzero.

$$
\begin{array}{rl}
 & \begin{array}{cccc} \mathcal{I}_{1:(j-1)} & \mathcal{I}_j & \mathcal{I}_{j+1} & \mathcal{I}_{(j+2):(k+1)} \end{array} \\
A'_j = & \left(\begin{array}{c|c|c|c}
\times & \times & \times & \times \\ \hline
\times & I & & \\ \hline
\times & & & \\ \hline
\times & & &
\end{array}\right)
\end{array},
\quad
\begin{array}{rl}
 & \begin{array}{cccc} \mathcal{I}_{1:(j-1)} & \mathcal{I}_j & \mathcal{I}_{j+1} & \mathcal{I}_{(j+2):(k+1)} \end{array} \\
A'_{j+1} = & \left(\begin{array}{c|c|c|c}
\times & \times & \times & \times \\ \hline
\times & \times & \times & \otimes \\ \hline
\times & \times & I & \\ \hline
\times & \otimes & &
\end{array}\right)
\end{array}. \tag{3.18}
$$

For that, suppose the $\otimes$ blocks are zero and let $A'_j := \lambda A'_j + A'_{j+1}$ for some large $\lambda > 0$. Then by the definition of positive definiteness (G is positive definite if $x^T G x > 0$ for all nonzero x) we find

$$A'_j(\mathcal{I}_{j:(j+1)}) \succ 0.$$

Let Q be a matrix of suitable scaled eigenvectors of $A'_j(\mathcal{I}_{j:(j+1)})$, define

$$T := \begin{array}{c} \begin{array}{ccc} \overbrace{\qquad}^{\mathcal{I}_{1:(j-1)}} & \overbrace{\qquad}^{\mathcal{I}_{j:(j+1)}} & \overbrace{\qquad\quad}^{\mathcal{I}_{(j+2):(k+1)}} \end{array} \\ \left(\begin{array}{c|c|c} I & & \\ \hline & Q & \\ \hline & & I \end{array}\right) \end{array},$$

and let

$$A'_i := T^\top A_i T \text{ for } i = 1, \ldots, j, j+2, \ldots, k.$$

Then an elementary calculation shows that $(A'_1, \ldots, A'_j, A'_{j+2}, \ldots, A'_k)$ is a length $k-1$ regular facial reduction sequence that satisfies the requirements of Lemma 9. However, we assumed that the shortest such sequence has length k. This contradiction completes the proof. $\square$

In Lemma 12 we construct a sequence of polynomial inequalities that must be satisfied by any $(x_1, \ldots, x_k)$ that complete $(\bar{x}_{k+1}, \ldots, \bar{x}_m)$ to a strictly feasible solution. We need some more notation. Given a strictly feasible solution

$$(x_1, \ldots, x_k, \bar{x}_{k+1}, \ldots, \bar{x}_m)$$

we will write δ_j for an affine combination of the "x" and "$\bar{x}$" terms with indices larger than j, in other words,

$$\delta_j = \gamma_{j+1} x_{j+1} + \cdots + \gamma_k x_k + \gamma_{k+1} \bar{x}_{k+1} + \cdots + \gamma_m \bar{x}_m + \gamma_{m+1} \tag{3.19}$$

where the γ_i are constants.

We will actually slightly abuse this notation. We will write δ_j more than once, but we may mean a different affine combination each time. For example, if $k = m = 4$, then we may write

$\delta_2 = 2x_3 + 3x_4 + 5$ on one line, and $\delta_2 = x_3 - 2x_4 - 3$ on another. Given that $\bar{x}_{k+1}, \ldots, \bar{x}_m$ are fixed, δ_k will always denote a constant.

Lemma 12. *Suppose that $(x_1, \ldots, x_k, \bar{x}_{k+1}, \ldots, \bar{x}_m)$ is strictly feasible in (P'). Then*

$$p_j(x_1, \ldots, x_k) \, > \, 0 \text{ for } j = 1, \ldots, k-1, \tag{3.20}$$

for some p_j polynomials defined as follows:

- *if $t_{j+1} \leq k$, then we choose p_j as*

$$p_j(x_1, \ldots, x_k) \, = \, (x_j + \delta_j)(x_{t_{j+1}} + \delta_{t_{j+1}}) - (\beta_{j+1} x_{j+1} + \delta_{j+1})^2 \tag{3.21}$$

  where β_{j+1} is a nonzero constant. In this case we call p_j a type 1 polynomial.

- *if $t_{j+1} = k+1$, then we choose p_j as*

$$p_j(x_1, \ldots, x_k) \, = \, (x_j + \delta_j) - (\beta_{j+1} x_{j+1} + \delta_{j+1})^2, \tag{3.22}$$

  where β_{j+1} is a nonzero constant. In this case we call p_j a type 2 polynomial.

□

Before we prove it, we discuss Lemma 12. First we note that by Lemma 11 we have $t_k = k+1$ so p_{k-1} is always type 2.

In Khachiyan's example $(Khachiyan)$ all inequalities come from type 2 polynomials, namely from $x_j - x_{j+1}^2$ for $j = 1, \ldots, k-1$. In contrast, among the inequalities (3.11) derived from $(Mild\text{-}SDP)$ the first two come from type 1 polynomials and the last one from a type 2 polynomial.

Proof (of Lemma 12). Fix $j \in \{1, \ldots, k-1\}$. Let $\ell_1 \in \mathcal{I}_j$ and $\ell_2 \in \mathcal{I}_{t_{j+1}}$ such that $(A'_{j+1})_{\ell_1, \ell_2} \neq 0$.

As stated, suppose that $(x_1, \ldots, x_k, \bar{x}_{k+1}, \ldots, \bar{x}_m)$ is strictly feasible in (P'). For brevity, define

$$S \, := \sum_{i=1}^{k} x_i A'_i + \sum_{i=k+1}^{m} \bar{x}_i A'_i + B'. \tag{3.23}$$

We distinguish two cases.

Case 1: Suppose $t_{j+1} \leq k$. Below we show the matrices that will be important in defining p_j :

$$
\underbrace{\begin{array}{c}
\begin{array}{cccccc} & \overbrace{}^{\mathcal{I}_j} & & \overbrace{}^{\mathcal{I}_{t_{j+1}}} & & \overbrace{}^{\mathcal{I}_{k+1}} \end{array} \\
\left(\begin{array}{c|c|c|c|c||c}
\times & \times & \times & \times & \times & \times \\ \hline
\times & I & & & & \\ \hline
\times & & & & & \\ \hline
\times & & & & & \\ \hline
\times & & & & & \\ \hline\hline
\times & & & & &
\end{array}\right)
\end{array}}_{A'_j},
\underbrace{\begin{array}{c}
\begin{array}{cccccc} & \overbrace{}^{\mathcal{I}_j} & & \overbrace{}^{\mathcal{I}_{t_{j+1}}} & & \overbrace{}^{\mathcal{I}_{k+1}} \end{array} \\
\left(\begin{array}{c|c|c|c|c||c}
\times & \times & \times & \times & \times & \times \\ \hline
\times & \times & \times & \bullet & & \\ \hline
\times & \times & \times & & & \\ \hline
\times & \bullet & & & & \\ \hline
\times & & & & & \\ \hline\hline
\times & & & & &
\end{array}\right)
\end{array}}_{A'_{j+1}},
$$

$$
\underbrace{\begin{array}{c}
\begin{array}{cccccc} & \overbrace{}^{\mathcal{I}_j} & & \overbrace{}^{\mathcal{I}_{t_{j+1}}} & & \overbrace{}^{\mathcal{I}_{k+1}} \end{array} \\
\left(\begin{array}{c|c|c|c|c||c}
\times & \times & \times & \times & \times & \times \\ \hline
\times & \times & \times & \times & \times & \times \\ \hline
\times & \times & \times & \times & \times & \times \\ \hline
\times & \times & \times & I & & \\ \hline
\times & \times & \times & & & \\ \hline\hline
\times & \times & \times & & &
\end{array}\right)
\end{array}}_{A'_{t_{j+1}}}. \tag{3.24}
$$

As usual, the empty blocks are zero, the $\times$ blocks may have arbitrary elements, and the $\bullet$ block is nonzero. (More precisely, $A'_{j+1}(\mathcal{I}_{j+1}) = I$, but we do not indicate this in equation (3.24), since the other entries suffice to derive the p_j polynomial.)

Define $\beta_{j+1} := (A'_{j+1})_{\ell_1,\ell_2}$. Let S' be the submatrix of S that contains rows and columns indexed by ℓ_1 and ℓ_2, then

$$S' = \begin{pmatrix} x_j + \delta_j & \beta_{j+1}x_{j+1} + \delta_{j+1} \\ \beta_{j+1}x_{j+1} + \delta_{j+1} & x_{t_{j+1}} + \delta_{t_{j+1}} \end{pmatrix}.$$

We define $p_j(x_1, \ldots, x_k)$ as the determinant of S', then p_j is a type 1 polynomial in the form (3.21). Since $S' \succ 0$, we see that $p_j(x_1, \ldots, x_k) > 0$ and the proof in this case is complete.

Case 2: Suppose $t_{j+1} = k+1$. Now p_j will mainly depend on two matrices that we show below:

$$\underbrace{\begin{array}{c} \begin{array}{cccc} \quad & \overbrace{}^{\mathcal{I}_j} & \quad & \overbrace{}^{\mathcal{I}_{t_{j+1}}} \end{array} \\ \left(\begin{array}{c|c|c||c} \times & \times & \times & \times \\ \hline \times & I & & \\ \hline \times & & & \\ \hline\hline \times & & & \end{array}\right) \end{array}}_{A'_j}, \underbrace{\begin{array}{c} \begin{array}{cccc} \quad & \overbrace{}^{\mathcal{I}_j} & \quad & \overbrace{}^{\mathcal{I}_{t_{j+1}}} \end{array} \\ \left(\begin{array}{c|c|c||c} \times & \times & \times & \times \\ \hline \times & \times & \times & \bullet \\ \hline \times & \times & \times & \\ \hline\hline \times & \bullet & & \end{array}\right) \end{array}}_{A'_{j+1}} \tag{3.25}$$

Again, the $\bullet$ blocks are nonzero. We again let S' be the submatrix of S that contains rows and columns indexed by ℓ_1 and ℓ_2.

Define $\lambda := (A'_{j+1})_{\ell_1,\ell_2}$, $\mu := S'_{\ell_2,\ell_2}$. Observe that μ depends only on $\bar{x}_{k+1}, \dots, \bar{x}_m$, the A'_i and B', in other words it is a constant. Then S' looks like

$$S' = \begin{pmatrix} x_j + \delta_j & \lambda x_{j+1} + \delta_{j+1} \\ \lambda x_{j+1} + \delta_{j+1} & \mu \end{pmatrix}.$$

Define

$$p_j(x_1, \dots, x_k) := \frac{1}{\mu} \det S'.$$

Since $S' \succ 0$ we have $\mu > 0$. So $p_j(x_1, \dots, x_k) > 0$ and $p_j(x_1, \dots, x_k)$ is a type 2 polynomial in the form required in (3.22) with $\beta_{j+1} = \lambda/\sqrt{\mu}$ (note that according to our definition of δ_j, if we divide it by a constant, the result is still δ_j). The proof in this case is now complete.

$\square$

Lemma 13. *Suppose that $(x_1, \dots, x_k, \bar{x}_{k+1}, \dots, \bar{x}_m)$ is strictly feasible in (P') and x_k is sufficiently large. Then*

$$x_j \geq d_{j+1} x_{j+1}^{\alpha_{j+1}} \textit{ for } j = 1, \dots, k-1, \tag{3.26}$$

where the d_{j+1} are positive constants and the α_{j+1} can be computed by the recursion

$$\alpha_{j+1} = \begin{cases} 2 - \dfrac{1}{\alpha_{j+2} \ldots \alpha_{t_{j+1}}} & \textit{if} \quad t_{j+1} \leq k \\ \qquad\qquad 2 & \textit{if} \quad t_{j+1} = k+1 \end{cases} \tag{3.27}$$

for $j = 1, \ldots, k-1$.

$\square$

Before proving it, we discuss Lemma 13. We have $t_k = k+1$ (by Lemma 11) hence Lemma 13 implies $\alpha_k = 2$. Hence, by induction the recursion (3.27) implies that $\alpha_j \in (1, 2]$ holds for all j. Thus, if x_k is large enough, then $x_j > 0$ for $j = 1, \ldots, k$.

It is also interesting to note that formula (3.27) is reminiscent of a continued fractions formula.

To illustrate Lemma 13 we show how from ($Mild$-SDP) we can deduce the inequalities (3.12) much more quickly than we did before. Precisely, we compute the exponents by the recursion (3.27) as

$$\begin{aligned} \alpha_4 &= 2 && \text{(since } t_4 = 5) \\ \alpha_3 &= 2 - 1/\alpha_4 = 3/2 && \text{(since } t_3 = 4) \\ \alpha_2 &= 2 - 1/\alpha_3 = 4/3 && \text{(since } t_2 = 3). \end{aligned} \tag{3.28}$$

The proof of Lemma 13 has two ingredients. First, from the messy looking type 1 inequalities (3.21) we deduce cleaned up versions

$$x_j x_{t_{j+1}} \geq \text{constant}\, x_{j+1}^2,$$

and we similarly clean up the type 2 inequalities (3.22). Then from the cleaned up inequalities we derive the required inequalities (3.26) and the recursion (3.27).

Since the proof of Lemma 13 is somewhat technical, we illustrate the cleaning up step with an example.

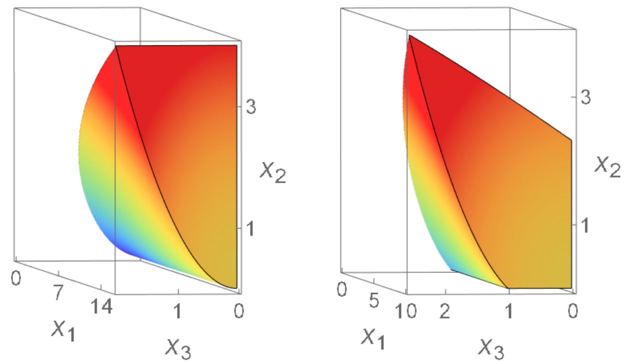


Figure 3.4: Feasible sets of (*Khachiyan*) (on left) and of inequalities derived from the perturbed Khachiyan SDP (3.29) (on the right)

Example 3.3. (Perturbed Khachiyan) Consider the SDP

$$\begin{pmatrix} x_1 - 2x_2 & & & x_2 - x_3 \\ & x_2 + x_3 & & x_3 \\ & & x_3 & \\ x_2 - x_3 & x_3 & & 1 \end{pmatrix} \succeq 0. \tag{3.29}$$

From its principal minors we deduce the inequalities

$$\begin{aligned} x_1 - 2x_2 &\geq (x_2 - x_3)^2 && (3.30) \\ x_2 + x_3 &\geq x_3^2. && (3.31) \end{aligned}$$

These two inequalities are a "perturbed" version of the inequalities in (*Khachiyan*), since we obtain them by replacing x_1 by $x_1 - 2x_2$ and x_2 by $x_2 \pm x_3$.

Suppose (x_1, x_2, x_3) is feasible in (3.29). Then the inequalities $x_1 \geq x_2^2$ and $x_2 \geq x_3^2$ no longer hold. However, assuming $x_3 \geq 10$, we claim that the following inequalities do:

$$\begin{aligned} x_1 &\geq \frac{1}{2}x_2^2 && (3.32) \\ x_2 &\geq \frac{1}{2}x_3^2. && (3.33) \end{aligned}$$

Indeed, (3.33) follows from (3.31) and $x_3 \geq 10$ directly. Using (3.33) we see that x_3 is lower order than x_2. A straightforward calculation shows

$$(x_2 - x_3)^2 \geq \frac{1}{2} x_2^2. \tag{3.34}$$

Hence from (3.30) we get

$$\begin{aligned} 0 &\leq x_1 - 2x_2 - (x_2 - x_3)^2 \\ &\leq x_1 - (x_2 - x_3)^2 \\ &\leq x_1 - \frac{1}{2} x_2^2, \end{aligned} \tag{3.35}$$

where the last inequality follows from using (3.34). Thus (3.32) follows, and the proof is complete.

$\Diamond$

We show the feasible set of (*Khachiyan*) (when $m = 3$) and the feasible set described by the inequalities (3.30) on Figure 3.4. From (*Khachiyan*) we removed the inequality $x_3 \geq 2$ and we normalized both sets by suitable bounds on x_3. Note that x_3 increases from right to left for better visibility.

Proof (of Lemma 13). We use an argument analogous to the one in Example 3.3. We use induction and show how to suppress the "δ" terms in the type 1 and type 2 polynomials at the cost of making x_k large and choosing suitable d_j constants.

Suppose that $(x_1, \ldots, x_k, \bar{x}_{k+1}, \ldots, \bar{x}_m)$ is strictly feasible in (P'). Then by Lemma 12 the inequalities $p_j(x_1, \ldots, x_k) > 0$ hold for $j = 1, \ldots, k-1$.

When $j = k-1$ we have $p_{k-1}(x_1, \ldots, x_k) = (x_{k-1} + \delta_{k-1}) - (\beta_k x_k + \delta_k)^2 > 0$. Using the definition of δ_{k-1} we get

$$(x_{k-1} + \gamma_k x_k + \delta_k) - (\beta_k x_k + \delta_k)^2 > 0,$$

where γ_k is a constant (possibly zero) and $\beta_k \neq 0$. Then for a suitable positive d_k we have $x_{k-1} \geq d_k x_k^2$ if x_k is sufficiently large.

For the inductive step we will adapt the O, Θ and o notation from theoretical computer science. Given functions $f, g : \mathbb{R}^k \to \mathbb{R}_+$ we say that

1. $f = O(g)$ (in words, f is big-Oh of g) if there are positive constants C_1 and C_2 such that for all $(x_1, \ldots, x_k)$ such that $(x_1, \ldots, x_k, \bar{x}_{k+1}, \ldots, \bar{x}_m)$ is strictly feasible in (P') and $x_k \geq C_1$ we have

$$f(x_1, \ldots, x_k) \leq C_2 g(x_1, \ldots, x_k).$$

2. $f = \Theta(g)$ (in words, f is big-Theta of g) if $f = O(g)$ and $g = O(f)$.

3. $f = o(g)$ (in words, f is little-oh of g) if for all $\epsilon > 0$ there is $\delta > 0$ such that if $(x_1, \ldots, x_k, \bar{x}_{k+1}, \ldots, \bar{x}_m)$ is strictly feasible in (P') and $x_k \geq \delta$ then

$$f(x_1, \ldots, x_k) \;\leq\; \epsilon g(x_1, \ldots, x_k).$$

The usual calculus of O, Θ and o carries over verbatim. For example

$$\begin{aligned} f = o(g) \quad &\Rightarrow \quad f = O(g) \\ f = O(g) \text{ and } h = O(g) \quad &\Rightarrow \quad f + h = O(g). \end{aligned} \tag{3.36}$$

Suppose next that $j+1 \leq k-1$ and we have proved the following: there are positive constants $d_{j+1}, \ldots, d_k$ and $\alpha_{j+2}, \ldots, \alpha_k$ derived from the recursion (3.27) such that the inequalities

$$\begin{aligned} x_{j+1} \;&\geq\; d_{j+2} x_{j+2}^{\alpha_{j+2}} \\ x_{j+2} \;&\geq\; d_{j+3} x_{j+3}^{\alpha_{j+3}} \\ &\;\vdots \\ x_{k-1} \;&\geq\; d_k x_k^{\alpha_k} \end{aligned} \tag{3.37}$$

hold for all $x_1, \ldots, x_k$ such that $(x_1, \ldots, x_k, \bar{x}_{k+1}, \ldots, \bar{x}_m)$ is strictly feasible in (P') and x_k is large enough.

We will construct positive constants d_{j+1}, and α_{j+1} according to the recursion (3.27) such that

$$x_j \;\geq\; d_{j+1} x_{j+1}^{\alpha_{j+1}} \tag{3.38}$$

holds for all $x_1, \ldots, x_k$ such that $(x_1, \ldots, x_k, \bar{x}_{k+1}, \ldots, \bar{x}_m)$ is strictly feasible in (P') and x_k is large enough.

We first observe that the recursion (3.27) implies $\alpha_{j+2}, \ldots, \alpha_k \in (1, 2]$, so by the inequalities (3.37) we have

$$x_s = o(x_\ell) \text{ when } s > \ell > j. \tag{3.39}$$

Assume that $(x_1, \ldots, x_k, \bar{x}_{k+1}, \ldots, \bar{x}_m)$ is strictly feasible in (P'). We distinguish two cases.

Case 1: First suppose that $t_{j+1} \leq k,$ in other words, the quadratic polynomial p_j is type 1 (see Lemma 12).

Then the inequality $p_j(x_1, \ldots, x_k) > 0$ implies

$$\begin{aligned}
0 &< (x_j + \delta_j)(x_{t_{j+1}} + \delta_{t_{j+1}}) - (\beta_{j+1}x_{j+1} + \delta_{j+1})^2 \\
&= (x_j + \gamma_{j+1}x_{j+1} + \delta_{j+1})(x_{t_{j+1}} + \delta_{t_{j+1}}) - (\beta_{j+1}x_{j+1} + \delta_{j+1})^2 \\
&= (x_j + \gamma_{j+1}x_{j+1} + o(x_{j+1}))(x_{t_{j+1}} + o(x_{t_{j+1}})) - (\beta_{j+1}x_{j+1} + o(x_{j+1}))^2 \\
&\leq (x_j + \Theta(x_{j+1}))\Theta(x_{t_{j+1}}) - \Theta(x_{j+1})^2
\end{aligned} \tag{3.40}$$

where γ_{j+1} is a constant (which may be zero). The first equality follows from the definition of δ_j (see 3.19). The second equality follows since by (3.39) and by $t_{j+1} > j + 1$ we have

$$|\delta_{j+1}| = o(x_{j+1}),\ |\delta_{t_{j+1}}| = o(x_{t_{j+1}}). \tag{3.41}$$

The last inequality follows from $\beta_{j+1} \neq 0$ and the calculus rules (3.36). We now continue (3.40):

$$\begin{aligned}
0 &< (x_j + \Theta(x_{j+1}))\Theta(x_{t_{j+1}}) - \Theta(x_{j+1})^2 \\
&= x_j\Theta(x_{t_{j+1}}) + \Theta(x_{j+1}x_{t_{j+1}}) - \Theta(x_{j+1})^2 \\
&\leq x_j\Theta(x_{t_{j+1}}) + o(x_{j+1}^2) - \Theta(x_{j+1})^2 \\
&\leq x_j\Theta(x_{t_{j+1}}) - \Theta(x_{j+1})^2,
\end{aligned} \tag{3.42}$$

where the second inequality follows, since $t_{j+1} > j + 1$ hence by (3.39) we have $x_{t_{j+1}} = o(x_{j+1})$. The last inequality follows from the calculus rules (3.36).

Next from the inequalities (3.37), using $t_{j+1} > j+1$ we learn that

$$x_{t_{j+1}}^{\alpha} = O(x_{j+1})$$

where $\alpha = \alpha_{j+2}\alpha_{j+3}\dots\alpha_{t_{j+1}}$. Hence $x_{t_{j+1}} = O(x_{j+1}^{1/\alpha})$.

We plug this last estimate into the last inequality in (3.42) and deduce

$$0 < x_j \Theta(x_{j+1}^{1/\alpha}) - \Theta(x_{j+1}^2).$$

Dividing by $x_{j+1}^{1/\alpha}$ and a constant, we see that $\alpha_{j+1} := 2 - 1/\alpha$ satisfies the recursion (3.27), as required.

Case 2 Suppose that $t_{j+1} = k+1$, in other words, the quadratic polynomial p_j is type 2. Then

$$\begin{aligned} p_j(x_1, \dots, x_k) &= (x_j + \delta_j) - (\beta_{j+1}x_{j+1} + \delta_{j+1})^2 \\ &= (x_j + \gamma_{j+1}x_{j+1} + \delta_{j+1}) - (\beta_{j+1}x_{j+1} + \delta_{j+1})^2 \end{aligned}$$

for some $\beta_{j+1} \neq 0$ and γ_{j+1} constants, where γ_{j+1} may be zero. By (3.39) we have $\delta_{j+1} = o(x_{j+1})$, hence

$$x_j \geq \Theta(x_{j+1}^2).$$

So we can set $\alpha_{j+1} = 2$, thereby completing the proof.

$\square$

As a prelude to Lemma 14, in Figure 3.5 we show three SDPs (for brevity we left out the $\succeq$ symbols). The first is (*Mild-SDP*). The second and third arise from it by shifting x_2 in the offdiagonal position to the right. Underneath we show the vector of the $\alpha = (\alpha_2, \alpha_3, \alpha_4)$ exponents in the inequalities derived by the recursion (3.27).

We see that α_2 increases from left to right and Lemma 14 presents a general result of this kind.

Lemma 14. *The α_j exponents in* (3.9) *are strictly increasing functions of the t_{j+1} tail-indices defined in Definition 3.4.*

$$
\underbrace{\begin{pmatrix} \boldsymbol{x_1} & & \boldsymbol{x_2} & & \\ & \boldsymbol{x_2} & & \boldsymbol{x_3} & \\ \boldsymbol{x_2} & & \boldsymbol{x_3} & & \boldsymbol{x_4} \\ & \boldsymbol{x_3} & & \boldsymbol{x_4} & \\ & & \boldsymbol{x_4} & & \mathbf{1} \end{pmatrix}}_{\boldsymbol{\alpha} = (\mathbf{4/3},\ \mathbf{3/2},\ \mathbf{2})} \to \underbrace{\begin{pmatrix} \boldsymbol{x_1} & & & \boldsymbol{x_2} & \\ & \boldsymbol{x_2} & & \boldsymbol{x_3} & \\ & & \boldsymbol{x_3} & & \boldsymbol{x_4} \\ \boldsymbol{x_2} & \boldsymbol{x_3} & & \boldsymbol{x_4} & \\ & & \boldsymbol{x_4} & & \mathbf{1} \end{pmatrix}}_{\boldsymbol{\alpha} = (\mathbf{5/3},\ \mathbf{3/2},\ \mathbf{2})} \to \underbrace{\begin{pmatrix} \boldsymbol{x_1} & & & & \boldsymbol{x_2} \\ & \boldsymbol{x_2} & & \boldsymbol{x_3} & \\ & & \boldsymbol{x_3} & & \boldsymbol{x_4} \\ & \boldsymbol{x_3} & & \boldsymbol{x_4} & \\ \boldsymbol{x_2} & & \boldsymbol{x_4} & & \mathbf{1} \end{pmatrix}}_{\boldsymbol{\alpha} = (\mathbf{2},\ \mathbf{3/2},\ \mathbf{2})}
$$

Figure 3.5: Shifting x_2 to the right increases α_2

Precisely, suppose we derived the inequalities

$$x_\ell \geq d_{\ell+1} x_{\ell+1}^{\alpha_{\ell+1}} \text{ for } \ell = 1, \ldots, k-1 \tag{3.43}$$

from (P') *using the recursion* (3.27).

Suppose also that $j \in \{1, \ldots, k-1\}$, $t_{j+1} \leq k$, *and we change* A'_{j+1} *so that* t_{j+1} *increases by* 1. *After the change we derive inequalities*

$$x_\ell \geq f_{\ell+1} x_{\ell+1}^{\omega_{\ell+1}} \text{ for } \ell = 1, \ldots, k-1, \tag{3.44}$$

using the recursion (3.27). *Here* $f_{\ell+1}$ *is a positive constant for all* ℓ.

Then

$$\omega_{\ell+1} \begin{cases} = \alpha_{\ell+1} & \text{if } \ell > j \\ > \alpha_{\ell+1} & \text{if } \ell = j \\ \geq \alpha_{\ell+1} & \text{if } \ell < j. \end{cases} \tag{3.45}$$

Proof. Recall from the proof of Lemma 12 that A'_{j+1} affects only polynomial p_j. Also recall from the proof of Lemma 13 that p_j does not affect inequalities (3.43) for $\ell > j$. So we conclude that $\omega_{\ell+1} = \alpha_{\ell+1}$ for all $\ell > j$.

We next prove $\omega_{j+1} > \alpha_{j+1}$. For brevity, let $s = t_{j+1}$ and recall that

$$\alpha_{j+1} = 2 - \frac{1}{\alpha},$$

where $\alpha = \alpha_{j+2} \cdot \cdots \cdot \alpha_s$.

We distinguish two cases. If $s < k,$ then formula (3.27) implies $\omega_{j+1} = 2 - 1/(\alpha \cdot \alpha_{s+1})$, hence $\omega_{j+1} > \alpha_{j+1},$ as wanted. If $s = k,$ then by the same formula $2 = \omega_{j+1}$ and $2 > \alpha_{j+1}$ so $\omega_{j+1} > \alpha_{j+1}$ again follows.

The remaining inequalities in (3.43) follow by induction using the recursion formula (3.27).

□

Lemma 15. *Suppose that* $t_{j+1} = j + 2$ *for* $j = 1, \ldots, k-1,$ *in other words,* t_{j+1} *is the smallest possible. Then in the inequalities* (3.37) *we have*

$$\alpha_{j+1} = 1 + \frac{1}{k-j} \text{ for } j = 1, \ldots, k-1. \tag{3.46}$$

Proof. We use induction. First suppose $j = k-1$. Since p_{k-1} is of type 2, we see $\alpha_{j+1} = \alpha_k = 2$, as wanted. Next assume that $1 \leq j < k-1$ and

$$\alpha_{j+2} = 1 + \frac{1}{k-j-1}.$$

By the recursion (3.27) we get

$$\alpha_{j+1} = 2 - \frac{1}{\alpha_{j+2}} = 1 + \frac{1}{k-j},$$

completing the proof. □

Proof (of Theorem 5). The result follows from Lemmas 10 through 15. Precisely, by Lemma 10 variable x_k can be arbitrarily large in a strictly feasible solution of (P'). By Lemma 12 we derive the polynomial inequalities (3.20). From these in Lemma 13 we derive the clean inequalities (3.26) via the recursion (3.27).

From the recursion (3.27) it directly follows that all α_{j+1} are at most 2. The lower bound on the α_{j+1} is proved as follows: by Lemma 14 the α_{j+1} are monotone functions of the tail-indices t_{j+1}. On the other hand $t_{j+1} \geq j+2$ for all j by Lemma 11 and when $t_{j+1} = j+2$ for all $j,$ then by Lemma 15 we have $\alpha_{j+1} = 1 + 1/(k-j)$. The proof is now complete. □

3.2.3 Computing the exponents by Fourier-Motzkin elimination

The recursion (3.27) gives a convenient way to compute the α_j exponents. Equivalently, we can compute the α_j via the well known Fourier-Motzkin elimination algorithm, designed for linear inequalities; this is an interesting contrast, since SDPs are highly nonlinear.

We do this as follows. If polynomial p_j is of type 1, then we suppress the lower order terms to get

$$x_j x_{t_{j+1}} \geq \text{constant}\, x_{j+1}^2, \tag{3.47}$$

see the last inequality in (3.42). If polynomial p_j is of type 2, then we similarly suppress the lower order terms to deduce

$$x_j \geq \text{constant}\, x_{j+1}^2. \tag{3.48}$$

After this, since $x_1, \ldots, x_k$ are all positive, we rewrite the inequalities in terms of $y_j := \log x_j$ for all j, then eliminate variables. For example, from the inequalites (3.11) we deduce

$$\begin{array}{rcl} y_1 + y_3 & \geq & 2y_2 \\ y_2 + y_4 & \geq & 2y_3 \\ y_3 & \geq & 2y_4. \end{array} \tag{3.49}$$

We add $\frac{1}{2}$ times the last inequality in (3.49) to the middle one to get

$$y_2 \geq \frac{3}{2} y_3. \tag{3.50}$$

We then add $\frac{2}{3}$ times (3.50) to the first inequality in (3.49) to get

$$y_1 \geq \frac{4}{3} y_2. \tag{3.51}$$

Finally, (3.50), (3.51) and the last inequality in (3.49) translate back to the inequalities (3.12).

3.3 When we do not even need a change of variables

As we previously discussed, the linear change of variables $x \leftarrow Mx$ is necessary to obtain a Khachiyan type hierarchy among the variables. Nevertheless, in this section we show a natural SDP in which large variables occur even without a change of variables; more precisely, the SDP is in the form of (P'). For completeness, we also revisit the example from [40], and show that the SDP therein is also in the regular form of (P').

Given a univariate polynomial of even degree $f(x) = \sum_{i=0}^{2n} a_i x^i$ we consider the problem of minimizing f over $\mathbb{R}$. We write this problem as

$$\begin{array}{rrcl} \sup & \lambda & & \\ s.t. & f - \lambda & \geq & 0. \end{array} \tag{3.52}$$

We will show that in the natural SDP formulation of (3.52) exponentially large variables appear naturally, although here by "exponentially large" we only mean in magnitude, not in size.

It is known that $f - \lambda$ is nonnegative iff it is a sum of squares (SOS), that is, $f = \sum_{i=1}^{t} g_i^2$ for a positive integer t and polynomials g_i.

Define the vector of monomials

$$z = (1, x, x^2, \ldots, x^n)^\top.$$

Then $f - \lambda$ is SOS if and only if (see [28, 37, 43, 62]) $f - \lambda = zz^\top \bullet Q$ for some $Q \succeq 0$. Matching monomials in $f - \lambda$ and $zz^\top \bullet Q$ we translate (3.52) into the SDP

$$\begin{array}{rrcl} \max & -A_0 \bullet Q & & \\ s.t. & A_i \bullet Q & = & a_i \text{ for } i = 1, \ldots, 2n \\ & Q & \in & \mathcal{S}_+^{n+1}. \end{array} \tag{3.53}$$

Here for all $i \in \{0, 1, \ldots, 2n\}$ the (k, ℓ) element of the matrix A_i is 1 if $k + \ell = i + 2$ for some $k, \ell \in \{1, \ldots, n+1\}$ and all other entries of A_i are zero.

The dual problem of (3.53) is

$$\begin{array}{rrcl} \min & \sum_{i=1}^{2n} a_i y_i & & \\ s.t. & \sum_{i=1}^{2n} y_i A_i + A_0 & \succeq & 0, \end{array} \tag{3.54}$$

whose constraints can be written as

$$\begin{pmatrix} 1 & y_1 & y_2 & \cdots & y_n \\ y_1 & y_2 & & \cdots & y_{n+1} \\ y_2 & & & \cdots & y_{n+2} \\ \vdots & & & \ddots & \vdots \\ y_n & y_{n+1} & y_{n+2} & \cdots & y_{2n} \end{pmatrix} \succeq 0.$$

Permuting rows and columns, this is equivalent to

$$\begin{pmatrix} y_{2n} & y_{2n-1} & y_{2n-2} & \cdots & y_n \\ y_{2n-1} & y_{2n-2} & & \cdots & y_{n-1} \\ y_{2n-2} & & & \cdots & y_{n-2} \\ \vdots & & & \ddots & \vdots \\ y_n & y_{n-1} & y_{n-2} & \cdots & 1 \end{pmatrix} \succeq 0. \tag{3.55}$$

Let us rename the variables so the even numbered ones come first, and the rest come afterwards, as

$$\begin{array}{rclrclcrcl} x_1 & = & y_{2n}, & x_2 & = & y_{2n-2}, & \ldots & x_n & = & y_2; \\ x_{n+1} & = & y_{2n-1}, & x_{n+2} & = & y_{2n-3}, & \ldots & x_{2n} & = & y_1. \end{array}$$

Then the constraints (3.55) become

$$\sum_{i=1}^{2n} x_i A'_i + B' \succeq 0. \tag{3.56}$$

Here the A'_i for $i = 1, \ldots, n$ and B' are defined as follows. In A'_i the (k, ℓ) and (ℓ, k) entry is 1, if $k + \ell = 2i$ and all other entries are zero. In B' the lower right corner is 1 and the other elements are zero.

For example, when $n = 3$ the constraints (3.56) look like

$$x_1 \begin{pmatrix} 1 & & & \\ & 0 & & \\ & & 0 & \\ & & & 0 \end{pmatrix} + x_2 \begin{pmatrix} 0 & & 1 & \\ & 1 & & \\ 1 & & 0 & \\ & & & 0 \end{pmatrix} + x_3 \begin{pmatrix} 0 & & & \\ & 0 & & 1 \\ & & 1 & \\ & 1 & & 0 \end{pmatrix} + \sum_{i=4}^{6} x_i A'_i + B' \succeq 0.$$

Thus $(A'_1, A'_2, \ldots, A'_n)$ is a regular facial reduction sequence and (3.56) is in the form of (P'). The tail-indices (cf. Definition 3.4) are $t_{j+1} = j + 2$ for $j = 1, \ldots, n-1$, hence we can derive the following inequalities, just like we did in Lemma 12:

$$x_j x_{j+2} \geq x_{j+1}^2 \text{ for } j = 1, \ldots, n-2; \text{ and } x_{n-1} \geq x_n^2.$$

Note that now the "δ" terms that appear in Lemma 12 are all zero, so we do not have to worry about "making x_k large." Hence by Lemma 15 we deduce that

$$x_j \geq x_{j+1}^{\alpha_{j+1}} \text{ for } j = 1, \ldots, n-1$$

hold, where $\alpha_{j+1} = 1 + 1/(n-j)$ for all j.

We translate these inequalities back to the original y_j variables, and obtain the following result:

Theorem 6. *Suppose that $y \in \mathbb{R}^{2n}$ is feasible in* (3.54)*. Then*

$$y_{2(n-j+1)} \geq y_{2(n-j)}^{1+1/(n-j)} \; for \, j = 1, \ldots, n-1.$$

$\square$

Combining these inequalities we obtain

$$y_{2n} \geq y_2^n.$$

Theorem 6 complements a result of Lasserre [28, Theorem 3.2], which states the following: if $\bar{x}$ minimizes the polynomial $f(x)$ then

$$(y_1, y_2, \ldots, y_{2n}) = (\bar{x}, \bar{x}^2, \ldots, \bar{x}^{2n})$$

is optimal in (3.54). On the one hand, Theorem 6 states bounds on all feasible solutions, on the other hand, it does not specify an optimal solution.

For completeness, we next revisit an example of O' Donnell in [40], and show how the SDP that arises in there is in the regular form of (P').

Example 3.4. We are given the polynomial with $2n$ variables

$$p(x,y) = p(x_1, \ldots, x_n, y_1, \ldots, y_n) = x_1 + \cdots + x_n - 2y_1$$

and the set K defined as

$$\begin{array}{rclrclcrcl} 2x_1y_1 &=& y_1, & 2x_2y_2 &=& y_2 & \ldots & 2x_ny_n &=& y_n \\ x_1^2 &=& x_1, & x_2^2 &=& x_2 & \ldots & x_n^2 &=& x_n \\ y_1^2 &=& y_2, & y_2^2 &=& y_3 & \ldots & y_n^2 &=& 0. \end{array}$$

Note that in the description of K the very last constraint $y_n^2 = 0$ breaks the pattern seen in the previous $n-1$ columns. We ask the following question:

- Is $p(x,y) \geq 0$ for all $(x,y) \in K$?

The answer is clearly yes, since for all $(x,y) \in K$ we have $x_1, \ldots, x_n \in \{0,1\}$ and $y_1 = \cdots = y_n = 0$.

On the other hand, the sum of squares procedure verifies the "yes" answer as follows. Let

$$z = (1, x_1, \ldots, x_n, y_1, \ldots, y_n)^\top$$

be a vector of monomials, and find $\lambda, \mu, \nu \in \mathbb{R}^n$ and $Q \succeq 0$ such that

$$\begin{aligned}
p(x,y) = z^\top Q z \quad &+ \quad \lambda_1(2x_1y_1 - y_1) + \mu_1(x_1^2 - x_1) + \nu_1(y_1^2 - y_2) \\
&+ \quad \lambda_2(2x_2y_2 - y_2) + \mu_2(x_2^2 - x_2) + \nu_2(y_2^2 - y_3) \\
&\vdots \\
&+ \quad \lambda_n(2x_ny_n - y_n) + \mu_n(x_n^2 - x_n) + \nu_n(y_n^2 - 0)
\end{aligned} \tag{3.57}$$

Matching coefficients on the left and right hand side, [40] shows that any Q feasible in (3.57) must be of the form

$$Q = \left(\begin{array}{ccccc|ccccc|c}
u_1 & 0 & \dots & 0 & 0 & -u_2 & \dots & 0 & 0 & 0 & 0 \\
0 & u_2 & \dots & 0 & 0 & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & 0 & \dots & -u_{n-1} & 0 & 0 & 0 \\
0 & 0 & \dots & u_{n-1} & 0 & 0 & \dots & 0 & -u_n & 0 & 0 \\
0 & 0 & \dots & 0 & u_n & 0 & \dots & 0 & 0 & -2 & 0 \\
\hline
-u_2 & \dots & 0 & 0 & 0 & 1 & \dots & 0 & 0 & 0 & 0 \\
\vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & \dots & -u_{n-1} & 0 & 0 & 0 & \dots & 1 & 0 & 0 & 0 \\
0 & \dots & 0 & -u_n & 0 & 0 & \dots & 0 & 1 & 0 & 0 \\
0 & \dots & 0 & 0 & -2 & 0 & \dots & 0 & 0 & 1 & 0 \\
\hline
0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0
\end{array}\right) \tag{3.58}$$

for suitable $u_1, \dots, u_n$. Looking at 2×2 subdeterminants of Q we see that the u_i satisfy

$$4 \leq u_n,\ u_n^2 \leq u_{n-1},\ \dots,\ u_2^2 \leq u_1, \tag{3.59}$$

which is the same as (*Khachiyan*), except we replaced the constant 2 by 4.

Let us define $E_{k\ell}$ to be the unit matrix in $\mathcal{S}^{2n+1}$ in which the (k, ℓ) and (ℓ, k) entries are 1 and the rest zero. Define

$$A'_1 = E_{11}, A'_i = E_{ii} - E_{i-1,n+i-1} \text{ for } i = 2, \ldots, n.$$

Then any Q feasible in (3.57) is written as

$$Q = u_1 A'_1 + u_2 A'_2 + \cdots + u_n A'_n + B' \succeq 0, \tag{3.60}$$

for a suitable B' (precisely, $B' = -2E_{n,2n} + \sum_{i=n+1}^{2n} E_{ii}$).

We see that $(A'_1, \ldots, A'_n)$ is a regular facial reduction sequence, thus the system (3.60) is in the regular form of (P').

$\Diamond$

We remark that (3.60) arises by concatenating 2×2 psd blocks of the form (3.1), then permuting rows and columns. In other words, (3.60) is the exact representation of (*Khachiyan*) (apart from the constant 2 being replaced by 4), that we discussed after Example 3.1.

Among followup papers of O' Donnell [40] we should mention the work of Raghavendra and Weitz [54] which gave SDPs which also have a sum-of-squares origin, and exponentially large solutions. It would be interesting to see whether those SDPs are also in the normal form of (P').

3.4 Conclusion

We showed that large size solutions do not just appear as pathological examples in semidefinite programs, but are quite common: after a linear change of variables, they appear in every strictly feasible SDP. As to "how large" they get, that depends on the singularity degree of a dual problem and the so-called tail-indices of the constraint matrices. Further, large solutions naturally appear in SDPs that come from minimizing a univariate polynomial, without any change of variables.

We also studied how to represent large solutions of SDPs in polynomial space. Our main tool was the regularized semidefinite program (P'). If (P) and (P') are strictly feasible, then in the latter we can verify that a strictly feasible solution exists, without computing the actual values of the “large” variables $x_1, \ldots, x_k$: see Figure 3.2. Further, SDPs that arise from polynomial optimization (Section 3.3) and the SDP that represents (*Khachiyan*) are naturally in the form of (P'). Hence in these SDPs we can also certify large solutions without computing their actual values.

Several questions remain open. For example, what can we say about large solutions in semidefinite programs that are not strictly feasible? The discussion after Example 3.1 shows that we do not have a complete answer.

Also, recall that we transform (P) into (P') by a linear change of variables (equivalent to operations (1) and (2) in Definition 3.2) and a similarity transformation (operation (3) in Definition 3.2). The latter has no effect on how large the variables are. We are thus led to the following question: are *all* SDPs with exponentially large solutions in the form of (P') (perhaps after a similarity transformation)? In other words, can we *always* certify large size solutions in SDPs using a regular facial reduction sequence? Answering this question would help us answer the greater question: can we decide feasibility of SDPs in polynomial time?

BIBLIOGRAPHY

[1] Amir Ali Ahmadi and Anirudha Majumdar. DSOS and SDSOS optimization: more tractable alternatives to sum of squares and semidefinite optimization. *SIAM Journal on Applied Algebra and Geometry*, 3(2):193–230, 2019.

[2] Amir Ali Ahmadi, Alex Olshevsky, Pablo A Parrilo, and John N Tsitsiklis. Np-hardness of deciding convexity of quartic polynomials and related problems. *Mathematical Programming*, 137(1):453–476, 2013.

[3] Amir Ali Ahmadi and Jeffrey Zhang. On the complexity of testing attainment of the optimal value in nonlinear optimization. *Mathematical Programming*, pages 1–21, 2019.

[4] Erling D Andersen and Knud D Andersen. The mosek interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In *High performance optimization*, pages 197–232. Springer, 2000.

[5] Alexander I Barvinok. Feasibility testing for systems of real quadratic equations. *Discrete & Computational Geometry*, 10(1):1–13, 1993.

[6] Heinz H Bauschke, Jonathan M Borwein, and Wu Li. Strong conical hull intersection property, bounded linear regularity, Jameson's property (G), and error bounds in convex optimization. *Mathematical Programming*, 86(1):135–160, 1999.

[7] Daniel Bienstock. A note on polynomial solvability of the cdt problem. *SIAM Journal on Optimization*, 26(1):488–498, 2016.

[8] Daniel Bienstock, Alberto Del Pia, and Robert Hildebrand. Complexity, exactness, and rationality in polynomial optimization. *arXiv preprint arXiv:2011.08347*, 2020.

[9] Grigoriy Blekherman. There are significantly more nonegative polynomials than sums of squares. *Israel Journal of Mathematics*, 153(1):355–380, 2006.

[10] Jon M Borwein and Henry Wolkowicz. Facial reduction for a cone-convex programming problem. *Journal of the Australian Mathematical Society*, 30(3):369–380, 1981.

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

[12] Peter Bürgisser and Felipe Cucker. *Condition: The geometry of numerical algorithms*, volume 349. Springer Science & Business Media, 2013.

[13] George B Dantzig. Linear programming. *Operations research*, 50(1):42–47, 2002.

[14] Dmitriy Drusvyatskiy and Henry Wolkowicz. The many faces of degeneracy in conic optimization. *Foundations and Trends® in Optimization*, 3(2):77–170, 2017.

[15] K. Fujisawa, M. Fukuda, M. Kojima, K. Nakata, M. Nakata, and M. Yamashita. SDPA (semidefinite programming algorithm) and SDPA-GMP User's Manual – Version 7.1.0. *Department of Mathematical and Computing Sciences, Tokyo Institute of Technology. Research Reports on Mathematical and Computing Sciences Series B-448*, 2008.

[16] Gene H Golub and James Hardy Wilkinson. Ill-conditioned eigensystems and the computation of the jordan canonical form. *SIAM review*, 18(4):578–619, 1976.

[17] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media, 2012.

[18] Didier Henrion. Semidefinite representation of convex hulls of rational varieties. *Acta Applicandae Mathematicae*, 115(3):319, 2011.

[19] Didier Henrion and Jean-Bernard Lasserre. Detecting global optimality and extracting solutions in GloptiPoly. In *Positive polynomials in control*, pages 293–310. Springer, 2005.

[20] Graham James Oscar Jameson. The duality of pairs of wedges. *Proceedings of the London Mathematical Society*, 3(3):531–547, 1972.

[21] Yuhan Jiang and Bernd Sturmfels. Bad projections of the psd cone. *arXiv preprint arXiv:2006.09956*, 2020.

[22] Cédric Josz and Didier Henrion. Strong duality in Lasserre's hierarchy for polynomial optimization. *Optimization Letters*, 10(1):3–10, 2016.

[23] N Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.

[24] Leonid Genrikhovich Khachiyan. A polynomial algorithm in linear programming. In *Doklady Akademii Nauk*, volume 244, pages 1093–1096. Russian Academy of Sciences, 1979.

[25] Victor Klee. Asymptotes and projections of convex sets. *Mathematica Scandinavica*, 8(2):356–362, 1961.

[26] 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.

[27] Masakazu Kojima, Shinji Mizuno, and Akiko Yoshise. A primal-dual interior point algorithm for linear programming. In *Progress in mathematical programming*, pages 29–47. Springer, 1989.

[28] Jean B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[29] Jean-Bernard Lasserre and Victor Magron. In SDP relaxations, inaccurate solvers do robust optimization. *SIAM Journal on Optimization*, 29(3):2128–2145, 2019.

[30] Minghui Liu and Gábor Pataki. Exact duality in semidefinite programming based on elementary reformulations. *SIAM J. Opt.*, 25(3):1441–1454, 2015.

[31] Minghui Liu and Gábor Pataki. Exact duals and short certificates of infeasibility and weak infeasibility in conic linear programming. *Mathematical Programming*, pages 1–46, 2017.

[32] Yanli Liu, Ernest K Ryu, and Wotao Yin. A new use of Douglas–Rachford splitting for identifying infeasible, unbounded, and pathological conic programs. *Mathematical Programming*, 177(1-2):225–253, 2019.

[33] Bruno F Lourenço. Amenable cones: error bounds without constraint qualifications. *to appear, Mathematical Programming, arXiv preprint arXiv:1712.06221*, 2017.

[34] Bruno F. Lourenço, Masakazu Muramatsu, and Takashi Tsuchiya. A structural geometrical analysis of weakly infeasible sdps. *Journal of the Operations Research Society of Japan*, 59(3):241–257, 2016.

[35] Theodore Samuel Motzkin. The arithmetic-geometric inequality. *Inequalities (Proc. Sympos. Wright-Patterson Air Force Base, Ohio, 1965)*, pages 205–224, 1967.

[36] Simone Naldi and Rainer Sinn. Conic programming: Infeasibility certificates and projective geometry. *Journal of Pure and Applied Algebra*, 225(7):106605, 2021.

[37] Yurii Nesterov. Squared functional systems and optimization problems. In *High performance optimization*, pages 405–440. Springer, 2000.

[38] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[39] Yurii Nesterov, Michael J Todd, and Yinyu Ye. Infeasible-start primal-dual methods and infeasibility detectors for nonlinear programming problems. *Mathematical Programming*, 84(2):227–268, 1999.

[40] Ryan O'Donnell. SOS is not obviously automatizable, even approximately. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[41] Dávid Papp and Sercan Yildiz. Sum-of-squares optimization without semidefinite programming. *SIAM Journal on Optimization*, 29(1):822–851, 2019.

[42] Panos M Pardalos and Stephen A Vavasis. Quadratic programming with one negative eigenvalue is NP-hard. *Journal of Global optimization*, 1(1):15–22, 1991.

[43] Pablo A Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming*, 96(2):293–320, 2003.

[44] Gábor Pataki. A simple derivation of a facial reduction algorithm and extended dual systems. *preprint http://gaborpataki.web.unc.edu/files/2018/07/fr.pdf*, 19, 2000.

[45] Gábor Pataki. A simple derivation of a facial reduction algorithm and extended dual systems. Technical report, Columbia University, 2000.

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

[47] 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.

[48] Gábor Pataki. Characterizing bad semidefinite programs: normal forms and short proofs. *SIAM Review*, 61(4):839–859, 2019.

[49] Gábor Pataki. Bad semidefinite programs: they all look the same. *SIAM J. Opt.*, 27(1):146–172, 2017.

[50] Javier Peña. Understanding the geometry of infeasible perturbations of a conic linear system. *SIAM Journal on Optimization*, 10(2):534–550, 2000.

[51] Javier Peńa and James Renegar. Computing approximate solutions for convex conic systems of constraints. *Mathematical Programming*, 87(3):351–383, 2000.

[52] Frank Permenter and Pablo Parrilo. Partial facial reduction: simplified, equivalent sdps via approximations of the psd cone. *Mathematical Programming*, pages 1–54, 2014.

[53] Lorant Porkolab and Leonid Khachiyan. On the complexity of semidefinite programs. *Journal of Global Optimization*, 10(4):351–365, 1997.

[54] Prasad Raghavendra and Benjamin Weitz. On the bit complexity of sum-of-squares proofs. *arXiv preprint arXiv:1702.05139*, 2017.

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

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

[57] James Renegar. A polynomial-time algorithm, based on newton's method, for linear programming. *Mathematical programming*, 40(1):59–93, 1988.

[58] James Renegar. On the computational complexity and geometry of the first-order theory of the reals. part i: Introduction. preliminaries. the geometry of semi-algebraic sets. the decision problem for the existential theory of the reals. *Journal of symbolic computation*, 13(3):255–299, 1992.

[59] James Renegar. Some perturbation theory for linear programming. *Mathematical Programming*, 65(1-3):73–91, 1994.

[60] James Renegar. Linear programming, complexity theory and elementary functional analysis. *Mathematical Programming*, 70(1-3):279–351, 1995.

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

[62] Naum Z Shor. Class of global minimum bounds of polynomial functions. *Cybernetics*, 23(6):731–734, 1987.

[63] Jos Sturm. Error bounds for linear matrix inequalities. *SIAM J. Optim.*, 10:1228–1248, 2000.

[64] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.

[65] Stephen A Vavasis. Quadratic programming is in NP. *Information Processing Letters*, 36(2):73–77, 1990.

[66] Stephen A Vavasis and Richard Zippel. Proving polynomial-time for sphere-constrained quadratic programming. Technical report, Cornell University, 1990.

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

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

[69] Yuzixuan Zhu, Gábor Pataki, and Quoc Tran-Dinh. Sieve-SDP: a simple facial reduction algorithm to preprocess semidefinite programs. *Mathematical Programming Computation*, 11(3):503–586, 2019.