Applications of optimization to factorization ranks and quantum information theory

Proefschrift ter verkrijging van de graad van doctor aan
Tilburg University
op gezag van de rector magnificus, prof. dr. K. Sijtsma,
in het openbaar te verdedigen ten overstaan van een door het college
voor promoties aangewezen commissie in
de portrettenzaal van de Universiteit
op maandag 30 september 2019 om 13.30 uur
door

Sander Jan Gribling

geboren te Capelle aan den IJssel.

Promotores:	prof. dr. M. Laurent
	prof. dr. R. M. de Wolf
Overige leden:	prof. dr. N. Bansal
	dr. J. Briët
	dr. H. Fawzi
	prof. dr. E. de Klerk
	prof. dr. F. Vallentin
	dr. J. C. Vera Lizcano

This research has been carried out at the Centrum Wiskunde & Informatica in Amsterdam.

Acknowledgements

First and foremost I would like to thank my advisors, Monique and Ronald. There are many good qualities one can look for in an advisor and find in the both of you, let me highlight two: thank you for showing me your passion for research while at the same time letting me choose which topics to work on! Another key lesson that you taught me is that results are only the first step in science, an equally important step is to then present them in a clear and concise way. It has been a great pleasure to work with both of you.

For taking part in my PhD committee and for providing valuable comments on this thesis I would like to thank Nikhil Bansal, Jop Briët, Hamza Fawzi, Etienne de Klerk, Frank Vallentin and Juan Vera Lizcano.

Next I would like to thank my other co-authors, David, Joran, and András. David, it was a lot of fun to discover the basics of quantum information theory together and I really enjoyed our discussions about matrix factorization ranks and optimization in general. Thanks for being there during the first two years of my time at CWI! It is needless to say that Part I would not have been the same without you. Next, Joran and András, whenever we started with a superposition of good and bad ideas, you always managed to amplify the good part; thanks for viewing the world in a quantum way! It has been great to explore the world of optimization from a quantum perspective together.

A special thanks also to my paranymphs, Roel and Pieter. Together with Hugo we have had too many card games, bad jokes, dinners and bike rides to count. Time goes so fast when you are having fun.

There are countless other people to thank for making life at CWI so much more enjoyable. I would hereby like to thank everyone for the many lunches, games of ping pong and foosball, gaming nights, conversations, and for simply saying 'Goedemorgen!' every day. To those of you who took part in the many, many games of ping pong and foosball: maybe it's best that you remain anonymous. Nevertheless, I would like to thank in particular Pieter, Mathé, Sven, Ruben and Isabella.

Thanks to the people from KAV Holland for letting me run away from the world of mathematics every week.

Finally I would like to thank my family, and especially Camille, for their endless support throughout the years.

Amsterdam　　　　　　　　　　　　　　　　　　　Sander Gribling
June, 2019.

Contents

Introduction

Optimization is a fundamental area in mathematics and computer science, with many real-world applications. The laws of quantum mechanics are one way to model this real world. For some physical experiments, this model predicts outcomes that are not possible under the laws of classical mechanics. In this thesis we study the difference between these predictions from the perspective of optimization. This study can be divided into two parts:

- *How can we use optimization techniques to understand and quantify the difference?*
- *How can this difference be exploited in order to solve optimization problems more efficiently?*

Just as the rest of this thesis, the introduction will be divided into two parts, addressing the above two research questions. In each of the parts of the introduction we will give examples that can be studied from different perspectives; these different perspectives will connect to the chapters that make up the part. This introduction is kept at a somewhat informal level. We assume that the reader has some basic understanding of graph theory and optimization, and we refer to the **Chapters 1**, **2**, **3**, **4**, and **9** for formal definitions of, and more background on, the more advanced concepts.

Part I

Gram matrices are basic objects that will play a central role in the first part of this thesis. A *Gram matrix* is a matrix A whose entries are given by the inner product between (real) vectors; i.e., a matrix A whose entries A_{ij} are of the form $A_{ij} = \langle x_i, x_j \rangle$, where $x_1, \ldots, x_n \in \mathbb{R}^d$ (for some $d, n \in \mathbb{N}$). We write $A = \mathrm{Gram}(x_1, \ldots, x_n)$, or we use the shorthand notation $A = \mathrm{Gram}(\{x_i\})$. It is well-known that a matrix A is positive semidefinite if and only if $A = \mathrm{Gram}(\{x_i\})$ for some vectors $x_1, \ldots, x_n \in \mathbb{R}^d$, where d can be chosen equal to the rank of A. We write S_+^n for the cone of $n \times n$ positive semidefinite matrices. Optimizing a linear function over the cone of positive semidefinite matrices (subject to some linear constraints) is known as *semidefinite programming*. Under mild conditions semidefinite programs can be solved efficiently, which makes them interesting not only in theory but also in practice.

A large part of this thesis focuses on Gram matrices of vectors with special properties. For instance, we consider the cone of $n \times n$ Gram matrices of entrywise nonnegative vectors, the *completely positive cone* CP^n. This cone has attracted a lot of attention due to its expressive power: many difficult optimization problems can be written as linear optimization problems over the completely positive cone. As an example we will use the stability number $\alpha(G)$ of a graph G, i.e., the maximum cardinality of a subset S of the vertices such that no edge has two endpoints in S. Equivalently, we can define the stability number of a graph $G = (V, E)$ on n vertices via the following quadratic program

$$\alpha(G) = \sup\Big\{ \sum_{i \in V} x_i : x \in \{0,1\}^n,\ x_i x_j = 0 \text{ if } \{i,j\} \in E \Big\}. \tag{1}$$

We can linearize the quadratic terms by introducing a matrix variable, and it can be shown that this leads to a characterization of $\alpha(G)$ as a linear optimization problem over the cone of completely positive matrices [dKP02]:

$$\begin{aligned} \alpha(G) = \sup\Big\{ \sum_{i,j \in V} X_{ij} : X \in \mathrm{CP}^n,\ \mathrm{Tr}(X) = 1, \\ X_{ij} = 0 \text{ if } \{i,j\} \in E \ \Big\}. \end{aligned} \tag{2}$$

Computing the stability number of a graph is an NP-hard problem. It follows that it is at least as hard to solve linear optimization problems over the completely positive cone. It is therefore natural to consider outer approximations of the cone of completely positive matrices. If we replace the cone CP^n in (2) by the cone S^n_+ of positive semidefinite matrices then we arrive at an efficiently computable upper bound on $\alpha(G)$, the celebrated Lovász theta number $\vartheta(G)$ [Lov79]:

$$\begin{aligned} \vartheta(G) = \sup\Big\{ \sum_{i,j \in V} X_{ij} : X \in \mathrm{S}^n_+,\ \mathrm{Tr}(X) = 1, \\ X_{ij} = 0 \text{ if } \{i,j\} \in E \ \Big\}. \end{aligned} \tag{3}$$

A second way to view the stability number of a graph is via a specific *nonlocal game*. In this game there are two parties, who are trying to convince a referee that they know a stable set of size k and that they have agreed on a labeling of the vertices in this stable set (say with numbers 1 up to k). This is a one-round game, where the two players may agree on a strategy before the game starts, but they are not allowed to communicate after the start of the game. The game works as follows. The referee asks each of the two parties to reveal a very small part of their stable set: a single vertex. Let us say that the first player has to reveal the ath vertex in the stable set, and the second player the bth vertex. The two parties do not know which vertex the other party has to reveal. The referee becomes convinced that the two parties know a labeled stable set of size k if their answers are consistent with the existence of such a labeled stable set, no matter what questions he asks: if the players are both asked to reveal the ath vertex ($a \in [k]$), then their answers should be same vertex, and if they are asked to reveal different vertices, then their answers should be different and non-adjacent vertices. To model the fact that the players have to answer consistently no matter which questions are asked, we assume that there is a distribution according to which the question pairs are drawn that is strictly positive on all possible question pairs. For concreteness, we may assume that the question pairs are drawn according to the uniform distribution on $[k] \times [k]$. We say that the players have a *perfect strategy* if they provide consistent answers with probability 1. Suppose the players use a deterministic strategy to provide their answers. Clearly, if there exists a (labeled) stable set of size k then the players have a perfect deterministic strategy: before the game starts they can agree on a labeling of a stable set of size k and then they can answer 'honestly'. Here by 'honestly' we mean that if a player is asked to reveal the ath vertex, then he/she reveals the ath vertex of the labeled stable set. In fact one can also show the reverse: if they have a perfect deterministic strategy, then there exists a stable set of size k. We can thus characterize $\alpha(G)$ as the largest $k \in \mathbb{N}$ for which there exists a perfect deterministic strategy for the above game.

Instead of deterministic strategies, we can try to give the players some more power. For instance we may allow the players to base their answers on two local measurements to a shared quantum mechanical system. If we do so, then we say that the players use a *quantum strategy.* We can then define the *quantum stability number* of G, denoted by $\alpha_q(G)$, as the largest $k \in \mathbb{N}$ for which there exists a perfect quantum strategy for the above game. For precise definitions of a quantum strategy and the quantum stability number we refer to Chapters 3 and 8. As we will see later, deterministic strategies form a special type of quantum strategies and therefore we have the inequality

$$\alpha(G) \leq \alpha_q(G).$$

A separation between $\alpha_q(G)$ and $\alpha(G)$ is a way to quantify the difference between the quantum mechanical model of the physical world and the classical models. It shows the power of *entanglement.* A mathematical separation between $\alpha_q(G)$ and $\alpha(G)$ can be turned into an *experimental* separation between the two physical models; we could try to build a quantum mechanical system with which we can play the above nonlocal game perfectly when the questions are drawn uniformly from $[\alpha_q(G)] \times [\alpha_q(G)]$.

Let us go back to the formulation of $\alpha(G)$ as a linear optimization problem over the cone of completely positive matrices introduced in Equation (2). We can view a nonnegative vector as a diagonal positive semidefinite matrix. This makes it natural to study the cone of Gram matrices of positive semidefinite matrices, i.e., the cone of matrices $A = (\langle X_i, X_j \rangle)$, where we now use the trace inner-product between the positive semidefinite matrices $X_1, \ldots, X_n$. This cone is called the *completely positive semidefinite cone* and denoted by CS_+^n. By construction we have the inclusions $\mathrm{CP}^n \subseteq \mathrm{CS}_+^n \subseteq \mathrm{S}_+^n$. It therefore makes sense to ask the following: what happens if we replace the cone CP^n by CS_+^n in (2)? It can be shown that the new parameter obtained in this way forms an upper bound on the quantum stability number of the graph G [LP15, Prop. 4.9]. As we will see in Chapter 8, the cone CS_+ can in fact be used to formulate the quantum stability number $\alpha_q(G)$: one can show that $\alpha_q(G)$ is at least k if and only if there exists a matrix $A \in \mathrm{CS}_+^{nk}$ that satisfies certain linear constraints.

Finally we mention a third way to view the stability number of a graph. Instead of looking at $\alpha(G)$ as a conic optimization problem or expressing it through a nonlocal game, we can also see it as a polynomial optimization problem by replacing the integrality constraint $x \in \{0,1\}^n$ in the program (1) by $x_i - x_i^2 = 0$ for all $i \in V$:

$$\alpha(G) = \sup\Big\{ \sum_{i \in V} x_i : x \in \mathbb{R}^V,\ x_i - x_i^2 = 0 \text{ for } i \in V,\ x_i x_j = 0 \text{ if } \{i,j\} \in E \Big\}. \tag{4}$$

In this way we can use the theory of polynomial optimization to define a hierarchy of semidefinite programming upper bounds that converges to $\alpha(G)$. What about the quantum stability number? It turns out that the same hierarchy when applied to *noncommutative* polynomials provides upper bounds on the quantum stability number.

The connections between matrix factorizations, polynomial optimization and nonlocal games are precisely the topics of the first part of this thesis. As we will see, the theory of C^**-algebras* (an infinite-dimensional analogue of matrix algebras) plays an important role in connecting the three topics.

Organization of Part I. We first provide in **Chapter 5** a unified approach to lower bounding four different matrix factorization ranks, based on techniques from (noncommutative) polynomial optimization. These four different factorization ranks are obtained by using nonnegative vectors or positive semidefinite matrices as factors, and they may be symmetric or not, depending on whether the same factors are used for the rows and for the columns. Then, in **Chapter 6**, we use semidefinite programming techniques to construct nonlocal games for which optimal quantum strategies require large quantum mechanical systems, this leads to a family of completely positive semidefinite matrices with a high factorization rank. We say that these quantum strategies use a large amount of *entanglement.* In **Chapter 7** we introduce a new measure for the amount of entanglement needed to generate a quantum strategy and we show that this measure can be phrased in the language of noncommutative polynomial optimization (and thus it can be approximated using hierarchies of semidefinite programs). Finally in **Chapter 8** we return to graph parameters, we study the quantum stability number and a quantum analogue of the chromatic number. We do so from the perspective of noncommutative polynomial optimization. This perspective allows us to define semidefinite programming hierarchies, in analogy to the case of the classical graph parameters. Notably, this perspective unifies some existing bounds on these quantum graph parameters.

Part II

As many applications require solving larger and larger optimization problems, the efficiency of optimization becomes more and more important, motivating us to find the best possible algorithms. For most of the optimization problems that we will see in Part I, even moderate-size instances are too large for currently available classical computers to deal with in reasonable time and/or memory. Should we improve our algorithms? Yes! But, we could also "cheat" and change our model of computation. We could use the model of *quantum computing*. This model of computation has been studied for several decades. Recent experimental progress on building quantum computers suggests that by changing to this model we are not "cheating"; this model of computation might soon be a reality and therefore we should focus our attention on finding new, faster, quantum algorithms. In this thesis we contribute by considering the following question:

> *Can we solve optimization problems more efficiently by exploiting quantum effects such as superposition, interference, and entanglement?*

Let us mention two of the most important quantum algorithms, the second of which we will use to connect to the topics in Part II of this thesis. One of the most remarkable quantum algorithms is due to Shor [Sho97]; he formulated a quantum algorithm that can find the prime factors of a given integer N in polynomial time, which is much faster than currently possible on a classical computer. Shor's algorithm solves a very specific problem, but that problem is a very central one in the field of cryptography. Several cryptographic schemes are based on the (unproven) assumption that finding prime factors of large integers is computationally hard. Below we will see the second classical (in the historical sense) example of a problem that quantum computers can solve faster than classical computers: the problem of searching an unsorted search space. Here the speed-up will be less significant, but it will be much more widely applicable.

Let us now consider the problem of searching an unsorted search space. This problem is fundamental in many (classical) algorithms. In this problem, one is given an unsorted list and the goal is to find an entry in the list with a particular property. Formally, this is modeled in the following way. One is given an n-bit string $x \in \{0,1\}^n$ and the goal is to find a 1: an index $i \in [n]$ such that $x_i = 1$. How difficult is it to solve this problem? To answer that question we need to agree on a way of accessing the string x. A natural way to access the string x on a classical computer is through queries to the individual bits of x, that is, through queries of the form "what is x_i?". Suppose that we are given the promise that there is only a single 1 in the string x. Then any classical algorithm (whether deterministic or probabilistic) will need to make at least $n/2$ queries to succeed with probability $1/2$ on every such input string. What about a quantum algorithm? Again, we need to specify the access to the string x. A natural analogue of the classical queries is to allow the quantum computer to query bits of x in *superposition*. One can show that there is an algorithm, called *Grover search*, that uses such queries and finds an index i such that $x_i = 1$ using a number of queries in the order of $\sqrt{n}$ [Gro96]. Hence, this quantum algorithm provides a quadratic speed-up compared to the best possible classical algorithm. We will see the Grover search algorithm in Chapter 9.

Can quantum computers solve other problems faster as well? Given the abundance of semidefinite programs in the first part of this thesis, a natural question to ask is whether quantum computers can solve semidefinite programs more efficiently. Or, more generally, can quantum computers solve *convex optimization problems* more efficiently? Precisely these questions are studied in Chapters 11 and 12. In

Chapter 11 we provide a novel quantum algorithm for solving semidefinite programs. This algorithm fits into the framework of (matrix) multiplicative weights update methods [AK16]. We then continue to study convex optimization problems in Chapter 12. There we consider the problem of solving convex optimization problems when access to the underlying convex set is only given implicitly, through an oracle. One can consider different types of oracle access to the convex set, for example a *membership oracle* or *a separation oracle.* Our main result in Chapter 12 is a quantum algorithm that uses membership oracle queries to construct a separation oracle; the number of membership queries needed is exponentially smaller than in the classical setting.

If quantum computers offer speed-ups, how large can those speed-ups be? How can we even lower bound the number of queries a quantum algorithm needs to make? After all, each query can be a superposition over all possible classical queries. It turns out that we can use polynomials to find such lower bounds. Suppose we have a quantum algorithm that on input $x \in \{0,1\}^n$ should return $f(x)$, where $f : \{0,1\}^n \to \{0,1\}$ is a Boolean function that is known in advance and access to x is through the type of queries we have described above. Then, the crucial observation made in [BBC$^+$01] is that the success probability of a t-query quantum algorithm is a polynomial p of degree $2t$ in the variables $x_1, \dots, x_n$. If the algorithm has to succeed with high probability on every input, then p is not just any polynomial; it is a polynomial that approximates f on each input $x \in \{0,1\}^n$. The *approximate degree* of a Boolean function f is defined as the smallest degree of a polynomial that approximates f on all its inputs to an error of, say, at most $1/3$. (This notion predates quantum computing by several decades, see for instance [MP68, NS94].) As an important example, one can show that the approximate degree of the OR function is of the order $\sqrt{n}$. Here the OR function is the function that maps all input strings $x \in \{0,1\}^n$ to 1, except the all-zero string which is mapped to 0. The OR function can be seen as a decision version of the problem that we have seen above: given a string $x \in \{0,1\}^n$, does there exist an index $i \in [n]$ for which $x_i = 1$? Since the approximate degree of the OR function is of the order $\sqrt{n}$, a quantum algorithm for the decision version of the search problem needs to make at least a number of queries of the order $\sqrt{n}$. Since Grover search can in particular be used to solve the decision version, it follows that Grover search is an optimal quantum algorithm.

Deriving lower bounds on the number of quantum queries needed to approximate a Boolean function through the approximate degree is known as the *polynomial method.* Given its tightness for the unsorted search problem, it is natural to ask if the polynomial method is tight in general. The answer is no, the version that we described above is not tight in general. But, the method can be strengthened by observing that polynomials corresponding to quantum algorithms satisfy certain additional properties [AA15, ABP19]. In fact, it has recently been shown that quantum algorithms correspond precisely to polynomials that are *completely bounded* [ABP19]. That is, to each quantum algorithm we can associate a completely bounded polynomial and vice versa. The notion of completely boundedness of polynomials connects quantum algorithms to the theory of operator spaces; a theory closely connected to that of C^*-algebras which we will see in Part I. Other than this link to operator spaces, what do we gain from this connection? As we will show in Chapter 10, the completely bounded norm of a polynomial can be expressed using semidefinite programming. Thus this connection leads to a new semidefinite programming characterization of the strength of quantum algorithms for computing Boolean functions in the query model.

Organization of Part II. A basic introduction to quantum computing is provided in **Chapter 9**. In **Chapter 10** we explain the connection between quantum algorithms, completely bounded polynomials, and semidefinite programming. At this point, each of the preceding chapters has contained at least one semidefinite program. We therefore turn our attention to solving semidefinite programs faster on quantum computers in **Chapter 11**. We then explore the more general problem of convex optimization, albeit with a different input model, in **Chapter 12**.

Publications

This dissertation is based on the following six articles (in order of the chapters in which they appear).

[GdLL19] S. Gribling, D. de Laat, and M. Laurent. Lower bounds on matrix factorization ranks via noncommutative polynomial optimization. *Foundations of Computational Mathematics*, Jan 2019.

[GdLL17] S. Gribling, D. de Laat, and M. Laurent. Matrices with high completely positive semidefinite rank. *Linear Algebra and its Applications*, 513:122–148, 2017.

[GdLL18] S. Gribling, D. de Laat, and M. Laurent. Bounds on entanglement dimensions and quantum graph parameters via noncommutative polynomial optimization. *Mathematical Programming*, 170:5–42, 2018.

[GL19] S. Gribling and M. Laurent. Semidefinite programming formulations for the completely bounded norm of a tensor. arXiv:`1901.04921`, 2019.

[vAGGdW17] J. van Apeldoorn, A. Gilyén, S. Gribling, and R. de Wolf. Quantum SDP-solvers: Better upper and lower bounds. In *Proceedings of the 58th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 403–414, 2017. arXiv:`1705.01843`.

[vAGGdW18] J. van Apeldoorn, A. Gilyén, S. Gribling, and R. de Wolf. Convex optimization using quantum oracles. arXiv:`1809.00643`, 2018.

The author has additionally co-authored the following article that is not included in this dissertation.

[vAG18b] J. van Apeldoorn and S. Gribling. Simon's problem for linear functions. arXiv:`1810.12030`, 2018.

Chapter 1

Semidefinite optimization

In this background chapter we define the main optimization frameworks that are used in this thesis: semidefinite optimization and, more generally, convex optimization. Semidefinite optimization is also known as semidefinite programming and abbreviated as SDP. We state some well-known results regarding the duality theory of semidefinite optimization and we provide complexity statements. Many excellent books and surveys exist about these topics, for instance [VB96, WSV00, BTN01, Lov03, BV04, AL12]. We refer to those sources for more information.

1.1 Semidefinite programming

A matrix $A \in \mathbb{C}^{n\times n}$ is called a *Hermitian* matrix if $A^* = A$, where the operation * maps a matrix to the entry-wise complex conjugate of its transpose. We let H^n denote the set of $n \times n$ Hermitian matrices. A fundamental result in linear algebra is that an $n \times n$ Hermitian matrix has n real (not necessarily distinct) eigenvalues. A Hermitian matrix $A \in \mathrm{H}^n$ is called *positive semidefinite* if all its eigenvalues are nonnegative. We use the notation $A \succeq 0$ to denote that A is positive semidefinite, and the notation H^n_+ for the set of $n \times n$ Hermitian positive semidefinite matrices. We let $\langle A, B\rangle = \mathrm{Tr}(A^*B)$ be the trace inner product on $\mathbb{C}^{n\times n}$. Let $A \in \mathrm{H}^n$. Then it is known that the following are equivalent:

(i) $A \succeq 0$,

(ii) $v^*Av \geq 0$ for all $v \in \mathbb{C}^n$,

(iii) $A = \mathrm{Gram}(v_1, \ldots, v_n)$ for some vectors $v_1, \ldots, v_n \in \mathbb{C}^d$ ($d \in \mathbb{N}$),

(iv) $A = V^*V$ for some $V \in \mathbb{C}^{d\times n}$ ($d \in \mathbb{N}$),

(v) $\langle A, B\rangle \geq 0$ for all $B \succeq 0$.

In fact the conditions (ii),(iii), and (iv) each directly imply that A is Hermitian. When A is a real-valued symmetric matrix we may restrict to real vectors in (ii) and (iii) and to real matrices in (iv) and (v). We use S^n to denote the set of real-valued symmetric $n \times n$ matrices and we let $\mathrm{S}^n_+ \subset \mathrm{S}^n$ be the subset of *real symmetric positive semidefinite matrices.* For $A \in \mathbb{R}^{n\times n}$ the adjoint A^* equals the transpose A^T of A.

Linear optimization over the cone of positive semidefinite matrices is known as *semidefinite optimization.* For integers $m, n \in \mathbb{N}$, a set of $n \times n$ matrices $C, A_1, \dots, A_m \in \mathrm{S}^n$ and a vector $b \in \mathbb{R}^m$ define a pair of semidefinite programs, a *primal* (P) and a *dual* (D):

$$\begin{array}{llll} (P) & \sup & \langle C, X\rangle \\ & \text{s.t.} & X \in \mathrm{S}^n_+ \\ & & \mathcal{A}(X) = b \end{array} \qquad \begin{array}{llll} (D) & \inf & \langle b, y\rangle \\ & \text{s.t.} & y \in \mathbb{R}^m \\ & & \mathcal{A}^*(y) - C \in \mathrm{S}^n_+ \end{array} \tag{1.1}$$

Here, $\mathcal{A} : \mathrm{S}^n \to \mathbb{R}^m$ is the linear operator defined by

$$\mathcal{A}(X) = (\mathrm{Tr}(A_1X), \dots, \mathrm{Tr}(A_mX)),$$

whose adjoint $\mathcal{A}^*$ acts on $\mathbb{R}^m$ as $\mathcal{A}^*(y) = \sum_{i=1}^m y_iA_i$, so that $\langle \mathcal{A}(X), y\rangle = \langle X, \mathcal{A}^*(y)\rangle$. The matrix $X \in \mathrm{S}^n_+$ and the vector $y \in \mathbb{R}^m$ are called the *variables* of respectively the primal (P) and the dual (D). Matrices $X \in \mathrm{S}^n_+$ (resp., vectors $y \in \mathbb{R}^m$) that satisfy the constraints of (P) (resp., (D)) are called *feasible solutions.* We say that an optimization problem is *feasible* if there exists a feasible solution. Let X and y be feasible solutions to (P) and (D), respectively; then we can compare their *objective values* $\langle C, X\rangle$ and $\langle b, y\rangle$:

$$\langle b, y\rangle = \langle \mathcal{A}(X), y\rangle = \langle X, \mathcal{A}^*(y)\rangle \geq \langle X, C\rangle, \tag{1.2}$$

where the inequality follows from $X, \mathcal{A}^*(y) - C \succeq 0$ and point (v) above. This shows that when (P) and (D) are both feasible the maximum in (P), its *optimal value*, is at most the minimum in (D). This is known as *weak duality.* We say that *strong duality* holds if the optimal values of (P) and (D) are equal.[1] Semidefinite programs do not always have strong duality, in addition they do not always attain their optimal values. But there is a sufficient condition known as *Slater's condition*, which is based on the concept of *strict feasibility.*[2] A matrix X whose eigenvalues are strictly positive is called *positive definite*, denoted $X \succ 0$; if it is also a feasible solution to (P) then we call it a *strictly feasible solution* to (P). Slater's condition allows us to say the following:

> If (P) has a strictly feasible solution, then strong duality holds. If in addition the primal optimal value is bounded from above, then the optimal value in (D) is finite and attained.

Notice that Slater's condition does not imply that the primal optimal value is attained (it could even be infinite). In our applications we will often have a strictly feasible primal whose set of feasible solutions (i.e., its *feasible region*) is bounded.

[1] Here we use the convention that the value of (P) (resp. (D)) is $-\infty$ (resp. $+\infty$) if it is infeasible.

[2] This is not the only sufficient condition for strong duality. For example, if (P) is feasible and there exist $y_0, \dots, y_m$ such that $\sum_{i=1}^m y_iA_i - y_0C \succ 0$, then strong duality holds [Bar02, Prop. IV.10.2].

It is easy to see that Slater's condition together with boundedness of the primal feasible region implies that both the primal and dual optimal values are finite and attained.

A special class of semidefinite programs is formed by *linear programs*, those SDPs for which all matrices involved are diagonal. For linear programs we always have strong duality.

We record an analogue of *Farkas' Lemma* for semidefinite programs.

Lemma 1.1 ([Lov03, Lem. 3.3])**.** *Let $B_1, \ldots, B_k, C \in \mathrm{S}^n$. Then the following are equivalent:*

(*) The system $\sum_{j=1}^k y_j B_j - C \succ 0$ has no solution in $y_1, \ldots, y_k \in \mathbb{R}$,

(**) There exists a symmetric matrix $Y \neq 0$ such that $\langle B_j, Y\rangle = 0$ for all $j \in [k]$, $\langle C, Y\rangle \geq 0$, and $Y \succeq 0$.

In Section 6.2.2 we will use the equivalent formulation given below.[3]

Lemma 1.2. *Let $A_1, \ldots, A_m \in \mathrm{S}^n$ and $b \in \mathbb{R}^m$. Assume that there exists a matrix $X_0 \in \mathrm{S}^n$ such that $\langle A_j, X_0\rangle = b_j$ for all $j \in [m]$. Then exactly one of the following two alternatives holds:*

(i) There exists a matrix $X \succ 0$ such that $\langle A_j, X\rangle = b_j$ for all $j \in [m]$.

(ii) There exists $y \in \mathbb{R}^m$ such that $\Omega = \sum_{j=1}^m y_j A_j \succeq 0$, $\Omega \neq 0$, and $b^T y \leq 0$.

The complexity of solving SDPs. Semidefinite programs can be used to model and approximate a variety of combinatorial optimization problems. This is useful for at least two reasons. Firstly, it allows us to apply the duality theory that we have seen above to prove properties of these combinatorial problems. Secondly, as we discuss now, under some mild conditions semidefinite programs can be solved efficiently allowing us to efficiently compute bounds on combinatorial problems. Let us first give a statement about the efficiency with which we can solve semidefinite programs in the Turing model of complexity.

Theorem 1.3 ([GLS81])**.** *Let $C, A_1, \ldots, A_m \in \mathrm{S}^n$ and $b \in \mathbb{R}^m$ be rational. The matrices $C, A_1, \ldots, A_m$ and the vector b together define a primal/dual pair of semidefinite programs as in Equation* (1.1)*. Let $\mathcal{F}$ be the feasible region of the primal problem (P) and assume we know a rational point $X_0 \in \mathcal{F}$ and rational numbers r and R such that*

$$X_0 + \tilde{B}(X_0, r) \subseteq \mathcal{F} \subseteq X_0 + \tilde{B}(X_0, R).$$

Here $\tilde{B}(X_0, r)$ is the ball of radius r, centered at X_0, in the lower-dimensional space

$$L = \{X \in \mathrm{S}^n : \mathcal{A}(X) = 0\}.$$

[3] To see the equivalence, set $C = -X_0$ and let $A_1, \ldots, A_m$ and $B_1, \ldots, B_k$ be such that

$$\mathrm{span}\{A_j : j \in [m]\}^\perp = \mathrm{span}\{B_j : j \in [k]\}.$$

Then, for any positive rational number $\varepsilon > 0$ one can find a rational matrix $X^* \in \mathcal{F}$ whose objective value is within additive error ε of the optimal value of (P), in time polynomial in $n, m, \log(\frac{R}{r}), \log(\frac{1}{\varepsilon})$, and the bit size of the data $X_0, C, A_1, \ldots, A_m, b$.

In [GLS81] this theorem is proven constructively using the ellipsoid method. There they show that the ellipsoid method can be used to efficiently optimize over a bounded convex set if we are given an efficient separation oracle for the convex set. A separation oracle for (P) needs to decide if a given rational matrix X is feasible for (P), and if it is not feasible then it needs to provide a hyperplane separating X from the feasible region of (P). The authors of [GLS81] then show that one can efficiently solve the separation problem for (P): we first check if the linear constraints are satisfied, if there is a violated constraint, then this provides a separating hyperplane. If all linear constraints are satisfied, then we check if $X \succeq 0$. The latter can be done efficiently using Gaussian elimination, which, if $X \not\succeq 0$, also provides a hyperplane separating X from S^n_+ (and thus from the feasible region of (P)).

In practice the more recent interior point methods are preferred (see for instance the book [NN94] or the monograph [Ren01]). Recently it has been shown that the runtime of a certain interior point method is also polynomial in the input size [dKV16] (under similar assumptions as Theorem 1.3). The currently (asymptotically) fastest method is a so-called cutting plane method due to Lee, Sidford, and Wong [LSW15]. Notice that here we aim for a runtime that scales polynomially with $\log(1/\varepsilon)$. Alternatively, we could also consider the regime where the runtime scales polynomially with $1/\varepsilon$. In the latter regime one can sometimes obtain a better dependence on the parameters n and m (see, e.g., the matrix multiplicative weight update method [AHK12]). In Chapter 11 we will present a *quantum* algorithm for solving SDPs whose runtime is sublinear in n and m (its dependence on the other parameters, such as $1/\varepsilon$, is less favourable).

The above shows that most SDPs are 'easy' to solve. Let us emphasize that this need not be the case when, for example, the conditions of the above theorem are not met. The above complexity statements assume we know a feasible point for the primal problem. In later chapters we will encounter *feasibility problems*, SDPs where the question is precisely whether the SDP is feasible. We do not know to which complexity class the SDP feasibility problem belongs; all we know is that either it belongs to $\mathrm{NP} \cap \mathrm{coNP}$ or it does not belong to $\mathrm{NP} \cup \mathrm{coNP}$ [Ram97]. Nevertheless the problem is decidable, we now describe how to do so. A key observation is that the feasible region of an SDP in primal form (P) can be described using polynomial inqualities on the entries of X. Indeed, the linear constraints provide linear (in)equalities on the $\binom{n+1}{2}$ real-valued entries of X. Moreover, positive semidefiniteness of a matrix X can be expressed using (an exponential number of) polynomial inequalities on the entries of X; $X \succeq 0$ if and only if the determinant of every principal submatrix of X is nonnegative. Therefore, the feasible region of an SDP can be described using polynomial inequalities on real variables. Testing feasibility of a set of polynomial inequalities can be done, under mild conditions, using Renegar's quantifier elimination method [Ren92]. It can be shown that this method can be used to decide if the feasible region of an SDP is non-empty, see [PK97]. There is a bound on the running time of Renegar's algorithm, but, for SDPs it does not run in polynomial time.

1.2 Convex optimization

Semidefinite programs form a special class of *convex optimization problems*. The general *convex optimization problem* is to maximize a linear function $c^T x$ over points $x \in K \subseteq \mathbb{R}^n$, where $c \in \mathbb{R}^n$ and K is a closed convex set:

$$\max \quad c^T x \quad \text{s.t. } x \in K.$$

As we have mentioned before, the ellipsoid method can be used to efficiently solve a convex optimization problem, if we are given access to an efficient separation oracle for K. Phrased somewhat informally, the ellipsoid method shows that we can do linear optimization over a convex set K using a polynomial number of queries to a separation oracle for K. This turns one type of access to K (separation) into another type of access (optimization). Besides optimization and separation, another natural way to access K is through queries of the form "does x belong to K?"; these queries are called membership queries. Grötschel, Lovász, and Schrijver [GLS88] showed that these different types of access are polynomially equivalent: given an oracle O that provides one of the types of access, we can construct an oracle for any of the other types of access that uses a polynomial number of queries to O. A fundamental question is thus how efficient these oracle 'reductions' can be made. Over the years progress has been made in both the number of queries and the time complexity needed for the oracle reductions. In Chapter 12 we contribute to this line of research by studying how efficient these reductions can be made on a *quantum computer*. We refer to that chapter for formal statements about the efficiency of solving convex optimization problems on classical and quantum computers given different types of access to the convex set K.

Chapter 2

Matrix factorization ranks

In this background chapter we motivate and define the four matrix factorization ranks that are of interest in the first part of this thesis: the nonnegative rank, the positive semidefinite rank, and their symmetric analogues, the completely positive rank and the completely positive semidefinite rank. We collect some known results and then we prove a first new result: the completely positive semidefinite rank can be quadratically smaller than the completely positive rank (Section 2.2, based on [GdLL17, Prop. 2.3]).

2.1 Matrix factorization ranks

Let $\{K^d\}_{d\in\mathbb{N}}$ be a sequence of cones that are each equipped with an inner product $\langle\cdot,\cdot\rangle$. Throughout we assume that each cone K^d is self-dual. A factorization of a matrix $A \in \mathbb{R}^{m\times n}$ over K^d is a decomposition of the form $A = (\langle X_i, Y_j\rangle)$ with $X_i, Y_j \in K^d$ for all indices $i \in [m], j \in [n]$, for some integer $d \in \mathbb{N}$. Following [GPT13], the smallest integer d for which such a factorization exists is called the *cone factorization rank* of A over $\{K^d\}$:

$$\min\Big\{d \in \mathbb{N} : \exists X_1,\ldots,X_m,Y_1,\ldots,Y_n \in K^d,\ A = \big(\langle X_i, Y_j\rangle\big)_{i\in[m],j\in[n]}\Big\}.$$

We use three sequences of cones in this thesis. First, we use the nonnegative orthant $\mathbb{R}^d_+$ with the usual inner product. The associated cone factorization rank is called the *nonnegative rank* and it is denoted by $\mathrm{rank}_+(A)$. Secondly, we use the cones of $d\times d$ real symmetric positive semidefinite matrices S^d_+ with the trace inner product $\langle X, Y\rangle = \mathrm{Tr}\big(X^TY\big)$, and thirdly we use their complex analogues, the cones of $d\times d$ complex Hermitian positive semidefinite matrices H^d_+ with the trace inner product $\langle X,Y\rangle = \mathrm{Tr}(X^*Y)$. The associated cone factorization ranks are the real and complex *positive semidefinite rank*, denoted $\text{psd-rank}_{\mathbb{K}}(A)$ where $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$. Both the nonnegative rank and the positive semidefinite rank are defined whenever the matrix A is entrywise nonnegative.

Factorization ranks & extension complexity. A fundamental problem in the area of optimization is that of linear optimization over a *polytope* P, a bounded subset of $\mathbb{R}^n$ defined by linear inequalities. Such problems are called *linear programs* (LPs). When using interior point methods, the time needed to solve an LP depends on the number of linear inequalities used to describe the underlying polytope P (see, e.g., [Kar84, Ren88, BTN01]): LPs that can be described with few inequalities can be solved efficiently. It is therefore important to find the most efficient formulation of a given polytope P. For instance, the ℓ_1-unit ball in $\mathbb{R}^n$ can be described using 2^n linear inequalities:

$$P = \big\{x \in \mathbb{R}^n : z^T x \leq 1 \text{ for all } z \in \{-1,1\}^n\big\}.$$

But one can describe it more succinctly, using $2n$ inequalities and n auxiliary variables, as the projection of the polytope

$$Q = \Big\{(x,y) \in \mathbb{R}^{2n} : -x_i \leq y_i,\ x_i \leq y_i \text{ for all } i \in [n],\ \sum_{i\in[n]} y_i = 1\Big\}$$

on the x-variables. The size of the smallest representation of P is called its extension complexity, it is formally defined as follows. The *linear extension complexity* of P is the smallest integer d for which P can be obtained as a linear image of the intersection between an affine subspace and the nonnegative orthant $\mathbb{R}^d_+$. Analogously, the *semidefinite extension complexity* of P is the smallest d such that P is a linear image of the intersection between an affine subspace and the cone S^d_+.

The motivation to study the linear and semidefinite extension complexities is that polytopes with small extension complexity admit efficient algorithms for linear optimization. Well-known examples include spanning tree polytopes [Mar91] and permutahedra [Goe15], which have polynomial linear extension complexity, and the stable set polytope of perfect graphs, which has polynomial semidefinite extension complexity [MGS81] (see, e.g., the surveys [CCZ10, FGP$^+$15]).

In a groundbreaking work, Yannakakis [Yan91] showed that the *symmetric* linear extension complexity of important combinatorial polytopes such as the traveling salesman polytope and the matching polytope is exponential in the number of vertices of the graph. The precise definition of symmetric extension complexity is not relevant for this thesis, but we want to point out that this enabled Yannakakis to immediately refute a polynomial-size linear formulation of the traveling salesman polytope proposed in [Swa86].[1]

How does this connect to factorization ranks? To answer this question we need to consider a certain matrix associated to the polytope: the *slack matrix* of P. The *slack matrix* S of P is the matrix

$$S = (b_i - a_i^T v)_{v\in V, i\in I},$$

where $P = \mathrm{conv}(V)$ and $P = \{x : a_i^T x \leq b_i \ (i \in I)\}$ are point and hyperplane representations of P. In other words, the matrix S records the amount of (nonnegative!) slack each vertex has in each inequality defining P. As Yannakakis [Yan91]

[1] A word of warning, symmetric extended formulations have nothing to do with the symmetric cone factorization ranks studied below.

showed, the linear extension complexity of a polytope P is given by the nonnegative rank of its slack matrix. More recently, it is shown that the semidefinite extension complexity of a polytope is equal to the (real) positive semidefinite rank of its slack matrix [GPT13].

The above connection to the nonnegative rank and to the positive semidefinite rank of the slack matrix can be used to show that some polytopes do not admit a small extended formulation. Recently this connection was used to show that the symmetry assumption of Yannakakis [Yan91] was not needed: the linear extension complexity of the cut polytope is exponential in the number of nodes n [FMP$^+$15]. Via known reductions this implies that the linear extension complexity of the traveling salesman polytope is $2^{\Omega(\sqrt{n})}$, and that there is a family of graphs for which the linear extension complexity of the stable set polytope is $2^{\Omega(\sqrt{n})}$ [FMP$^+$15]. Subsequent work showed that there in fact exists a family of graphs whose stable set polytopes have extension complexity $2^{\Omega(n/\log(n))}$ [GJW18]. To summarize, we know that the linear extension complexities of the cut polytope, the traveling salesman polytope, and the stable set polytope (for certain graphs) are of the form $2^{\Omega(n^c)}$ for some constants $c > 0$. Later it was shown that also the semidefinite extension complexities of these polytopes are of the form $2^{\Omega(n^c)}$, albeit with smaller constants $c > 0$ [LRS15]. Surprisingly, the linear extension complexity of the matching polytope is also exponential [Rot17], even though linear optimization over this set is polynomial time solvable [Edm65]. It is an open question whether the semidefinite extension complexity of the matching polytope is exponential. Some evidence has been provided in [BBCH$^+$17] where it is shown that there exists no *symmetric* semidefinite extended formulation of the matching polytope.

Besides this link to extension complexity, both of these factorization ranks also have connections to (quantum) communication complexity. For the nonnegative rank see, e.g., [FFGT15], and for the positive semidefinite rank see, e.g., [FMP$^+$15, JSWZ13].

As another application we mention that factorizations through the cone $\mathbb{R}^d_+$ are important in machine learning. Consider for instance the task of dividing a collection of text documents into clusters of 'related' documents. Let A be the matrix whose (i, j)th entry A_{ij} indicates the number of occurrences of the ith word in the jth document. Then a nonnegative matrix factorization $A = VF$, where $V \in \mathbb{R}^{m\times k}_+, F \in \mathbb{R}^{k\times n}_+$ can be used to cluster the documents according to dominant 'topics' (e.g., assign document j to cluster $\ell \in [k]$ for which $F_{\ell j} = \mathrm{argmax}_h F_{hj}$). See, e.g., the book [Moi18] for more details.

Symmetric cone factorization ranks. For a square symmetric $n \times n$ matrix $A \in \mathrm{S}^n$ we are also interested in *symmetric* analogues of the above matrix factorization ranks, where we require the same factors for the rows and columns (i.e., $X_i = Y_i$ for all $i \in [n]$). The symmetric analogue of the nonnegative rank is the *completely positive rank*, denoted cp-rank(A), which uses the cones $K^d = \mathbb{R}^d_+$, and the symmetric analogue of the positive semidefinite rank is the *completely positive semidefinite rank*, denoted cpsd-rank$_{\mathbb{K}}(A)$, which uses the cones $K^d = \mathrm{S}^d_+$ if $\mathbb{K} = \mathbb{R}$ and $K^d = \mathrm{H}^d_+$ if $\mathbb{K} = \mathbb{C}$. These symmetric factorization ranks are not always well defined since not every symmetric nonnegative matrix admits a symmetric factorization by nonnegative vectors or positive semidefinite matrices. The symmetric matrices for which these parameters are well defined form convex cones known as the *completely positive cone*, denoted CP^n, and the *completely positive semidefinite cone*, denoted CS_+^n. To see that these sets form convex cones it suffices to observe that $\mathrm{Gram}(\lambda X_1, \ldots, \lambda X_n) = \lambda^2 \,\mathrm{Gram}(X_1, \ldots, X_n)$ and that $\mathrm{Gram}(X_1 \oplus Y_1, \ldots, X_n \oplus Y_n) = \mathrm{Gram}(X_1, \ldots, X_n) + \mathrm{Gram}(Y_1, \ldots, Y_n)$. Here we use the fact that the direct sum of two nonnegative vectors (or two positive semidefinite matrices) is again a nonnegative vector (or positive semidefinite matrix). By considering the tensor product of two factorizations we see that the completely positive (semidefinite) cones are closed under the tensor product. We have the inclusions

$$\mathrm{CP}^n \subseteq \mathrm{CS}_+^n \subseteq \mathrm{S}_+^n \cap \mathbb{R}_+^{n \times n}.$$

These inclusions are known to be strict for $n \geq 5$, while for $n \leq 4$ we have equality throughout $\mathrm{CP}^4 = \mathrm{S}_+^4 \cap \mathbb{R}_+^{4\times 4}$. For details on these cones see [BSM03, BLP17, LP15] and references therein. Note that membership in the cone CS_+^n does not depend on whether we use real symmetric or complex Hermitian positive semidefinite matrices as factors because mapping a Hermitian $d \times d$ matrix X to

$$\frac{1}{\sqrt{2}} \begin{pmatrix} \mathrm{Re}(X) & \mathrm{Im}(X) \\ \mathrm{Im}(X)^T & \mathrm{Re}(X) \end{pmatrix} \in \mathrm{S}^{2d} \tag{2.1}$$

is an isometry that preserves positive semidefiniteness. It follows that for a matrix $A \in \mathrm{CS}_+^n$ we have

$$\text{cpsd-rank}_{\mathbb{R}}(A) \leq 2\, \text{cpsd-rank}_{\mathbb{C}}(A)$$

and for a matrix $A \in \mathbb{R}_+^{m \times n}$ we have

$$\text{psd-rank}_{\mathbb{R}}(A) \leq 2\, \text{psd-rank}_{\mathbb{C}}(A).$$

Basic properties of the cones CP and CS_+. One of the basic questions one can ask about a set is whether it is closed. The cone of completely positive $n \times n$ matrices CP^n is closed. This can be seen as follows. As we mention later (see Eq. (2.2)), the cp-rank of an $n \times n$ matrix $A \in \mathrm{CP}^n$ is upper bounded by a function that only depends on n. From this, and the observation that the factors of a symmetric factorization are bounded in norm, one can derive that the cone CP^n is closed using a compactness argument.

What about the cone CS_+^n? One could try to follow the same strategy as we followed for the cone CP^n. Again, the factors of a symmetric factorization are bounded in norm. However, for this cone we do not have an upper bound on the factorization rank in terms of n. In fact, as we explain in the paragraph below Equation (2.3), we know that for $n \geq 10$ the cone CS_+^n is not closed and thus such an upper bound cannot exist, for $n \geq 10$. We do not know if CS_+^n is closed for $n \in \{5, 6, 7, 8, 9\}$ and for those n it remains an open question whether such an upper bound on the cpsd-rank exists.

Understanding why small matrices with a large cpsd-rank exist and why their cpsd-rank is large remains a challenging task. In Chapters 5 and 6 we use techniques from semidefinite optimization to shed some light on this topic.

Knowing that the cone CS_+^n is not closed for large enough n motivates studying its closure. A description of the closure of the completely positive semidefinite cone in terms of factorizations by positive elements in von Neumann algebras can be found in [BLP17]. Such factorizations were used to show a separation between the closure of CS_+^n and the cone $\mathrm{S}_+^n \cap \mathbb{R}_+^{n\times n}$ of doubly nonnegative matrices (see [FW14, LP15]).

Symmetric cone factorizations & optimization. The study of the cones CP^n and CS_+^n is motivated in particular by their use to model classical and quantum information optimization problems. For instance, graph parameters such as the stability number and the chromatic number can be written as linear optimization problems over the completely positive cone [dKP02, GL08b], and the same holds, more generally, for quadratic problems with mixed binary variables [Bur09]. The completely positive cone can moreover be used to express some models of uncertainty in (mixed integer) linear programs, see for example [NTZ11, HK18]. The cp-rank is widely studied in the linear algebra community; see, e.g., [BSM03, SMBJS13, SMBB$^+$15, BSU14].

The completely positive semidefinite cone was first studied in [LP15] to describe quantum analogues of the stability number and of the chromatic number of a graph (see Chapter 8). This was later extended to general graph homomorphisms in [SV17] and to graph isomorphism in [AMR$^+$19]. In addition, as shown in [MR14, SV17], there is a close connection between the completely positive semidefinite cone and the set of quantum correlations. This also gives a relation between the completely positive semidefinite rank and the minimal entanglement dimension necessary to realize a quantum correlation. We will revisit the connection between CS_+ and quantum correlations in Chapter 3 and use it in Chapter 6 to construct matrices whose completely positive semidefinite rank is exponentially large in the matrix size.

Known upper bounds. The following inequalities hold for the nonnegative rank and the positive semidefinite rank:

$$\text{psd-rank}_{\mathbb{C}}(A) \leq \text{psd-rank}_{\mathbb{R}}(A) \leq \text{rank}_+(A) \leq \min\{m, n\}$$

for any $m \times n$ nonnegative matrix A, where the last inequality holds in light of the nonnegative factorization $A = I_m A = A I_n$. By Carathéodory's theorem, the completely positive rank of a matrix in CP^n is at most $\binom{n+1}{2} + 1$. In [SMBB$^+$15] it is shown that this bound can be strengthened to

$$\text{cp-rank}(A) \leq \binom{n+1}{2} - 4 \quad \text{for} \quad A \in \mathrm{CP}^n \quad \text{and} \quad n \geq 5. \tag{2.2}$$

One can sometimes obtain tighter bounds by comparing the cp-rank with the rank: in [HL83, BB03] the following bound is shown

$$\text{cp-rank}(A) \leq \binom{\text{rank}(A)+1}{2} - 1 \quad \text{for} \quad A \in \mathrm{CP}^n,\ \text{rank}(A) \geq 2. \tag{2.3}$$

As we hinted at before, the situation for the cpsd-rank is very different. Exploiting the connection between the completely positive semidefinite cone and quantum correlations (see Chapter 3), it follows from results in [Slo19] that the cone CS_+^n is not closed for $n \geq 1942$. The results in [DPP19] show that this already holds for $n \geq 10$. As a consequence there does not exist an upper bound on the cpsd-rank as a function of the matrix size. For small matrix sizes very little is known. It is an open problem whether CS_+^5 is closed, and we do not even know how to construct a 5×5 matrix whose cpsd-rank exceeds 5.

By taking direct sums of factors, it is easy to see that each of the above mentioned factorization ranks is subadditive.

To obtain upper bounds on the factorization rank of a given matrix one can employ heuristics that try to construct small factorizations. Many such heuristics exist for the nonnegative rank (see the overview [Gil17] and references therein), factorization algorithms exist for completely positive matrices (see the recent paper [GD18], also [DD12] for structured completely positive matrices), and algorithms to compute positive semidefinite factorizations are presented in the recent work [VGG18].

Known lower bounds. Due to the embeddings of $\mathbb{R}_+^d$ in $\mathbb{R}^d$, S^d in $\mathbb{R}^{\binom{d+1}{2}}$, and H_+^d in $\mathbb{R}^{d^2}$, we have the trivial lower bounds

$$\mathrm{rank}_+(A) \geq \mathrm{rank}(A), \qquad \text{psd-rank}_{\mathbb{C}}(A)^2 \geq \mathrm{rank}(A),$$

for $A \in \mathbb{R}_+^{m\times n}$. Similarly, for $A \in \mathrm{CP}^n$ we have

$$\text{cp-rank}(A) \geq \mathrm{rank}(A),$$

and for $A \in \mathrm{CS}_+^n$ we have

$$\text{cpsd-rank}_{\mathbb{C}}(A)^2 \geq \mathrm{rank}(A). \tag{2.4}$$

Similar bounds hold for the real (completely) positive semidefinite rank. In Chapter 5 we define new generic lower bounds on each of the factorization ranks and we compare our bounds more extensively to existing generic lower bounds. We refer to for instance [BSM03] for more lower bounds on the cp-rank of structured matrices.

Complexity. The rank_+, cp-rank, and psd-rank are known to be computable; this follows using Renegar's quantifier elimination method [Ren92] since upper bounds exist on these factorization ranks that depend only on the matrix size, see [BR06] for a proof for the case of the cp-rank.[2] These algorithms in general do not run in polynomial time. However, for a fixed integer k one can check in polynomial time in the size of the matrix whether the nonnegative rank is at most k [AGKM16, Moi16] and whether the positive semidefinite rank is at most k [Shi18].[3] It is known that computing the nonnegative rank is NP-hard [Vav09]. In fact, determining the rank_+

[2] For matrices with rational entries these factorization ranks are computable in the bit model. For real-valued matrices they are computable in the real-number model.

[3] Similar to the previous footnote, we need to distinguish between matrices with rational or real entries. Again, the computational model is respectively the bit model and the real-number model.

and psd-rank of an integer-valued matrix are both equivalent to the existential theory of the reals [Shi16, Shi17]. For the cp-rank and the cpsd-rank no such results are known, but there is no reason to assume they are any easier. In fact, since no a priori upper bound exists on the cpsd-rank, it is not even clear whether the cpsd-rank is computable in general. It is known that deciding membership in the completely positive cone is NP-hard [DG14].

2.2 Separating cp-rank and cpsd-rank

For the completely positive rank we have the quadratic upper bound (2.2), and completely positive matrices have been constructed whose completely positive rank grows quadratically in the size of the matrix. This is the case, for instance, for the matrices

$$M_k = \begin{pmatrix} I_k & \frac{1}{k}J_k \\ \frac{1}{k}J_k & I_k \end{pmatrix} \in \mathrm{CP}^{2k},$$

whose cp-rank is known to be equal to k^2, see Proposition 2.1. Here $I_k \in \mathrm{S}^k$ is the identity matrix and $J_k \in \mathrm{S}^k$ is the all-ones matrix. This means the completely positive rank of these matrices is within a constant factor of the upper bound $\binom{2k+1}{2} - 4$ given in Equation (2.2). The significance of the matrices M_k stems from the Drew-Johnson-Loewy conjecture [DJL94] which was recently disproved [BSU14, BSU15]. This conjecture states that $\lfloor n^2/4 \rfloor$ is an upper bound on the completely positive rank of $n \times n$ matrices, which means the matrices M_k are sharp for this bound.

It was observed in [PSVW18] that by combining the rank lower bound (2.4) on the completely positive semidefinite rank with (2.3) we obtain the following relation:

$$\Omega(\text{cp-rank}(A)^{1/4}) \leq \text{cpsd-rank}(A) \leq \text{cp-rank}(A) \quad \text{for } A \in \mathrm{CP}^n.$$

This leads to the natural question of how fast cpsd-rank(M_k) grows. We show in Proposition 2.2 below that the completely positive semidefinite rank grows linearly for the matrices M_k, and we exhibit a link to the question of existence of Hadamard matrices. More precisely, we show that $\text{cpsd-rank}_{\mathbb{C}}(M_k) = k$ for all k, and $\text{cpsd-rank}_{\mathbb{R}}(M_k) = k$ if and only if there exists a real Hadamard matrix of order k. In particular, this shows that the real and complex completely positive semidefinite ranks can be different.

A *real Hadamard* matrix of order k is a $k \times k$ matrix with pairwise orthogonal columns and whose entries are ±1-valued. Likewise a *complex Hadamard* matrix of order k is a $k \times k$ matrix with pairwise orthogonal columns and whose entries are complex valued with unit modulus. A complex Hadamard matrix exists for any order; take for example

$$(H_k)_{i,j} = e^{2\pi \mathbf{i}(i-1)(j-1)/k} \quad \text{for} \quad i,j \in [k], \tag{2.5}$$

the matrix corresponding to the discrete Fourier transform. On the other hand, it is still an open conjecture whether a real Hadamard matrix exists for each order k that is a multiple of 4.

It is well-known that the completely positive rank of M_k equals k^2, for completeness we provide a proof. Here, the support of a vector $u \in \mathbb{R}^d$ is the set of indices $i \in [d]$ for which $u_i \neq 0$.

Proposition 2.1 (folklore)**.** *The completely positive rank of M_k is equal to k^2.*

Proof. For $i \in [k]$ consider the vectors $v_i = 1/\sqrt{k}\, e_i \otimes \mathbf{1}$ and $u_i = 1/\sqrt{k}\, \mathbf{1} \otimes e_i$, where e_i is the ith basis vector in $\mathbb{R}^k$ and $\mathbf{1}$ is the all-ones vector in $\mathbb{R}^k$. The vectors $v_1, \ldots, v_k, u_1, \ldots, u_k$ are nonnegative and form a Gram representation of M_k, which shows cp-rank$(M_k) \leq k^2$.

To prove the lower bound, suppose $M_k = \mathrm{Gram}(v_1, v_2, \ldots, v_k, u_1, u_2, \ldots, u_k)$ with $v_i, u_i \in \mathbb{R}^d_+$. In the remainder of the proof we show $d \geq k^2$. We have $(M_k)_{i,j} = \delta_{ij}$ for $1 \leq i, j \leq k$. Since the vectors v_i are nonnegative, they must have disjoint supports. The same holds for the vectors $u_1, \ldots, u_k$. Since $(M_k)_{i,j} = 1/k > 0$ for $1 \leq i \leq k$ and $k+1 \leq j \leq 2k$, the support of v_i overlaps with the support of u_j for each i and j. This means that for each $i \in [k]$, the size of the support of the vector v_i is at least k. This is only possible if $d \geq k^2$. □

Proposition 2.2 ([GdLL17])**.** *For each $k \in \mathbb{N}$ we have* cpsd-rank$_{\mathbb{C}}(M_k) = k$. *Moreover, we have* cpsd-rank$_{\mathbb{R}}(M_k) = k$ *if and only if there exists a real Hadamard matrix of order k.*

Proof. The lower bound cpsd-rank$_{\mathbb{C}}(M_k) \geq k$ follows because I_k is a principal submatrix of M_k and cpsd-rank$_{\mathbb{C}}(I_k) = k$. To show cpsd-rank$_{\mathbb{C}}(M_k) \leq k$, we give a factorization by Hermitian positive semidefinite $k \times k$ matrices. For this consider the complex Hadamard matrix H_k in (2.5) and define the factors

$$X_i = e_i e_i^T \quad \text{and} \quad Y_i = \frac{u_i u_i^*}{k} \quad \text{for} \quad i \in [k],$$

where e_i is the ith standard basis vector of $\mathbb{R}^k$ and u_i is the ith column of H_k. By direct computation it follows that $M_k = \mathrm{Gram}(X_1, \ldots, X_k, Y_1, \ldots, Y_k)$.

We now show that cpsd-rank$_{\mathbb{R}}(M_k) = k$ if and only if there exists a real Hadamard matrix of order k. One direction follows directly from the above proof: If a real Hadamard matrix of size k exists, then we can replace H_k by this real matrix and this yields a factorization by real positive semidefinite $k \times k$ matrices.

Now assume cpsd-rank$_{\mathbb{R}}(M_k) = k$ and let $X_1, \ldots, X_k, Y_1, \ldots, Y_k \in \mathrm{S}^k_+$ be a Gram representation of M. We first show there exist two orthonormal bases $u_1, \ldots, u_k$ and $v_1, \ldots, v_k$ of $\mathbb{R}^k$ such that $X_i = u_i u_i^T$ and $Y_i = v_i v_i^T$. For this we observe that $I = \mathrm{Gram}(X_1, \ldots, X_k)$, which implies $X_i \neq 0$ and $X_i X_j = 0$ for all $i \neq j$. Hence, for all $i \neq j$, the range of X_j is contained in the kernel of X_i. Therefore the range of X_i is orthogonal to the range of X_j. We now have

$$\sum_{i \in [k]} \dim(\mathrm{range}(X_i)) = \dim\Big(\sum_{i \in [k]} \mathrm{range}(X_i)\Big) \leq k$$

and $\dim(\mathrm{range}(X_i)) \geq 1$ for all i. From this it follows that $\mathrm{rank}(X_i) = 1$ for all $i \in [k]$. This means there exist $u_1, \ldots, u_k \in \mathbb{R}^k$ such that $X_i = u_i u_i^T$ for all i. From

$I = \mathrm{Gram}(X_1, \ldots, X_k)$ it follows that the vectors $u_1, \ldots, u_k$ form an orthonormal basis of $\mathbb{R}^k$. The same argument can be made for the matrices Y_i, thus $Y_i = v_i v_i^T$ and the vectors $v_1, \ldots, v_k$ form an orthonormal basis of $\mathbb{R}^k$. Up to an orthogonal transformation we may assume that the first basis is the standard basis; that is, $u_i = e_i$ for $i \in [k]$. We then obtain

$$\frac{1}{k} = (M_k)_{i,j+k} = \langle e_i, v_j \rangle^2 = \big((v_j)_i\big)^2 \quad \text{for} \quad i, j \in [k],$$

hence $(v_j)_i = \pm 1/\sqrt{k}$. Therefore, the $k \times k$ matrix whose kth column is $\sqrt{k}\, v_k$ is a real Hadamard matrix. □

The above proposition leaves open the value of $\text{cpsd-rank}_{\mathbb{R}}(M_k)$ for the cases where a real Hadamard matrix of order k does not exist. Extensive experimentation using a heuristic (see [GdLL17, Section 2.2]) suggests that for $k = 3, 5, 6, 7$ the real completely positive semidefinite rank of M_k equals $2k$, which leads to the following question:

Question 2.3. Is the real completely positive semidefinite rank of M_k equal to $2k$ if a real Hadamard matrix of size $k \times k$ does not exist?

Note that the lower bounds we develop in Chapter 5 are on the *complex* completely positive semidefinite rank (which is k), therefore they cannot be used to answer the above question.

We also used the heuristic from [GdLL17, Section 2.2] to check numerically that the aforementioned matrices from [BSU14], which have completely positive rank greater than $\lfloor n^2/4 \rfloor$, have small (smaller than n) real completely positive semidefinite rank. In fact, for every completely positive $n \times n$ matrix we tried in our numerical experiments, we could always find a cpsd factorization in dimension n, which leads to the following question:

Question 2.4. Is the real (or complex) completely positive semidefinite rank of a completely positive $n \times n$ matrix upper bounded by n?

Chapter 3

Quantum information theory

Here we give some basic mathematical background on quantum information theory. For more details see for example [NC00], or the lecture notes [Wat11, dW11].

Which set of rules governs the physical world around us? Are the laws of classical mechanics the correct model? Or does the world behave according to the laws of quantum mechanics? To answer these questions one can study the predictions that each of these models makes about certain experiments. In this chapter we explore the predictions made about probability distributions arising from measurements to a (quantum) mechanical system. In Part II of this thesis we will study the difference between classical computers (Turing machines) and *quantum computers*, computers acting according to the laws of quantum mechanics. See Chapter 9 for some background information on the topic of quantum computing.

Below we first explain some basic terminology, leading up to the type of probability distributions that can occur between two parties who simultaneously measure parts of the same physical system. These distributions are called *bipartite correlations*. We then explain the framework of nonlocal games, which can be used to quantify the difference between classical and quantum correlations. Finally we show how bipartite quantum correlations are related to the cone of completely positive semidefinite matrices which we have seen in the previous chapter.

3.1 The basics

A physical system can be described by a state. We can learn information about a state by measuring it, and we can try to alter a state by acting on it. Below we describe the mathematical model, according to the laws of quantum mechanics, of a state and the allowed operations to it. We end the section with an example illustrating the concepts.

Quantum states. The state of a quantum mechanical system with finitely many degrees of freedom is described by a *density matrix* ρ, that is, a Hermitian positive semidefinite matrix whose trace is equal to 1. We call ρ a *pure state* if it has rank one, else it is called a *mixed state*. Whenever we refer to a unit vector $\psi \in \mathbb{C}^d$ as a state, it should be understood as the pure state $\rho = \psi\psi^*$. We exclusively work with column vectors, so the state $\rho = \psi\psi^*$ is indeed a $d \times d$ density matrix. For two states $\phi, \psi \in \mathbb{C}^d$ we refer to the complex number $\phi^*\psi$ as the *amplitude* of ψ in the state ϕ. Throughout this thesis we almost exclusively work with pure states. For infinite-dimensional systems a pure state can be described by a unit vector in a complex separable Hilbert space.

Quantum operations. The postulates of quantum mechanics say that the pure state ψ of a quantum mechanical system can evolve in one of the following two ways. We can apply a unitary U to ψ to obtain the new quantum state $U\psi$, such evolutions are studied in Chapter 9. Or, we can measure the system.

Definition 3.1 (POVM). *A* positive operator-valued measurement (POVM) *with m possible outcomes is described by a collection of Hermitian positive semidefinite operators $E_1, \dots, E_m$ that satisfy $\sum_{i\in[m]} E_i = I$. When measuring the pure state ψ, the probability of observing outcome $i \in [m]$ is given by $\langle \psi, E_i\psi\rangle = \mathrm{Tr}(E_i\psi\psi^*)$.*

We sometimes refer to a POVM as a *measurement device.* Notice that the values $\langle \psi, E_i\psi\rangle$ can indeed be viewed as a probability of observing outcome i: it is a value between 0 and 1 and $\sum_{i=1}^m \langle \psi, E_i\psi\rangle = \langle \psi, \psi\rangle = 1$. Often, each outcome of a measurement is associated to a numerical value. It thus makes sense to talk about the expected outcome of a measurement. To a measurement (POVM) $\{E_1, \dots, E_m\}$ whose outcomes are labeled by $v_1, \dots, v_m \in \mathbb{R}$ we can associate the Hermitian operator $\sum_{i=1}^m v_iE_i$. This operator is called the *observable* associated to the measurement. It connects a pure state ψ to the expected outcome under the measurement: $\psi \mapsto \langle \psi, (\sum_{i=1}^m v_iE_i)\psi\rangle$.

A special class of POVMs is formed by those in which all operators E_i are projectors. Such a POVM is called a *projective measurement* (PVM). For a PVM we can talk about the post-measurement state. If we observe outcome i when we are measuring ψ with a PVM $E_1, \dots, E_m$, then ψ *collapses* to its projection on the range of E_i, i.e., the state $E_i\psi/\sqrt{\langle \psi, E_i\psi\rangle}$.

An important example of a PVM is the *measurement in the computational basis*, given by $\{e_1e_1^*, \dots, e_de_d^*\}$ where $e_i \in \mathbb{C}^d$ is the ith standard basis vector ($i \in [d]$). When using this measurement on a state $\psi \in \mathbb{C}^d$ the probability of observing outcome i equals $\psi^*e_ie_i^*\psi = |\psi_i|^2$.

Quantum states & linear functionals. To a pure state $\psi \in \mathbb{C}^d$ we can associate the linear functional $\tau : \mathbb{C}^{d\times d} \to \mathbb{C}$ defined as

$$A \mapsto \langle \psi, A\psi\rangle = \psi^*A\psi = \mathrm{Tr}(A\psi\psi^*).$$

The linear functional τ maps measurement operators $E_1, \dots, E_m$ to the probability of observing outcome i when using that measurement: $\tau(E_i) = \psi^*E_i\psi$. By linearity it maps observables to the expected outcome of the associated measurement on ψ. In fact, the linear functional τ maps elements from the matrix algebra $\mathbb{C}^{d\times d}$ to complex numbers. The infinite-dimensional analogue of a matrix algebra is the

$*$-algebra $\mathcal{B}(\mathcal{H})$ of bounded operators on a Hilbert space $\mathcal{H}$. For a state $\psi \in \mathcal{H}$ we could analogously define $\tau : \mathcal{B}(\mathcal{H}) \to \mathbb{C}$ by $A \mapsto \langle \psi, A\psi \rangle$. In Section 4.1.2 we will encounter such linear functionals in the context of noncommutative polynomial optimization. There we will see that, under certain conditions, we can also associate a quantum state ψ to a linear functional τ (through the GNS construction, see the proof of Theorem 4.5).

Composite systems. A quantum mechanical system is often composed of several subsystems. We sometimes call these subsystems *registers* or *parts*. In the finite-dimensional setting, this can be modeled by assuming a tensor product structure on the Hilbert space. An important example is that of an n-qubit system where the associated Hilbert space is given by $(\mathbb{C}^2)^{\otimes n}$. A fundamental concept is that of an entangled state:

Definition 3.2 (Entangled state). *A finite-dimensional k-partite state*

$$\psi \in \mathbb{C}^{d_1} \otimes \cdots \otimes \mathbb{C}^{d_k}$$

is called entangled *if it cannot be written as a tensor product $\psi = \psi_1 \otimes \cdots \otimes \psi_k$ where $\psi_i \in \mathbb{C}^{d_i}$ for $i \in [k]$.*

In Section 3.2.3 we will see that one way to model distinct ‘parts’ of an infinite-dimensional quantum system is to assume that measurements that are done to different parts commute.

Example 3.3. The state $\psi = \frac{1}{\sqrt{2}} e_1 \otimes e_1 + \frac{1}{\sqrt{2}} e_2 \otimes e_2 \in \mathbb{C}^2 \otimes \mathbb{C}^2$ is called an *EPR-pair* [EPR35]. It is an example of a 2-partite entangled state. If we measure the first register of this state in the *computational basis*, that is, if we use the PVM $\{E_1 = e_1 e_1^* \otimes I_2, E_2 = e_2 e_2^* \otimes I_2\}$ (where I_2 is the identity operator on $\mathbb{C}^2$), then the probability of seeing outcome i equals

$$\psi^* E_i \psi = (\frac{1}{\sqrt{2}} e_1 \otimes e_1 + \frac{1}{\sqrt{2}} e_2 \otimes e_2)^* (e_i e_i^* \otimes I_2)(\frac{1}{\sqrt{2}} e_1 \otimes e_1 + \frac{1}{\sqrt{2}} e_2 \otimes e_2) = 1/2,$$

and the post-measurement state is given by $e_i \otimes e_i$. The linear functional associated to ψ is defined as

$$\tau(A) = \psi^* A \psi = \frac{A_{11} + A_{14} + A_{41} + A_{44}}{2}. \qquad \triangle$$

3.2 Bipartite correlations

An important question is what advantage entangled states have compared to states that are not entangled. Here we focus on quantum mechanical systems composed of two subsystems, say states on $\mathbb{C}^d \otimes \mathbb{C}^d$, and we assume each subsystem is controlled by a different party. The tensor structure of the Hilbert space suggests to think of the two parties as being separated: they are not allowed to interact with each other's subsystem. This setting is called the *bipartite setting* and it has been widely used to study entanglement (for a survey, see, e.g., [PV16]).

We assume that each of the subsystems is measured by a different party, let us call the two parties Alice and Bob. We study the resulting joint probability distribution on the outcomes. Formally, this means that Alice and Bob each have a POVM acting on $\mathbb{C}^d$, say $\{E_i\}_{i\in I}$ and $\{F_j\}_{j\in J}$. Then we consider the probability distribution arising from the joint measurement $\{E_i \otimes F_j\}_{(i,j)\in I\times J}$ on $\mathbb{C}^d \otimes \mathbb{C}^d$.

The above Example 3.3 can be phrased in this language. Let us say that Alice measures the first qubit and Bob the second one. Then the example corresponds to Alice using the PVM $\{e_1e_1^*, e_2e_2^*\}$ and Bob the trivial PVM $\{I\}$. The corresponding probability distribution corresponding to this experiment is very simple to describe: Bob only has one possible outcome (hence he sees it with probability 1), and, as the example showed, Alice sees each of her outcomes with probability 1/2. Of course such measurement statistics can easily be obtained by separated parties: Alice simply flips an unbiased coin.

A more interesting situation arises when we let both Alice and Bob use the PVM $\{e_1e_1^*, e_2e_2^*\}$. It is easy to verify that in this case the outcome of Alice's measurement always equals that of Bob's: the outcomes $(1,1)$ and $(2,2)$ are each observed with probability 1/2. We would still consider such a probability distribution classical since the two separated parties can obtain such statistics through the use of *shared randomness*. In fact, if Alice and Bob know each other's measurement device then we can always use shared randomness to reproduce the probability distribution.[1] Hence, to observe the power of entanglement we need to allow both parties to use several measurement devices. The difference between parties that use entanglement and parties that only use shared randomness can then be observed by looking at the set of joint probability distributions conditioned on the choice of measurement devices.

Below we formalize the above notions. We first describe the general setting of bipartite correlations. Then we define classical and quantum correlations. It turns out that in the quantum setting we need to distinguish between finite-dimensional and infinite-dimensional Hilbert spaces: the tensor model and the commuting operator model.

3.2.1 The general setting

Formally, the setting of bipartite correlations is as follows. We have two parties, called Alice and Bob. Alice has several measurement devices (POVMs) labeled by elements from a finite set S, each with measurement outcomes taken from a finite set A. Similarly, Bob has measurement devices labeled by elements from a finite set T, each with outcomes taken from a finite set B. The parties do not know which measurement device the other party uses, and they do not communicate. For convenience we set $\Gamma = A\times B\times S\times T$ throughout. The probability that the parties' outcomes are $a \in A$ and $b \in B$ when they use measurement devices labeled by $s \in S$ and $t \in T$ is given by a *bipartite correlation* $P(a, b|s, t)$.

[1] Let us be more precise using the language introduced below. For all finite sets of outcomes A and B we have $C_{qc}(A \times B \times \{1\} \times \{1\}) = C_{loc}(A \times B \times \{1\} \times \{1\})$.

Definition 3.4. *Let* $\Gamma = A \times B \times S \times T$. *Then* $P \in \mathbb{R}^\Gamma$ *is a* bipartite correlation *if it satisfies* $P(a,b|s,t) \geq 0$ *for all* $(a,b,s,t) \in \Gamma$ *and* $\sum_{a,b} P(a,b|s,t) = 1$ *for all* $(s,t) \in S \times T$.

Which bipartite correlations $P = (P(a,b|s,t)) \in \mathbb{R}^\Gamma$ are possible depends on the additional resources available to the two parties Alice and Bob.

3.2.2 Classical correlations

We say that Alice and Bob *behave deterministically* if they each decide on their outcome through a function that maps measurement devices to outcomes. That is, Alice has a function $a : S \to A$ such that she answers $a(s)$ when using measurement device s, and similarly Bob has such a function $b : T \to B$. In terms of bipartite correlations this means the following. A correlation P is *deterministic* if there are functions $a : S \to A$ and $b : T \to B$ such that $P(a(s), b(t)|s,t) = 1$ for all $s \in S, t \in T$. Such a correlation is of the form $P(a,b|s,t) = P_A(a|s)\, P_B(b|t)$ for all $(a,b,s,t) \in \Gamma$, where $P_A = (P_A(a|s))$ and $P_B = (P_B(b|t))$ take their values in $\{0,1\}$ and satisfy

$$\sum_a P_A(a|s) = \sum_b P_B(b|t) = 1 \quad \text{for all} \quad (s,t) \in S \times T. \tag{3.1}$$

Deterministic correlations can be achieved without using any additional resources.

When the parties use local randomness the above functions P_A and P_B are convex combinations of 0/1-valued ones, that is, P_A and P_B take their values in $[0,1]$ and satisfy (3.1).

When the parties have access to shared randomness the resulting correlation P is a convex combination of deterministic correlations and P is said to be a *classical correlation.* The classical correlations form a polytope, the convex hull of the deterministic correlations, denoted $C_{loc}(\Gamma)$. Note that the dimension of $C_{loc}(\Gamma)$ equals $|\Gamma| - |S||T|$. Valid linear inequalities for $C_{loc}(\Gamma)$ are known as *Bell inequalities* [Bel64]. We will see an example of such an inequality in Section 3.3.2.

3.2.3 Quantum correlations

The tensor model. We now study the situation where Alice and Bob perform measurements on a quantum mechanical system. As above, let us say that the state of the quantum mechanical system is $\psi \in \mathbb{C}^d \otimes \mathbb{C}^d$, where Alice measures the first part of the system and Bob the second part.

The measurements of Alice and Bob are modeled by POVMs on their part of the state. For each $s \in S$ Alice has a POVM $\{E_s^a\}_{a \in A}$, and similarly Bob has a POVM $\{F_t^b\}_{b \in B}$ for each $t \in T$. The probability of observing outcome $(a,b) \in A \times B$ when using measurement devices s and t respectively is given by

$$P(a,b|s,t) = \psi^*(E_s^a \otimes F_t^b)\psi = \mathrm{Tr}((E_s^a \otimes F_t^b)\psi\psi^*). \tag{3.2}$$

Using the properties of the tensor product it follows that if the state ψ can be written as $\psi = \psi_A \otimes \psi_B$, then $P(a,b|s,t) = (\psi_A^* E_s^a \psi_A)(\psi_B^* F_t^b \psi_B)$ for all (a,b,s,t). That is, if ψ is not entangled (see Definition 3.2), then for any choice of measurement devices the resulting correlation P will be classical. On the other hand, if the state ψ is entangled, then it can be used to produce a nonclassical correlation P (see, e.g., [PR92, GP92]).

A correlation of the above form (3.2) is called a *quantum correlation*; when $\psi \in \mathbb{C}^d \otimes \mathbb{C}^d$ and $E_s^a, F_t^b \in \mathbb{C}^{d\times d}$ it is said to be *realizable in the tensor model in local dimension* d (or in *dimension* d^2), and we refer to $\big\{\psi, \{E_s^a\}, \{F_t^b\}\big\}$ as a *tensor operator representation* of P. Let $C_q^d(\Gamma)$ be the set of such correlations and define

$$C_q(\Gamma) = \bigcup_{d\in\mathbb{N}} C_q^d(\Gamma).$$

The following Lemma 3.5 shows that $C_q(\Gamma)$ is convex. The set $C_q^1(\Gamma)$ contains the deterministic correlations.[2] Hence, by Carathéodory's theorem, the lemma below implies that $C_{loc}(\Gamma) \subseteq C_q^c(\Gamma)$ holds for $c = \dim(C_{loc}(\Gamma)) + 1 = |\Gamma| - |S||T| + 1$. This shows that a quantum state can be used as an alternative to shared randomness.

Lemma 3.5. *Let* $P_1 \in C_q^d(\Gamma)$ *and* $P_2 \in C_q^{d'}(\Gamma)$ *and let* $\lambda \in [0, 1]$. *Then*

$$\lambda P_1 + (1 - \lambda)P_2 \in C_q^{d+d'}(\Gamma).$$

Proof. Let $\big\{\psi, \{E_s^a\}, \{F_t^b\}\big\}$ (in local dimension d) and $\big\{\phi, \{\widetilde{E}_s^a\}, \{\widetilde{F}_t^b\}\big\}$ (in local dimension d') be tensor operator representations of P_1 and P_2 respectively. Let us define $I_1 = [d]$ and $I_2 = [d']$, so that $\psi = (\psi_{ij})_{(i,j)\in I_1\times I_1}$ and $\phi = (\phi_{ij})_{(i,j)\in I_2\times I_2}$. Then we can say that $\mathbb{C}^d \oplus \mathbb{C}^{d'}$ has basis states labeled by the disjoint union $I_1 \sqcup I_2$ and correspondingly $(\mathbb{C}^d \oplus \mathbb{C}^{d'}) \otimes (\mathbb{C}^d \oplus \mathbb{C}^{d'})$ has basis states labeled by $(I_1 \sqcup I_2) \times (I_1 \sqcup I_2)$. Let $\Psi \in \mathbb{C}^{d+d'} \otimes \mathbb{C}^{d+d'}$ be such that $\Psi_{(i,j)} = \sqrt{\lambda}\,\psi_{ij}$ if $(i, j) \in I_1 \times I_1$, $\Psi_{(i,j)} = \sqrt{1-\lambda}\,\phi_{ij}$ if $(i, j) \in I_2 \times I_2$, and 0 otherwise. Then

$$\Big\{\Psi, \{E_s^a \oplus \widetilde{E}_s^a\}, \{F_t^b \oplus \widetilde{F}_t^b\}\Big\}$$

is a tensor operator representation of $\lambda P_1 + (1-\lambda)P_2$ in local dimension $d + d'$. □

A way to quantify the amount of entanglement used to realize a quantum correlation is by considering the smallest dimension needed to realize $P \in C_q(\Gamma)$ in the tensor model:

$$D_q(P) = \min\big\{d^2 : d \in \mathbb{N},\, P \in C_q^d(\Gamma)\big\}. \tag{3.3}$$

Computing $D_q(P)$ is NP-hard [Sta18]. As we have observed above, entanglement can be used as an alternative to shared randomness. As a result, the entanglement dimension of a *classical* correlation can be strictly larger than 1; informally, entanglement dimension does not capture “quantumness” perfectly. However, if the entanglement dimension is large enough then we can certify non-classical behavior. Indeed, since for $c = |\Gamma| - |S||T| + 1$ we have $C_{loc}(\Gamma) \subseteq C_q^c(\Gamma)$, one can certify that $P \notin C_{loc}(\Gamma)$ by showing that $D_q(P) > c^2$. This is a sufficient condition, but not a necessary condition: there exist Γ and $P \in C_q(\Gamma) \setminus C_{loc}(\Gamma)$ with $D_q(P) \leq c^2$. We

[2] In fact, $C_q^1(\Gamma)$ consists of the correlations obtained using only local randomness.

will see an example of such a correlation in Section 3.3.2. In Chapter 7 we propose a more refined measure for the amount of entanglement needed to realize a quantum correlation: the average entanglement dimension. That measure has the property that it is strictly larger than 1 *if and only if* the correlation is not classical.

If A, B, S, and T all contain at least two elements, then Bell [Bel64] showed that the inclusion $C_{loc}(\Gamma) \subseteq C_q(\Gamma)$ is strict; that is, quantum entanglement can be used to obtain nonclassical correlations. He did so by giving a linear inequality that is valid for the polytope $C_{loc}(\Gamma)$ but for which there exists a $P \in C_q(\Gamma)$ that violates it. This is why valid linear inequalities for $C_{loc}(\Gamma)$ are referred to as Bell inequalities. In Section 3.3.2 we give an example of a Bell inequality, arising from the CHSH game, that can be violated using bipartite quantum correlations. In this example $|A| = |B| = |S| = |T| = 2$.

The commuting operator model. In the above model we assumed a tensor product structure $\mathbb{C}^d \otimes \mathbb{C}^d$ on the underlying finite-dimensional Hilbert space. One could do the same in infinite dimensions by considering the Hilbert space $\mathcal{H} \otimes \mathcal{H}$ for a separable Hilbert space $\mathcal{H}$. The idea of a tensor structure on the Hilbert space and on the POVMs is that in such a structure there is no order in which the measurements take place: Alice cannot figure out if Bob has already measured his part of the state. In infinite dimensions we can also choose to encode this by enforcing the POVM of Alice to commute with Bob's POVM. The latter model is called the *commuting model* (or *relativistic field theory model*).

Formally, a correlation $P \in \mathbb{R}^\Gamma$ is called a *commuting quantum correlation* if it is of the form

$$P(a,b|s,t) = \langle \psi, X_s^a Y_t^b \psi \rangle = \mathrm{Tr}(X_s^a Y_t^b \psi\psi^*), \tag{3.4}$$

where $\{X_s^a\}_a$ and $\{Y_t^b\}_b$ are POVMs consisting of bounded operators on a separable Hilbert space $\mathcal{H}$, satisfying $[X_s^a, Y_t^b] = X_s^a Y_t^b - Y_t^b X_s^a = 0$ for all $(a,b,s,t) \in \Gamma$, and where ψ is a unit vector in $\mathcal{H}$. We refer to $\big\{\psi, \{X_s^a\}, \{Y_t^b\}\big\}$ as a *commuting operator representation* of P. Such a correlation is said to be *realizable in dimension* $d = \dim(\mathcal{H})$ *in the commuting model.* We denote the set of such correlations by $C_{qc}^d(\Gamma)$ and set $C_{qc}(\Gamma) = C_{qc}^\infty(\Gamma)$. Similar to Lemma 3.5 one can use a direct sum construction to show that the set $C_{qc}(\Gamma)$ is convex. The smallest dimension needed to realize a quantum correlation $P \in C_{qc}(\Gamma)$ is given by

$$D_{qc}(P) = \min\big\{d \in \mathbb{N} \cup \{\infty\} : P \in C_{qc}^d(\Gamma)\big\}. \tag{3.5}$$

If $P \in C_q^d(\Gamma)$ has a decomposition (3.2) with $d \times d$ matrices E_s^a, F_t^b, then P has a decomposition (3.4) with $d^2 \times d^2$ matrices $X_s^a = E_s^a \otimes I$ and $Y_t^b = I \otimes F_t^b$. This shows the inclusion

$$C_q^d(\Gamma) \subseteq C_{qc}^{d^2}(\Gamma),$$

and thus

$$D_{qc}(P) \leq D_q(P) \text{ for all } P \in C_q(\Gamma). \tag{3.6}$$

The difference between the tensor and commuting operator models. As said above we have the inclusion $C_q^d(\Gamma) \subseteq C_{qc}^{d^2}(\Gamma)$. Conversely, each finite-dimensional commuting quantum correlation can be realized in the tensor model, although not necessarily in the same dimension [Tsi06] (see, e.g., [DLTW08] for a proof). This shows that

$$C_q(\Gamma) = \bigcup_{d \in \mathbb{N}} C_{qc}^d(\Gamma) \subseteq C_{qc}(\Gamma). \tag{3.7}$$

Whether the two sets $C_q(\Gamma)$ and $C_{qc}(\Gamma)$ coincide is known as Tsirelson's problem [Tsi06]. This problem was settled in a recent breakthrough of Slofstra [Slo19], where he showed that $C_q(\Gamma)$ is not closed (for certain Γ). This indeed settled the problem because it was previously known that $C_{qc}(\Gamma)$ is closed [Fri12, Prop. 3.4]. More recently, in [DPP19] it was shown that $C_q(\Gamma)$ is not closed when $|A| \geq 2$, $|B| \geq 2$, $|S| \geq 5$, $|T| \geq 5$.

Since, for a fixed Γ, the set of all tensor operator representations in local dimension $d \in \mathbb{N}$ is compact, one sees that the set $C_q^d(\Gamma)$ is closed for all d. So, when $C_q(\Gamma)$ is not closed, the inclusions $C_q^d(\Gamma) \subset C_q(\Gamma)$ are all strict and therefore there is a sequence of quantum correlations $\{P_i\}_{i \in \mathbb{N}} \subseteq C_q(\Gamma)$ with entanglement dimension $D_q(P_i) \to \infty$ as $i \to \infty$.

In view of Equation (3.7) we know that the closure of $C_q(\Gamma)$ is a subset of $C_{qc}(\Gamma)$, for all Γ. Whether the closure of $C_q(\Gamma)$ equals $C_{qc}(\Gamma)$ for all Γ has been shown to be equivalent to having a positive answer to Connes' embedding conjecture in operator theory [JNP⁺11, Oza13]. This conjecture has been shown to have equivalent reformulations in many different fields; in Section 4.3 we will give an algebraic reformulation in terms of trace positivity of noncommutative polynomials due to Klep and Schweighofer [KS08].

Remarks. Above we chose to define bipartite quantum correlations using pure states and POVMs. Alternatively, one could define them using a mixed state and PVMs. Due to convexity the sets $C_q(\Gamma)$ and $C_{qc}(\Gamma)$ do not change if we replace the pure state $\psi\psi^*$ by a mixed state ρ in (3.2) and (3.4). It is shown in [SVW16] that this also does not change the parameter $D_q(P)$, but it is unclear whether or not $D_{qc}(P)$ might decrease. Another variation would be to use projective measurements (PVMs) instead of POVMs, where the operators are projectors instead of positive semidefinite matrices. This again does not change the sets $C_q(\Gamma)$ and $C_{qc}(\Gamma)$ [NC00], but the dimension parameters can be larger when restricting to PVMs.

3.3 Nonlocal games

We now view the bipartite correlation setting of the previous section from the perspective of optimization. We consider linear optimization over the sets of classical and quantum correlations. For a given objective function, a difference in the optimal value over these sets can be used to conclude that the set of classical correlations is strictly contained in the set of quantum correlations. With a simple change of language, linear optimization over the set of bipartite correlations arises naturally in games. We now refer to Alice and Bob as players. The game consists of a referee giving each of the players a question, afterwards the players have to respond with an answer. The players decide on a strategy before the game starts: to each question they associate a measurement device, and the outcomes of those measurements form their answers. To each pair of outcomes we associate a payoff (which may depend on the pair of questions). Linear optimization over the set of bipartite correlations then corresponds to Alice and Bob trying to maximize their expected payoff. When the two players are not allowed to communicate during the game, such a game is called a *nonlocal game*.

Formally, a *nonlocal game* G is defined by two finite sets of questions S and T, two finite sets of answers A and B, a probability distribution $\pi\colon S \times T \to [0,1]$ and a predicate[3] $f\colon A \times B \times S \times T \to \{0,1\}$. The predicate f determines the rules of the game: given question pair $(s,t) \in S \times T$, the pairs of answers $(a,b) \in A \times B$ such that $f(a,b,s,t) = 1$ are called *correct*, all other pairs are *wrong*. Alice and Bob receive a question pair $(s,t) \in S \times T$ with probability $\pi(s,t)$ and they win the game if their answers are correct. They know the game parameters π and f, but they do not know each other's questions, and they cannot communicate after they receive their questions. Their answers (a,b) are determined according to some correlation $P \in \mathbb{R}^\Gamma$, called their *strategy*, on which they may agree before the start of the game, and which can be classical or quantum depending on whether P belongs to $C_{loc}(\Gamma)$, $C_q(\Gamma)$, or $C_{qc}(\Gamma)$. Then their corresponding winning probability is given by

$$\sum_{(s,t)\in S\times T} \pi(s,t) \sum_{(a,b)\in A\times B} P(a,b|s,t) f(a,b,s,t). \tag{3.8}$$

A strategy P is called *perfect* if the above winning probability is equal to one, that is, if for all $(a,b,s,t) \in \Gamma$ we have

$$\big(\pi(s,t) > 0 \quad \text{and} \quad f(a,b,s,t) = 0\big) \quad \Longrightarrow \quad P(a,b|s,t) = 0. \tag{3.9}$$

In other words, the probability of giving a wrong answer equals zero.

Computing the maximum winning probability of a nonlocal game is the problem of finding a bipartite correlation that maximizes (3.8). This is an instance of linear optimization (of the function (3.8)) over $C_{loc}(\Gamma)$ in the classical setting, and over $C_q(\Gamma)$ or $C_{qc}(\Gamma)$ in the quantum setting. Since the inclusion $C_{loc}(\Gamma) \subseteq C_q(\Gamma)$ can be strict, the maximum winning probability can be higher when the parties have access to entanglement. A famous example of such a game is due to Clauser, Horne, Shimony, and Holt [CHSH69]; we will discuss this game in detail in Section 3.3.2. In fact there are nonlocal games that can be won with probability 1 by using entanglement, but only with probability strictly less than 1 in the classical setting; see for example the Mermin-Peres magic square game [Mer90, Per90]. In general it is hard to determine the maximum (quantum) winning probability of a game[4]: as we will see below, certain hard combinatorial problems such as max-cut can be phrased

[3] To make the above analogy with linear optimization correct one would need to consider a real-valued function f. For this thesis it suffices to only consider 0/1-valued functions f.

[4] Slofstra showed that for a certain class of games called *linear system games* it is undecidable to determine if such a game has a perfect strategy in $C_{qc}(\Gamma)$ [Slo16], or in $C_q(\Gamma)$ or its closure $\mathrm{cl}(C_q(\Gamma))$ [Slo19]. In particular, the problem of determining whether the maximum winning probability of a game equals 1 over $C_{qc}(\Gamma)$ is undecidable. As we point out later, this implies that there is no general stopping criterion for the noncommutative sum-of-squares hierarchy that we will mention in Chapter 4.

as linear optimization over the set of classical correlations. However, for certain classes of games and strategies determining the maximum winning probability becomes easy. Below we introduce one such class, called *XOR games*, for which the maximum winning probability when using quantum strategies can be determined using semidefinite programming.

3.3.1 XOR games

A nonlocal game G is called an *XOR game* when Alice and Bob each output a single bit, that is, $A = B = \{0,1\}$, and the predicate f is of the form $f(a,b,s,t) = 1$ if and only if $a \oplus b = g(s,t)$ for some function $g : S \times T \to \{0,1\}$. In other words, the rules of the game are such that whether a pair of answers is correct or wrong only depends on the logical XOR of the answers. XOR games are special in the sense that the maximum winning probability over quantum strategies can be expressed using semidefinite programming, as we explain below.

Let $P \in C_q(\Gamma)$ be a quantum strategy and suppose that $\psi, \{E_s^a\}, \{F_t^b\}$ are as in (3.2), that is, $P(a,b|s,t) = \psi^*(E_s^a \otimes F_t^b)\psi$ for all a,b,s,t. Then one can verify[5] that the winning probability of P in the XOR game G (cf. (3.8)) can be written as

$$\sum_{(s,t)\in S\times T} \pi(s,t) \sum_{a,b\in\{0,1\}} f(a,b,s,t)\,\psi^*(E_s^a \otimes F_t^b)\psi$$
$$= \frac{1}{2} + \frac{1}{2} \sum_{(s,t)\in S\times T} \pi(s,t)(-1)^{g(s,t)}\,\psi^*\big((E_s^0 - E_s^1)\otimes(F_t^0 - F_t^1)\big)\psi.$$

This suggests changing variables and working with the matrices $E_s = E_s^0 - E_s^1$ and $F_t = F_t^0 - F_t^1$: We consider

$$\sum_{(s,t)\in S\times T} \pi(s,t)(-1)^{g(s,t)}\psi^*(E_s \otimes F_t)\psi. \tag{3.10}$$

This change of variables does not lose any information: A Hermitian matrix whose eigenvalues lie in $[-1,1]$ can be written, uniquely, as the difference between two Hermitian positive semidefinite matrices that sum to the identity.

Notice that one possible strategy for Alice and Bob is to each base their answer on an unbiased coin flip, this strategy has a winning probability of 1/2. The above quantity (3.10) represents the *bias* that the strategy P has towards winning. It is equal to the probability of winning minus the probability of losing.

The problem of maximizing the bias of the game G over quantum strategies $P \in C_q(\Gamma)$ is thus

$$\max\Big\{ \sum_{(s,t)\in S\times T} \pi(s,t)(-1)^{g(s,t)}\,\psi^*(E_s \otimes F_t)\psi \;:\; d\in\mathbb{N},\ \psi\in\mathbb{C}^d\otimes\mathbb{C}^d \text{ unit},$$
$$E_s, F_t \in \mathrm{H}_+^d,\ (E_s)^2 = I = (F_t)^2 \text{ for all } s\in S, t\in T\Big\}. \tag{3.11}$$

[5] Here we use the identity $f(a,b,s,t) = \frac{1}{2} + \frac{1}{2}(-1)^{a+b+g(s,t)}$ for all $a,b\in\{0,1\}, s\in S, t\in T$.

We can observe that $\psi^*(E_s \otimes F_t)\psi = \langle (E_s \otimes I)\psi, (I \otimes F_t)\psi \rangle$ and that both vectors $x_s = (E_s \otimes I)\psi$ and $y_t = (I \otimes F_t)\psi$ have norm at most one. It follows that the quantum bias of the game (3.11) can be upper bounded by the semidefinite program

$$\max_{\|x_s\|,\|y_t\| \leq 1} \sum_{(s,t)\in S\times T} \pi(s,t)(-1)^{g(s,t)} \langle x_s, y_t \rangle. \tag{3.12}$$

To see that this indeed an SDP, let us define the $S \times T$ matrix B with entries

$$B_{s,t} = \pi(s,t)(-1)^{g(s,t)}. \tag{3.13}$$

Then (3.12) can be written as the following SDP in primal form:

$$\begin{aligned} \max \quad & \frac{1}{2}\Big\langle \begin{pmatrix} 0 & B \\ B^T & 0 \end{pmatrix}, X \Big\rangle \qquad (3.14) \\ \text{s.t.} \quad & X \in \mathrm{S}_+^{S\cup T} \\ & X_{ii} \leq 1 \quad \text{for } i \in S \cup T. \end{aligned}$$

For notational convenience, here we assumed that the sets S and T are disjoint.

Remarkably, Tsirelson [Tsi87] showed that the quantum bias of the game in fact equals the value of the SDP (3.14). That is, from a solution $X = \mathrm{Gram}(\{x_s\}, \{y_t\})$ to the semidefinite program (3.14) one can construct a quantum strategy P that has a bias equal to the value of the SDP. We recall this construction in Chapter 6, Theorem 6.13. In that chapter we exploit this connection between optimal strategies for XOR games and semidefinite programming to construct quantum correlations which require a lot of entanglement.

What about the bias of an XOR game over classical strategies? By considering the bias of a deterministic strategy, one can show that the maximum classical bias can be computed by restricting to 1-dimensional vectors in the above SDP:

$$\max_{x_s,y_t \in \{\pm 1\}} \sum_{(s,t)\in S\times T} \pi(s,t)(-1)^{g(s,t)} x_s y_t. \tag{3.15}$$

Notice that this problem is exactly the max-cut problem in a complete bipartite graph where the edge weights are given by $\pi(s,t)(-1)^{g(s,t)}$, which is known to be an NP-hard problem [MRR03, Lem. 3].

3.3.2 The Clauser-Horne-Shimony-Holt game

We now illustrate the above concepts and the relation to this thesis via a classical example of an XOR game: the CHSH game [CHSH69]. In this game each player both receives and responds with a single bit, that is, $A = B = S = T = \{0,1\}$. The distribution π is the uniform distribution on $S \times T$. The rules of the game are such that the players win if the logical XOR of their answers equals the logical AND of their questions. That is, $g : \{0,1\}^2 \to \{0,1\}$ is the function $g(s,t) = st$, and $f(a,b,s,t) = 1$ if and only if $a \oplus b = g(s,t) = st$.

Using the formulation of Equation (3.15), it is easy to see that the classical bias is at most $\frac{1}{2}$. On the other hand, the quantum bias of the game is at least

$\frac{1}{\sqrt{2}}$ (which is strictly larger than $\frac{1}{2}$) which can be seen from the following feasible solution to (3.12):

$$x_0 = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}, \quad x_1 = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}, \quad y_0 = \begin{pmatrix}1\\0\end{pmatrix}, \quad y_1 = \begin{pmatrix}0\\1\end{pmatrix}.$$

Indeed, we have

$$\sum_{(s,t)\in S\times T} \pi(s,t)(-1)^{g(s,t)}\langle x_s, y_t\rangle = \frac{1}{4}\big(\langle x_0, y_0\rangle + \langle x_0, y_1\rangle + \langle x_1, y_0\rangle - \langle x_1, y_1\rangle\big) = \frac{1}{\sqrt{2}}.$$

Using the dual of the SDP (3.14) one can show that the quantum bias of the CHSH game is in fact equal to $1/\sqrt{2}$. Translating this back to the language of winning probabilities, the CHSH game can be won with probability at most $\frac{3}{4} = 0.75$ using classical strategies, but it can be won with probability $\frac{1}{2} + \frac{1}{2\sqrt{2}} \approx 0.85$ using quantum strategies. Note that the classical winning probability is attained for the deterministic strategy where Alice and Bob always output 0. The difference between the maximum quantum and classical winning probabilities suggests a way to test whether classical mechanics is the correct model for the physical world: come up with an experiment that wins the CHSH game with probability strictly larger than 0.75. This is precisely what has been done in laboratories around the world, eventually leading to the first loop-hole free Bell inequality violation [Hea15].

What about the smallest entanglement dimension needed to win the CHSH game with probability $\frac{1}{2} + \frac{1}{2\sqrt{2}}$? Tsirelson showed that the SDP for the quantum bias of an XOR game is tight by constructing a quantum strategy from unit vectors x_s, y_t ($s \in S, t \in T$) achieving the same bias. His construction, which we recall in Theorem 6.13, combined with the above 2-dimensional vectors, shows that this winning probability can be achieved using a strategy with local dimension equal to 2.[6] In this case it is easy to see that the same bias cannot be realized in a smaller local dimension (since such a strategy would be classical).

In Chapter 6 we construct XOR games with which we can certify that certain quantum correlations have a large minimal entanglement dimension $D_q(P)$. The techniques used there combine the fundamental work of Tsirelson with the duality theory of semidefinite programming and the theory of universal rigidity. As an example of hitting a small object with a huge hammer, let us give an alternative way to show that 2 is the smallest local dimension in which the quantum strategy corresponding to x_0, x_1, y_0, y_0 can be realized. One can use Theorem 6.6 and Theorem 6.10 (pick $\lambda_0 = \lambda_1 = \mu_0 = \mu_1$) to show that the corresponding correlation matrix is extreme, the lower bound on the local dimension then follows from Corollary 6.17.

[6] The fact that there exists an optimal strategy with local dimension equal to 2 was already known [CHSH69].

3.4 Relation to completely positive semidefinite matrices

The study of completely positive semidefinite matrices and the cpsd-rank is strongly motivated by the study of the set of bipartite quantum correlations. Here we mention two connections (i) and (ii). The first one is that $C_q(\Gamma)$ can be seen as the projection of an affine slice of the cone CS_+.

(i) For $\Gamma = A\times B\times S\times T$, there exists an affine subspace $\mathcal{L}$ and a linear projection π such that

$$C_q(\Gamma) = \pi\big(\mathrm{CS}_+^{(A\times S)\sqcup(B\times T)} \cap \mathcal{L}\big).$$

This connection, which we state formally in Theorem 3.6 below, can be found in [SV17, Thm. 3.2] (see also [MR14]). In Chapter 6 this link allows us to construct CS_+-matrices with large complex completely positive semidefinite rank by finding quantum correlations that cannot be realized in a small local dimension.

Theorem 3.6 ([SV17, Thm. 3.2]). *Let $P = \big(P(a,b|s,t)\big) \in \mathbb{R}^\Gamma$ be a bipartite correlation. Then, P is a quantum correlation that can be realized in local dimension d if and only if there exists a completely positive semidefinite matrix M, with rows and columns indexed by the disjoint union $(A\times S)\sqcup(B\times T)$, satisfying the following conditions:*

$$\text{cpsd-rank}_{\mathbb{C}}(M) \le d, \tag{3.16}$$

$$M_{(a,s),(b,t)} = P(a,b|s,t) \quad \text{for all} \quad a\in A, b\in B, s\in S, t\in T, \tag{3.17}$$

and

$$\sum_{a\in A, b\in B} M_{(a,s),(b,t)} = \sum_{a,a'\in A} M_{(a,s),(a',s')} = \sum_{b,b'\in B} M_{(b,t),(b',t')} = 1 \tag{3.18}$$

for all $s,s'\in S$ and $t,t'\in T$.

When the two parties have the same question sets ($S=T$) and the same answer sets ($A=B$), a bipartite correlation $P\in\mathbb{R}^\Gamma$ is called *synchronous* if it satisfies

$$P(a,b|s,s) = 0 \;\text{ for all } s\in S \text{ and } a\neq b\in A. \tag{3.19}$$

In other words, a bipartite correlation P is synchronous if equal inputs imply equal outputs. Note that synchronicity does not a priori imply that Alice and Bob use the same set of measurement devices. However, in the proof of Proposition 3.7 we will see that Alice and Bob can always realize a synchronous quantum correlation P using related measurement devices: $F_t^b = (E_t^b)^T$. We have seen an example of a synchronous quantum correlation in the introduction to Section 3.2: Alice and Bob each measuring one qubit of an EPR-pair in the computational basis is an example of a synchronous correlation $P\in\mathbb{R}^\Gamma$ (where $\Gamma = \{0,1\}\times\{0,1\}\times\{1\}\times\{1\}$). The sets of synchronous (commuting) quantum correlations are denoted $C_{q,s}(\Gamma)$ and $C_{qc,s}(\Gamma)$, respectively. We have $C_{q,s}(\Gamma) \subseteq C_{qc,s}(\Gamma)$ and the set $C_{qc,s}(\Gamma)$ is closed.[7]

To a synchronous correlation $P \in \mathbb{R}^\Gamma$ ($\Gamma = A \times A \times S \times S$) we can associate an $|A| \times |S|$ matrix M_P whose entries are defined by

$$(M_P)_{(s,a),(t,b)} = P(a,b|s,t) \text{ for } (a,b,s,t) \in \Gamma. \tag{3.20}$$

In Proposition 3.7 below we show that the matrix M_P can be used to derive a more economical form of Theorem 3.6: For a bipartite correlation P we have that P is a synchronous quantum correlation realizable in dimension d^2 if and only if $M_P \in \mathrm{CS}_+^{A\times S}$ and $\text{cpsd-rank}_{\mathbb{C}}(M_P) \leq d$. Hence $C_{q,s}(\Gamma)$ can be seen as an affine section of the CS_+ cone:

(ii) For $\Gamma = A \times A \times S \times S$, there exists an affine subspace[8] $\mathcal{L}_s$ such that

$$C_{q,s}(\Gamma) = \mathrm{CS}_+^{A\times S} \cap \mathcal{L}_s.$$

Moreover the CS_+-matrix M_P associated to a synchronous quantum correlation P through (3.20) satisfies $D_q(P) = \text{cpsd-rank}_{\mathbb{C}}(M_P)^2$.

In [PSS+16] the set of synchronous (commuting) quantum correlations has been characterized using C^*-algebras (see Theorem 8.3). Here we combine their proof technique with that of [SV17] (see also [MR16b]) to derive Proposition 3.7. A key ingredient of the proof below is the following identity. Let $K, X, Y \in \mathbb{C}^{d\times d}$ then

$$\mathrm{vec}(K)^*(X \otimes Y)\mathrm{vec}(K) = \mathrm{vec}(K)^*\mathrm{vec}(XKY^T) = \mathrm{Tr}\big(K^*XKY^T\big). \tag{3.21}$$

See for example [Wat11, Sec. 2.4] for the first identity.

Proposition 3.7 ([GdLL18, Prop. 1]). *Let $P = \big(P(a,b|s,t)\big) \in \mathbb{R}^\Gamma$ be a synchronous bipartite correlation. Then, P is a quantum correlation that can be realized in the tensor model in local dimension d if and only if the $|A| \times |S|$ matrix M_P associated to P through* (3.20) *is completely positive semidefinite and has* $\text{cpsd-rank}_{\mathbb{C}}(M_P) \leq d$. *Therefore, for a synchronous quantum correlation P we have that $D_q(P) = \text{cpsd-rank}_{\mathbb{C}}(M_P)^2$.*

Proof. Suppose first that $P \in C_{q,s}(\Gamma)$ and that $\big\{\psi, \{E_s^a\}, \{F_t^b\}\big\}$ is a realization of P in local dimension d as in (3.2). We will show that the matrix M_P defined in (3.20) is completely positive semidefinite and has $\text{cpsd-rank}_{\mathbb{C}}(M_P) \leq d$.

Taking the Schmidt decomposition of ψ, there exist nonnegative scalars $\{\lambda_i\}$ and orthonormal bases $\{u_i\}$ and $\{v_i\}$ of $\mathbb{C}^d$ such that $\psi = \sum_{i=1}^d \sqrt{\lambda_i}\, u_i \otimes v_i$.[9] Let

[7] The synchronous correlation sets are already rich enough in the sense that it is still the case that Connes' embedding conjecture holds if and only if $\mathrm{cl}(C_{q,s}(\Gamma)) = C_{qc,s}(\Gamma)$ for all Γ [DP16, Thm. 3.7]. The quantum graph parameters discussed in Section 8 will be defined through optimization problems over synchronous quantum correlations.

[8] This affine subspace corresponds to the equalities $\sum_{a\in A, b\in A} M_{(a,s),(b,t)} = 1$ (for all $s,t \in S$) that we have seen before in (3.18).

[9] The Schmidt decomposition $\psi = \sum_{i=1}^d \sqrt{\lambda_i}\, u_i \otimes v_i$ of $\psi \in \mathbb{C}^d \otimes \mathbb{C}^d$ can be viewed as the singular value decomposition $\sum_{i=1}^d \sqrt{\lambda_i} u_i v_i^*$ of the matrix $A \in \mathbb{C}^{d\times d}$ for which $\psi = \sum_{i,j=1}^d A_{ij} e_i \otimes e_j$.

U be the unitary matrix that represents the basis transformation from $\{v_i\}$ to $\{u_i\}$: $u_i = Uv_i$ for all i. If we replace ψ by $\sum_{i=1}^d \sqrt{\lambda_i}\, v_i \otimes v_i$ and E_s^a by $U^* E_s^a U$, then $\big\{ \sum_{i=1}^d \sqrt{\lambda_i}\, v_i \otimes v_i, \{E_s^a\}, \{F_t^b\}\big\}$ still realizes P and is of the same dimension d.

Given such a realization $\big\{\psi = \sum_{i=1}^d \sqrt{\lambda_i}\, v_i \otimes v_i, \{E_s^a\}, \{F_t^b\}\big\}$ of P, we define the matrices

$$K = \sum_{i=1}^d \sqrt{\lambda_i}\, v_i v_i^*, \quad X_s^a = K^{1/2} E_s^a K^{1/2}, \quad Y_t^b = K^{1/2} (F_t^b)^T K^{1/2}.$$

Notice that $\mathrm{vec}(K) = \psi$. Moreover, we have the identity

$$\mathrm{vec}(K)^* (E_s^a \otimes F_t^b) \mathrm{vec}(K) = \mathrm{Tr}(K E_s^a K (F_t^b)^T) = \mathrm{Tr}(K^{1/2} E_s^a K^{1/2} K^{1/2} (F_t^b)^T K^{1/2}), \tag{3.22}$$

where we use the identity (3.21) in the first equality. Hence, we have

$$P(a,b|s,t) = \langle X_s^a, Y_t^b\rangle \quad \text{for all} \quad a,b,s,t,$$

and

$$\langle K, K\rangle = 1, \quad \sum_a X_s^a = \sum_b Y_t^b = K \quad \text{for all} \quad s,t.$$

For any $s \in S$, as P is synchronous we have $1 = \sum_{a,b} P(a,b|s,s) = \sum_a P(a,a|s,s)$. Using the Cauchy–Schwarz inequality we see that

$$\begin{aligned} 1 = \sum_a P(a,a|s,s) &= \sum_a \langle X_s^a, Y_s^a\rangle \le \sum_a \langle X_s^a, X_s^a\rangle^{1/2} \langle Y_s^a, Y_s^a\rangle^{1/2} \\ &\le \Big(\sum_a \langle X_s^a, X_s^a\rangle\Big)^{1/2} \Big(\sum_a \langle Y_s^a, Y_s^a\rangle\Big)^{1/2} \\ &\le \Big\langle \sum_a X_s^a, \sum_a X_s^a\Big\rangle^{1/2} \Big\langle \sum_a Y_s^a, \sum_a Y_s^a\Big\rangle^{1/2} = \langle K, K\rangle = 1. \end{aligned}$$

Thus all inequalities above are equalities. The first inequality being an equality shows that there exist $\alpha_{s,a} \ge 0$ such that $X_s^a = \alpha_{s,a} Y_s^a$ for all a,s. The second inequality being an equality shows that there exist $\beta_s \ge 0$ such that $\|X_s^a\| = \beta_s \|Y_s^a\|$ for all a,s. Hence,

$$\beta_s \|Y_s^a\| = \|X_s^a\| = \|\alpha_{s,a} Y_s^a\| = \alpha_{s,a} \|Y_s^a\| = \alpha_{s,a} \|Y_s^a\| \quad \text{for all} \quad a,s,$$

which shows $X_s^a = \beta_s Y_s^a$ for all a,s. Since $\sum_a X_s^a = K = \sum_a Y_s^a$, we have $\beta_s = 1$ for all s. Thus $X_s^a = Y_s^a$ for all a,s. Therefore,

$$(M_P)_{(s,a),(t,b)} = \langle X_s^a, X_t^b\rangle \quad \text{for all} \quad a,b,s,t,$$

which shows M_P is completely positive semidefinite with $\text{cpsd-rank}_{\mathbb{C}}(M_P) \le d$.

For the other direction we suppose $\{X_s^a\} \subset \mathrm{H}^{\text{cpsd-rank}(M_P)}$ are Hermitian positive semidefinite matrices such that $(M_P)_{(s,a),(t,b)} = \langle X_s^a, X_t^b\rangle$ for all a,s,t,b. Then,

$$1 = \sum_{a,b} P(a,b|s,t) = \sum_{a,b} \langle X_s^a, X_t^b\rangle = \Big\langle \sum_a X_s^a, \sum_b X_t^b\Big\rangle \quad \text{for all} \quad s,t.$$

Therefore, the equality case of the Cauchy-Schwarz inequality shows the existence of a matrix K such that $K = \sum_a X_s^a$ for all s. We have $\langle K, K \rangle = 1$ and thus $\mathrm{vec}(K)$ is a unit vector. Moreover, since the factorization of M_P is chosen of smallest possible size, the matrix K is invertible. Set $E_s^a = K^{-1/2} X_s^a K^{-1/2}$ for all s, a, so that $\sum_a E_s^a = I$ for all s. Then, using again (3.22) we obtain

$$P(a,b|s,t) = (M_P)_{(s,a),(t,b)} = \langle X_s^a, X_t^b \rangle = \mathrm{vec}(K)^* (E_s^a \otimes (E_t^b)^T) \mathrm{vec}(K),$$

which shows that P has a realization of local dimension $\text{cpsd-rank}_{\mathbb{C}}(M_P)$. □

What happens if instead of requiring M_P to be completely positive semidefinite, we require M_P to be completely positive? Then we exactly recover classical synchronous correlations! To see this, we first use Lemma 3.5 to show that a classical bipartite correlation is a quantum correlation which has a tensor operator representation of a specific form:

Lemma 3.8. *Let $P = \big(P(a,b|s,t)\big) \in \mathbb{R}^\Gamma$ be a bipartite correlation. Then, P is a classical bipartite quantum correlation if and only if P has a tensor operator representation $\big\{\psi, \{E_s^a\}, \{F_t^b\}\big\}$ for which $\psi = \sum_{i\in[d]} \lambda_i e_i \otimes e_i$ for some unit vector $\lambda \in \mathbb{R}_+^d$ (where $d \in \mathbb{N}$), and the POVMs $\{E_s^a\}_a$ and $\{F_t^b\}_b$ all consist of diagonal matrices.*

Proof. We first show the 'only if' statement. As observed before Lemma 3.5, any deterministic bipartite correlation has a tensor operator representation in local dimension $d = 1$, which implies that any convex combination of deterministic strategies (and thus any classical correlation) has a tensor operator representation of the desired form (by Lemma 3.5).

We now show the 'if' statement. Let P be a quantum correlation which has a tensor operator representation $\big\{\psi, \{E_s^a\}, \{F_t^b\}\big\}$ for which $\psi = \sum_{i\in[d]} \lambda_i e_i \otimes e_i$ for some unit vector $\lambda \in \mathbb{R}_+^d$ (where $d \in \mathbb{N}$), and the POVMs $\{E_s^a\}_a$ and $\{F_t^b\}_b$ all consist of diagonal matrices. Then we have, for all $(a,b,s,t) \in \Gamma$, that

$$\begin{aligned} P(a,b|s,t) &= \psi^*(E_s^a \otimes F_t^b)\psi \\ &= \sum_{i=1}^d \lambda_i^2 (E_s^a)_{ii} (F_t^b)_{ii}. \end{aligned}$$

Let us now define the correlations $P_i = (P_i(a,b|s,t)) = \big((E_s^a)_{ii}(F_t^b)_{ii}\big)$ for $i \in [d]$. From the fact that $\{E_s^a\}_a$ and $\{F_t^b\}_b$ are diagonal POVMs, it follows that each P_i is a classical bipartite correlation using only local randomness (see Section 3.2.2). Furthermore, since ψ is a unit vector we have that $\sum_{i=1}^d \lambda_i^2 = 1$. This shows that $P = \sum_{i=1}^d \lambda_i^2 P_i$ is a convex combination of classical bipartite correlations and therefore P is itself classical. □

Using the same proof technique that we used for Proposition 3.7, we arrive at the following.

Proposition 3.9. *Let $P = \big(P(a,b|s,t)\big) \in \mathbb{R}^{\Gamma}$ be a synchronous bipartite correlation. Then, P is a classical correlation if and only if the $|A| \times |S|$ matrix M_P associated to P through* (3.20) *is completely positive.*

Proof. We follow the same structure as the proof of Proposition 3.7. In both directions we will use the observation that a matrix M is completely positive if and only if it has a symmetric factorization $M = \big(\mathrm{Tr}(D_iD_j)\big)$ by diagonal positive semidefinite matrices D_i (in some finite dimension).

Suppose first that $P \in C_{loc,s}(\Gamma)$ and that $\big\{\psi, \{E_s^a\}, \{F_t^b\}\big\}$ is a tensor operator realization of P, where ψ is of the form $\psi = \sum_{i\in[d]} \lambda_i e_i \otimes e_i$ for some unit vector $\lambda \in \mathbb{R}_+^d$ (where $d \in \mathbb{N}$), and the POVMs $\{E_s^a\}_a$ and $\{F_t^b\}_b$ all consist of diagonal matrices (which we may assume by Lemma 3.8). We will show that the matrix M_P defined in (3.20) is completely positive.

Given such a realization $\big\{\psi = \sum_{i=1}^d \lambda_i\, e_i \otimes e_i, \{E_s^a\}, \{F_t^b\}\big\}$ of P, we define the (diagonal!) matrices

$$K = \mathrm{Diag}(\lambda), \quad X_s^a = K^{1/2}E_s^aK^{1/2}, \quad Y_t^b = K^{1/2}(F_t^b)^TK^{1/2}.$$

Notice that $\mathrm{vec}(K) = \psi$. Moreover, using Equation (3.22) we have the identity

$$\mathrm{vec}(K)^*(E_s^a \otimes F_t^b)\mathrm{vec}(K) = \mathrm{Tr}(X_s^aY_t^b) = P(a,b|s,t) \qquad \text{for all } a,b,s,t. \tag{3.23}$$

Also, it follows from the fact that ψ is a unit vector and that $\{E_s^a\}_a$ and $\{F_t^b\}_b$ are POVMs that

$$\langle K, K\rangle = 1, \quad \sum_a X_s^a = \sum_b Y_t^b = K \qquad \text{for all } s,t.$$

As in the proof of Proposition 3.7 we can now use synchronicity of P and the Cauchy-Schwarz inequality to show that $X_s^a = Y_s^a$ for all a, s. Therefore Equation (3.23) shows that,

$$(M_P)_{(s,a),(t,b)} = P(a,b|s,t) = \langle X_s^a, X_t^b\rangle \qquad \text{for all } a,b,s,t,$$

which shows that M_P is completely positive since the matrices X_s^a are diagonal and positive semidefinite.

For the other direction suppose $M_P \in \mathrm{CP}^{A\times S}$, let $d = \text{cp-rank}(M_P)$ and suppose $\{X_s^a\} \subset \mathrm{S}_+^d$ are diagonal positive semidefinite matrices such that $(M_P)_{(s,a),(t,b)} = \langle X_s^a, X_t^b\rangle$ for all a,s,t,b. Then,

$$1 = \sum_{a,b} P(a,b|s,t) = \sum_{a,b}\langle X_s^a, X_t^b\rangle = \Big\langle \sum_a X_s^a, \sum_b X_t^b\Big\rangle \qquad \text{for all } s,t.$$

Therefore, the equality case of the Cauchy-Schwarz inequality shows the existence of a (diagonal!) matrix K such that $K = \sum_a X_s^a$ for all s. We have $\langle K, K\rangle = 1$ and thus $\mathrm{vec}(K)$ is a unit vector. Moreover, since the factorization of M_P is chosen of smallest possible size, the matrix K is invertible. Set $E_s^a = K^{-1/2}X_s^aK^{-1/2}$ for all s, a, so that $\sum_a E_s^a = I$ for all s. Then, using again (3.22) we obtain

$$\begin{aligned} P(a,b|s,t) = (M_P)_{(s,a),(t,b)} = \langle X_s^a, X_t^b\rangle &= \mathrm{vec}(K)^*(E_s^a \otimes (E_t^b)^T)\mathrm{vec}(K) \\ &= \mathrm{vec}(K)^*(E_s^a \otimes E_t^b)\mathrm{vec}(K) \end{aligned}$$

which shows that P has a realization of local dimension d. It now remains to observe that all matrices K, E_s^a, X_s^a are diagonal. Therefore K is of the form $\mathrm{Diag}(\lambda)$ where $\lambda \in \mathbb{R}_+^d$ and $\sum_{i=1}^d \lambda_i^2 = 1$, and the vector ψ is of the form $\psi = \mathrm{vec}(K) = \sum_{i=1}^d \lambda_i \, e_i \otimes e_i$. It thus follows from Lemma 3.8 that P is a classical correlation. □

Chapter 4

Noncommutative polynomial optimization

In this thesis we will use techniques from commutative and noncommutative polynomial optimization to study matrix factorization ranks (Chapter 5), the amount of entanglement needed to realize a bipartite quantum correlation (Chapter 7), and quantum analogues of graph parameters (Chapter 8). Since these techniques are so fundamental to the first part of this thesis we introduce them here. We discuss known convergence and flatness results for commutative and tracial polynomial optimization. Although the commutative case was developed first, here we treat the commutative and tracial cases together. Tracial optimization is an adaptation of eigenvalue optimization as developed in [PNA10], but here we only discuss the commutative and tracial cases, as these are most relevant to our work. We will view polynomial optimization from the "moment side" since that is most relevant to our applications; that is, we rely on properties of linear functionals rather than real algebraic results on sums of squares.

This chapter is based on the appendix of the paper "Lower bounds on matrix factorization ranks via noncommutative polynomial optimization", by S. Gribling, D. de Laat, and M. Laurent [GdLL19], with the exception of the new Section 4.3 which highlights some differences between commutative and noncommutative polynomial optimization.

Before diving into the technical details, let us start with a motivating example.

Example 4.1. Consider the following optimization problem:

$$\sup\Big\{\frac{1}{d}\,\mathrm{Tr}(XYX)\;:\; d\in\mathbb{N},\, X,Y\in \mathrm{H}^d_+ \text{ s.t. } \|X\|,\|Y\|\leq 1\Big\}. \tag{4.1}$$

That is, we want to maximize the normalized trace of the product XYX where X and Y are $d\times d$ Hermitian positive semidefinite matrices (for some $d\in\mathbb{N}$) that have operator norm at most 1. How can we solve this problem?

In this case we can easily solve it by hand: the optimal value is equal to 1. To see that the optimal value is at least 1, pick $d = 1$ and $X = Y = 1$. On the other hand, since both X and Y have operator norm at most 1, the product XYX has operator norm at most 1. That means that the eigenvalues of XYX all have absolute value at most 1. In particular, the normalized trace, which is the average of the eigenvalues, of XYX is at most 1.

Problem (4.1) is an example of a tracial polynomial optimization problem: as we see below, it is of the form minimize/maximize the normalized trace of a polynomial subject to polynomial inequalities. Often it is not so easy to solve such problems by hand. We need a more systematic approach. Let us give a second proof that the optimal value is at most 1. For this let us first rewrite the feasible region of problem (4.1) using polynomial inequalities:

$$\sup\Big\{\frac{1}{d}\operatorname{Tr}(XYX) \,:\, d \in \mathbb{N},\, X, Y \in \mathrm{H}_+^d \text{ s.t. } I - X^2 \succeq 0,\, I - Y \succeq 0, I + Y \succeq 0\Big\}.$$

This problem is indeed equivalent to problem (4.1): The polynomial inequality $I - X^2 \succeq 0$ implies that the eigenvalues of X lie in the interval $[-1, 1]$ and the polynomial inequalities $I - Y \succeq 0$ and $I + Y \succeq 0$ together imply the same for Y. Here we chose two different ways to encode that the eigenvalues of a matrix lie in the interval $[-1, 1]$, which shows that the encoding is not unique; the reason we did so will become apparent later on. We will use some basic properties of the matrix trace, namely

(i) $\frac{1}{d}\operatorname{Tr}(I) = 1$,

(ii) for a Hermitian positive semidefinite matrix A we have $\operatorname{Tr}(A) \geq 0$,

(iii) the trace is additive.

These properties combined show that we can certify that the optimal value is at most 1 by giving an algebraic proof that $I - XYX$ is positive semidefinite for all feasible X and Y. Indeed, then we would have

$$0 \overset{\text{(ii)}}{\leq} \operatorname{Tr}(I - XYX) \overset{\text{(iii)}}{=} \operatorname{Tr}(I) - \operatorname{Tr}(XYX) \overset{\text{(i)}}{=} d - \operatorname{Tr}(X^2Y),$$

for all feasible X and Y, which shows the desired inequality after dividing by d. How can we give such an algebraic proof of positive semidefiniteness? We can algebraically manipulate the polynomial inequalities that define the feasible region. For example, multiplying the inequality $I - Y \succeq 0$ from the left and right by X shows that $X(I - Y)X \succeq 0$. Adding this to the inequality $I - X^2 \succeq 0$ shows that $I - XYX \succeq 0$:

$$I - XYX = I - X^2 + X^2 - XYX = (I - X^2) + X(I - Y)X \succeq 0,$$

and therefore properties (i)–(iv) show that the normalized trace of X^2Y is at most 1. We say that $I - X^2 + X(I - Y)X$ is a *weighted sum of squares* where the weights are polynomials that are nonnegative on the feasible region: in this case $I - X^2$ and $I - Y$. Notice that since we work with matrices, the order of multiplication matters, i.e., we need to view X and Y as noncommutative variables. △

The second proof suggests a more systematic approach. Suppose we want to maximize the normalized trace of a polynomial f over a feasible region defined by polynomial inequalities. Then we can derive upper bounds on that maximum using the following strategy: find the smallest $\lambda \in \mathbb{R}$ for which $\lambda I - f$ can be expressed as a weighted sum of Hermitian squares where the weights are chosen from the polynomials defining the feasible region. Finding the smallest such λ is in general not a tractable problem since the degrees of the polynomials used in the algebraic certificate are unbounded.

To turn this into a tractable approach we need to consider sum-of-squares certificates whose degree is bounded by $t \in \mathbb{N}$; in that case the smallest λ can be found using semidefinite programming. In this chapter we explain how to do so, but we do it from the dual point of view. That is, we don't search for a weighted sum of squares decomposition $\lambda I - f$ directly, instead we consider maps on noncommutative polynomials that behave like the matrix trace. To be more precise, we consider linear functionals on the space of noncommutative polynomials that map weighted sums of squares to nonnegative real numbers, and that satisfy certain other properties of the matrix trace. (In the above example an optimal linear functional would be the map that evaluates a polynomial at $X = Y = 1$.)

How good is this approach? As it turns out, if the problem description is sufficiently nice, then it is quite good. As the degree bound t goes to infinity, the obtained upper bounds converge to an infinite-dimensional analogue of the tracial polynomial optimization problem, and under certain conditions we even recover the finite-dimensional optimum.

4.1 Linear functionals on the space of polynomials

4.1.1 Basic notions

Noncommutative polynomials. We denote the set of all words in the symbols $x_1, \ldots, x_n$ by $\langle \mathbf{x} \rangle = \langle x_1, \ldots, x_n \rangle$, where the empty word is denoted by 1. We do not assume any commutativity here, so for instance $x_1 x_2^2$, $x_2^2 x_1$ and $x_2 x_1 x_2$ are distinct words in $\langle \mathbf{x} \rangle$. This is a semigroup with involution, where the binary operation is concatenation, and the involution of a word $w \in \langle \mathbf{x} \rangle$ is the word w^* obtained by reversing the order of the symbols in w. In particular, this means we assume that $x_i^* = x_i$ holds for all symbols i. The set of all real linear combinations of these words is denoted by $\mathbb{R}\langle \mathbf{x} \rangle$, and its elements are called *noncommutative polynomials*. The involution $w \mapsto w^*$ extends to $\mathbb{R}\langle \mathbf{x} \rangle$ by linearity. In this way $\mathbb{R}\langle \mathbf{x} \rangle$ is a $*$-algebra. A polynomial $p \in \mathbb{R}\langle \mathbf{x} \rangle$ is called *symmetric* if $p^* = p$ and $\mathrm{Sym}\,\mathbb{R}\langle \mathbf{x} \rangle$ denotes the set of symmetric polynomials.[1] The degree (or length) of a word $w \in \langle \mathbf{x} \rangle$ is the number of symbols composing it, denoted as $|w|$ or $\deg(w)$, and the degree of a polynomial $p = \sum_w p_w w \in \mathbb{R}\langle \mathbf{x} \rangle$ is the maximum degree of a word w with $p_w \neq 0$. Given $t \in \mathbb{N} \cup \{\infty\}$, we let $\langle \mathbf{x} \rangle_t$ be the set of words w of degree $|w| \leq t$, so that $\langle \mathbf{x} \rangle_\infty = \langle \mathbf{x} \rangle$, and $\mathbb{R}\langle \mathbf{x} \rangle_t$ is the real vector space of noncommutative polynomials p of

[1] In quantum information theory it is more common to call a function symmetric if its value remains unchanged whenever its input is permuted. The two notions are not the same.

degree $\deg(p) \leq t$. Given $t \in \mathbb{N}$, we let $\langle \mathbf{x} \rangle_{=t}$ be the set of words of degree exactly equal to t.

For a set $S \subseteq \operatorname{Sym} \mathbb{R}\langle \mathbf{x} \rangle$ and $t \in \mathbb{N} \cup \{\infty\}$, the *truncated quadratic module* at degree $2t$ associated to S, denoted $\mathcal{M}_{2t}(S)$, is defined as the cone generated by all polynomials $p^*gp \in \mathbb{R}\langle \mathbf{x} \rangle_{2t}$ with $g \in S \cup \{1\}$:

$$\mathcal{M}_{2t}(S) = \operatorname{cone}\Big\{p^*gp : p \in \mathbb{R}\langle \mathbf{x} \rangle,\ g \in S \cup \{1\},\ \deg(p^*gp) \leq 2t\Big\}. \tag{4.2}$$

Notice that $\mathcal{M}_{2t}(S)$ in particular includes all polynomials of the form p^*p, the so-called *Hermitian squares*.

Likewise, for a set $T \subseteq \mathbb{R}\langle \mathbf{x} \rangle$, we can define the *truncated left ideal* at degree $2t$, denoted by $\mathcal{I}_{2t}(T)$, as the vector space spanned by all polynomials $ph \in \mathbb{R}\langle \mathbf{x} \rangle_{2t}$ with $h \in T$:

$$\mathcal{I}_{2t}(T) = \operatorname{span}\big\{ph : p \in \mathbb{R}\langle \mathbf{x} \rangle,\, h \in T,\, \deg(ph) \leq 2t\big\}. \tag{4.3}$$

We say that the Minkowski sum $\mathcal{M}(S) + \mathcal{I}(T)$ is *Archimedean* when there exists a scalar $R > 0$ such that

$$R - \sum_{i=1}^{n} x_i^2 \in \mathcal{M}(S) + \mathcal{I}(T). \tag{4.4}$$

Linear functionals. Throughout we are interested in the space $\mathbb{R}\langle \mathbf{x} \rangle_t^*$ of real-valued linear functionals on $\mathbb{R}\langle \mathbf{x} \rangle_t$. We list some basic definitions: A linear functional $L \in \mathbb{R}\langle \mathbf{x} \rangle_t^*$ is *symmetric* if $L(w) = L(w^*)$ for all $w \in \langle \mathbf{x} \rangle_t$ and *tracial* if $L(ww') = L(w'w)$ for all $w, w' \in \langle \mathbf{x} \rangle_t$. A linear functional $L \in \mathbb{R}\langle \mathbf{x} \rangle_{2t}^*$ is said to be *positive* if $L(p^*p) \geq 0$ for all $p \in \mathbb{R}\langle \mathbf{x} \rangle_t$. Many properties of a linear functional $L \in \mathbb{R}\langle \mathbf{x} \rangle_{2t}^*$ can be expressed as properties of its associated moment matrix (also known as its *Hankel matrix*). For $L \in \mathbb{R}\langle \mathbf{x} \rangle_{2t}^*$ we define its associated *moment matrix*, which has rows and columns indexed by words in $\langle \mathbf{x} \rangle_t$, by

$$M_t(L)_{w,w'} = L(w^*w') \quad \text{for} \quad w, w' \in \langle \mathbf{x} \rangle_t,$$

and as usual we set $M(L) = M_\infty(L)$. Notice that $|\langle \mathbf{x} \rangle_t| = \sum_{k=0}^t n^k = \frac{n^{t+1}-1}{n-1}$ and therefore the moment matrix is an $\frac{n^{t+1}-1}{n-1} \times \frac{n^{t+1}-1}{n-1}$ matrix. Properties of L correspond to properties of $M_t(L)$. In particular, L is symmetric if and only if $M_t(L)$ is symmetric, and L is positive if and only if $M_t(L)$ is positive semidefinite. In fact, one can even express nonnegativity of a linear form $L \in \mathbb{R}\langle \mathbf{x} \rangle_{2t}^*$ on $\mathcal{M}_{2t}(S)$ in terms of certain associated positive semidefinite moment matrices. For this, given a polynomial $g \in \mathbb{R}\langle \mathbf{x} \rangle$, define the linear form $gL \in \mathbb{R}\langle \mathbf{x} \rangle_{2t-\deg(g)}^*$ by $(gL)(p) = L(gp)$. Then we have

$$L(p^*gp) \geq 0 \text{ for all } p \in \mathbb{R}\langle \mathbf{x} \rangle_{t-d_g} \iff M_{t-d_g}(gL) \succeq 0, \quad (d_g = \lceil \deg(g)/2 \rceil),$$

and thus $L \geq 0$ on $\mathcal{M}_{2t}(S)$ if and only if $M_{t-d_g}(gL) \succeq 0$ for all $g \in S \cup \{1\}$. Similarly, we can express the condition $L = 0$ on $\mathcal{I}_{2t}(T)$ by enforcing linear equalities on the entries of $M_t(L)$.

The moment matrix also allows us to define a property called *flatness.* For $t, \delta \in \mathbb{N}$, $\delta \leq t$, a linear functional $L \in \mathbb{R}\langle \mathbf{x} \rangle_{2t}^*$ is called *δ-flat* if the rank of $M_t(L)$ is equal to that of its principal submatrix indexed by the words in $\langle \mathbf{x} \rangle_{t-\delta}$, that is,

$$\operatorname{rank}(M_t(L)) = \operatorname{rank}(M_{t-\delta}(L)). \tag{4.5}$$

We call L *flat* if it is δ-flat for some $\delta \geq 1$. When $t = \infty$, L is said to be *flat* when $\operatorname{rank}(M(L)) < \infty$, which is equivalent to $\operatorname{rank}(M(L)) = \operatorname{rank}(M_s(L))$ for some $s \in \mathbb{N}$.

A key example of a linear functional on $\mathbb{R}\langle \mathbf{x} \rangle$ is given by the *trace evaluation* at a given matrix tuple $\mathbf{X} = (X_1, \ldots, X_n) \in (\mathrm{H}^d)^n$:

$$p \in \mathbb{R}\langle \mathbf{x} \rangle \mapsto \operatorname{Tr}(p(\mathbf{X})).$$

Here $p(\mathbf{X})$ denotes the matrix obtained by substituting x_i by X_i in p, and throughout $\operatorname{Tr}(\cdot)$ denotes the usual matrix trace, which satisfies $\operatorname{Tr}(I) = d$ where I is the identity matrix in H^d. We mention in passing that we use $\operatorname{tr}(\cdot)$ to denote the *normalized matrix trace*, which satisfies $\operatorname{tr}(I) = 1$ for $I \in \mathrm{H}^d$. The trace evaluation provides a linear functional that is symmetric ($\operatorname{Tr}(p(\mathbf{X})) = \operatorname{Tr}(p^*(\mathbf{X}))$), tracial ($\operatorname{Tr}(p(\mathbf{X})q(\mathbf{X})) = \operatorname{Tr}(q(\mathbf{X})p(\mathbf{X}))$), and positive ($\operatorname{Tr}(p^*(\mathbf{X})p(\mathbf{X})) \geq 0$). Moreover, the trace evaluation functional is flat, since the matrix algebra $\mathbb{C}^{d\times d}$ is finite-dimensional. Throughout, we use $L_{\mathbf{X}}$ to denote the real part of the above functional, that is, $L_{\mathbf{X}}$ denotes the linear form on $\mathbb{R}\langle \mathbf{x} \rangle$ defined by

$$L_{\mathbf{X}}(p) = \operatorname{Re}(\operatorname{Tr}(p(X_1, \ldots, X_n))) \quad \text{for} \quad p \in \mathbb{R}\langle \mathbf{x} \rangle. \tag{4.6}$$

Observe that $L_{\mathbf{X}}$ too is a symmetric tracial positive linear functional on $\mathbb{R}\langle \mathbf{x} \rangle$. Moreover, $L_{\mathbf{X}}$ is nonnegative on $\mathcal{M}(S)$ if the matrix tuple $\mathbf{X}$ is taken from the *matrix positivity domain* $\mathcal{D}(S)$ associated to the finite set $S \subseteq \operatorname{Sym} \mathbb{R}\langle \mathbf{x} \rangle$, defined as

$$\mathcal{D}(S) = \bigcup_{d\geq 1} \Big\{ \mathbf{X} = (X_1, \ldots, X_n) \in (\mathrm{H}^d)^n : g(\mathbf{X}) \succeq 0 \text{ for all } g \in S \Big\}. \tag{4.7}$$

Similarly, the linear functional $L_{\mathbf{X}}$ is zero on $\mathcal{I}(T)$ if the matrix tuple $\mathbf{X}$ is taken from the *matrix variety* $\mathcal{V}(T)$ associated to the finite set $T \subseteq \mathbb{R}\langle \mathbf{x} \rangle$, defined as

$$\mathcal{V}(T) = \bigcup_{d\geq 1} \big\{ \mathbf{X} \in (\mathrm{H}^d)^n : h(\mathbf{X}) = 0 \text{ for all } h \in T \big\}, \tag{4.8}$$

4.1.2 C^*-algebras

In the next section we will discuss (flat) linear functionals extensively. In order to link them to polynomial optimization problems we need to discuss an infinite-dimensional generalization of matrix algebras, namely C^*-algebras admitting a tracial state. Below we give a brief introduction to the terminology, and we discuss two useful results of Artin and Wedderburn. We refer to, e.g., the book [Bla06] for (far) more information about C^*-algebras. In Remark 4.3 below we explain a connection between C^*-algebras and quantum states.

A C^**-algebra* $\mathcal{A}$ can be defined as a norm-closed $*$-subalgebra of the space $\mathcal{B}(\mathcal{H})$ of bounded operators on a complex Hilbert space $\mathcal{H}$.[2] Here, the involution * on $\mathcal{B}(\mathcal{H})$ is the usual adjoint operation, and a $*$-subalgebra is a subalgebra that is closed under taking adjoints. In particular, we have $\|a^*a\| = \|a\|^2$ for all elements a in the algebra, where $\|\cdot\|$ is the operator norm in $\mathcal{B}(\mathcal{H})$. Such an algebra $\mathcal{A}$ is said to be *unital* if it contains the identity operator (denoted 1). When $\mathcal{H}$ has finite dimension d this means $\mathcal{A}$ is a *matrix* $*$*-algebra*, i.e., $\mathcal{A}$ is a subalgebra of $\mathbb{C}^{d\times d}$ that is closed under taking complex conjugates. Examples of matrix $*$-algebras include the full matrix algebra $\mathbb{C}^{d\times d}$ or the $*$-algebra generated by given matrices $X_1,\dots,X_n \in \mathbb{C}^{d\times d}$, denoted $\mathbb{C}\langle X_1,\dots,X_n\rangle$. An algebra is called *finite-dimensional* if it is finite-dimensional as a vector space. The following results due to Artin and Wedderburn (see [Wed64, BEK78]) will be useful in the next section: Any finite-dimensional C^*-algebra is ($*$-isomorphic to) a matrix $*$-algebra containing the identity, and in turn any such matrix $*$-algebra is isomorphic to a direct sum of full matrix algebras. We record the latter result for future reference:

Theorem 4.2 ([Wed64, BEK78])**.** *Let $\mathcal{A}$ be a complex matrix $*$-subalgebra of $\mathbb{C}^{d\times d}$ containing the identity. Then there exists a unitary matrix U and integers K, m_k, n_k for $k \in [K]$ such that*

$$U\mathcal{A}U^* = \bigoplus_{k=1}^{K} (\mathbb{C}^{n_k\times n_k} \otimes I_{m_k}) \quad \textit{and} \quad d = \sum_{k=1}^{K} m_k n_k.$$

An element b in a C^*-algebra $\mathcal{A}$ is called *positive*, denoted $b \succeq 0$, if it is of the form $b = a^*a$ for some $a \in \mathcal{A}$. For finite sets $S \subseteq \operatorname{Sym}\mathbb{R}\langle \mathbf{x}\rangle$ and $T \subseteq \mathbb{R}\langle \mathbf{x}\rangle$, the C^*-algebraic analogues of the matrix positivity domain and matrix variety are the sets

$$\mathcal{D}_{\mathcal{A}}(S) = \big\{\mathbf{X} = (X_1,\dots,X_n) \in \mathcal{A}^n : X_i^* = X_i \text{ for } i \in [n],\ g(\mathbf{X}) \succeq 0 \text{ for all } g \in S\big\},$$
$$\mathcal{V}_{\mathcal{A}}(T) = \big\{\mathbf{X} = (X_1,\dots,X_n) \in \mathcal{A}^n : X_i^* = X_i \text{ for } i \in [n],\ h(\mathbf{X}) = 0 \text{ for all } h \in T\big\}.$$

A *state* τ on a unital C^*-algebra $\mathcal{A}$ is a linear form $\tau : \mathcal{A} \to \mathbb{C}$ on $\mathcal{A}$ that is *positive*, i.e., $\tau(a^*a) \geq 0$ for all $a \in \mathcal{A}$, and satisfies $\tau(1) = 1$. Since $\mathcal{A}$ is a complex algebra, every state τ is Hermitian: $\tau(a) = \overline{\tau(a^*)}$ for all $a \in \mathcal{A}$. We say that that a state is *tracial* if $\tau(ab) = \tau(ba)$ for all $a, b \in \mathcal{A}$ and *faithful* if $\tau(a^*a) = 0$ implies $a = 0$. A useful fact is that on a full matrix algebra $\mathbb{C}^{d\times d}$ the normalized matrix trace is the unique tracial state (see, e.g., [BK12]). Now, given a tuple $\mathbf{X} = (X_1,\dots,X_n) \in \mathcal{A}^n$ in a C^*-algebra $\mathcal{A}$ with tracial state τ, the second key example of a symmetric tracial positive linear functional on $\mathbb{R}\langle \mathbf{x}\rangle$ is given by the *trace evaluation map*, which we again denote by $L_{\mathbf{X}}$ and is defined by

$$L_{\mathbf{X}}(p) = \tau(p(X_1,\dots,X_n)) \quad \text{for all} \quad p \in \mathbb{R}\langle \mathbf{x}\rangle.$$

[2] There is another, more abstract, definition of a C^*-algebra. The Gelfand-Naimark theorem states that the two definitions are equivalent (see, e.g., [Bla06, II.6.4.10]). We chose to work with the one that is closest to our applications.

Remark 4.3. *We have introduced two types of states: quantum states and states on a unital C^*-algebra. The similarity in terminology is not a coincidence, as we will now explain. We defined a quantum state as a unit vector ψ on a (separable) Hilbert space $\mathcal{H}$. As we have mentioned In Section 3.1, to a quantum state we can associate a linear functional $\tau : \mathcal{B}(\mathcal{H}) \to \mathbb{C}$ on the C^*-algebra $\mathcal{B}(\mathcal{H})$ by setting $\tau(A) = \langle \psi, A\psi \rangle$. One can easily verify that τ is a state on $\mathcal{B}(\mathcal{H})$. As it turns out, to each state on a C^*-algebra we can also associate a quantum state. The theorem below is a formulation that can be found, e.g., in the book [KR97, Thm. 4.5.2].*

Theorem 4.4. *If τ is a state on a unital C^*-algebra $\mathcal{A}$, then there exists a cyclic representation π of $\mathcal{A}$ on a Hilbert space $\mathcal{H}$, and a unit cyclic vector v such that $\tau(A) = \langle v, \pi(A)v \rangle$ for all $A \in \mathcal{A}$.*[3]

The proof of this statement relies on the Gelfand-Naimark-Segal construction that we will see in the next section.

In fact, the study of C^*-algebras was (initially) motivated by its connection to quantum mechanics. In particular, through Heisenberg's "matrix mechanics" and the consecutive work of von Neumann. See for instance the introduction of the book [BR87] for a discussion on this connection.

4.1.3 Flat extensions and representations of linear forms

In the previous section we have seen that the key examples of symmetric tracial linear functionals on $\mathbb{R}\langle \mathbf{x} \rangle_{2t}$ are trace evaluations at elements of a (finite-dimensional) C^*-algebra. In this section we present some results that provide conditions under which, conversely, a symmetric tracial linear functional on $\mathbb{R}\langle \mathbf{x} \rangle_{2t}$ ($t \in \mathbb{N} \cup \{\infty\}$) that is nonnegative on $\mathcal{M}(S)$ and zero on $\mathcal{I}(T)$ arises from trace evaluations at elements in the intersection of the C^*-algebraic analogues of the matrix positivity domain of S ($\mathcal{D}_\mathcal{A}(S)$) and the matrix variety of T ($\mathcal{V}_\mathcal{A}(T)$). In Theorem 4.5 and Theorem 4.6 below we first consider the case $t = \infty$ and then in Theorem 4.7 we consider the finite case: $t \in \mathbb{N}$.

The proofs of Theorem 4.5 and Theorem 4.6 use a classical Gelfand–Naimark–Segal (GNS) construction. In these proofs it is convenient to work with the concept of the null space of a linear functional $L \in \mathbb{R}\langle \mathbf{x} \rangle_{2t}^*$, which is defined as the vector space

$$N_t(L) = \big\{ p \in \mathbb{R}\langle \mathbf{x} \rangle_t : L(qp) = 0 \text{ for all } q \in \mathbb{R}\langle \mathbf{x} \rangle_t \big\}.$$

We use the notation $N(L) = N_\infty(L)$ for the nontruncated null space. Recall that $M_t(L)$ is the moment matrix associated to L, its rows and columns are indexed by words in $\langle \mathbf{x} \rangle_t$, and its entries are given by $M_t(L)_{w,w'} = L(w^* w')$ for words $w, w' \in \langle \mathbf{x} \rangle_t$. The null space of L can therefore be identified with the kernel of $M_t(L)$: A polynomial $p = \sum_{w \in \langle \mathbf{x} \rangle_t} c_w w$ belongs to $N_t(L)$ if and only if its coefficient vector (c_w) belongs to the kernel of $M_t(L)$.

In Section 4.1.1 we defined a linear functional $L \in \mathbb{R}\langle \mathbf{x} \rangle_{2t}^*$ to be δ-flat based on the rank stabilization property (4.5) of its moment matrix: $\mathrm{rank}(M_t(L)) =$

[3] A representation $\pi : \mathcal{A} \to \mathcal{B}(\mathcal{H})$ is cyclic if there exists a unit vector v such that the set $\{\pi(A)v : A \in \mathcal{A}\}$ is dense in $\mathcal{H}$. Such a vector v is called a unit cyclic vector.

$\mathrm{rank}(M_{t-\delta}(L))$. This definition can be reformulated in terms of a decomposition of the corresponding polynomial space using the null space: the form L is δ-flat if and only if

$$\mathbb{R}\langle\mathbf{x}\rangle_t = \mathbb{R}\langle\mathbf{x}\rangle_{t-\delta} + N_t(L).$$

Recall that L is said to be flat if it is δ-flat for some $\delta \geq 1$. Finally, in the nontruncated case ($t = \infty$) L was called flat if $\mathrm{rank}(M(L)) < \infty$. We can now see that $\mathrm{rank}(M(L)) < \infty$ if and only if there exists an integer $s \in \mathbb{N}$ such that $\mathbb{R}\langle\mathbf{x}\rangle = \mathbb{R}\langle\mathbf{x}\rangle_s + N(L)$.

Theorem 4.5 below is implicit in several works (see, e.g., [NPA12, BKP16]). Here we assume that the Minkowski sum $\mathcal{M}(S) + \mathcal{I}(T)$ is Archimedean, which we recall means that there exists a scalar $R > 0$ such that

$$R - \sum_{i=1}^{n} x_i^2 \in \mathcal{M}(S) + \mathcal{I}(T). \tag{4.4}$$

Archimedeanity is only required to prove the implication (1) ⇒ (2).

Theorem 4.5. *Let $S \subseteq \mathrm{Sym}\,\mathbb{R}\langle\mathbf{x}\rangle$ and $T \subseteq \mathbb{R}\langle\mathbf{x}\rangle$ with $\mathcal{M}(S) + \mathcal{I}(T)$ Archimedean. Given a linear form $L \in \mathbb{R}\langle\mathbf{x}\rangle^*$, the following are equivalent:*

(1) L is symmetric, tracial, nonnegative on $\mathcal{M}(S)$, zero on $\mathcal{I}(T)$, and $L(1) = 1$;

(2) there is a unital C^*-algebra $\mathcal{A}$ with tracial state τ and $\mathbf{X} \in \mathcal{D}_{\mathcal{A}}(S) \cap \mathcal{V}_{\mathcal{A}}(T)$ with

$$L(p) = \tau(p(\mathbf{X})) \quad \textit{for all} \quad p \in \mathbb{R}\langle\mathbf{x}\rangle. \tag{4.9}$$

Proof. We first prove the easy direction (2) ⇒ (1): We have

$$L(p^*) = \tau(p^*(\mathbf{X})) = \tau(p(\mathbf{X})^*) = \overline{\tau(p(\mathbf{X}))} = \overline{L(p)} = L(p),$$

where we use that τ is Hermitian and $X_i^* = X_i$ for $i \in [n]$. Moreover, L is tracial since τ is tracial. In addition, for $g \in S \cup \{1\}$ and $p \in \mathbb{R}\langle\mathbf{x}\rangle$ we have

$$L(p^*gp) = \tau(p^*(\mathbf{X})g(\mathbf{X})p(\mathbf{X})) = \tau(p(\mathbf{X})^*g(\mathbf{X})p(\mathbf{X})) \geq 0,$$

since $g(\mathbf{X})$ is positive in $\mathcal{A}$ as $\mathbf{X} \in \mathcal{D}_{\mathcal{A}}(S)$ and τ is positive. Similarly $L(hq) = \tau(h(\mathbf{X})q(\mathbf{X})) = 0$ for all $h \in T$, since $\mathbf{X} \in \mathcal{V}_{\mathcal{A}}(\mathbf{T})$.

We show (1) ⇒ (2) by applying a GNS construction. Consider the quotient vector space $\mathbb{R}\langle\mathbf{x}\rangle/N(L)$, and denote the equivalence class of p in $\mathbb{R}\langle\mathbf{x}\rangle/N(L)$ by $\overline{p}$. We can equip this quotient with the inner product $\langle\overline{p},\overline{q}\rangle = L(p^*q)$ for $p, q \in \mathbb{R}\langle\mathbf{x}\rangle$, so that the completion $\mathcal{H}$ of $\mathbb{R}\langle\mathbf{x}\rangle/N(L)$ is a separable Hilbert space. As $N(L)$ is a left ideal in $\mathbb{R}\langle\mathbf{x}\rangle$, the operator

$$X_i \colon \mathbb{R}\langle\mathbf{x}\rangle/N(L) \to \mathbb{R}\langle\mathbf{x}\rangle/N(L),\, \overline{p} \mapsto \overline{x_i p} \tag{4.10}$$

is well defined. We have

$$\langle X_i\,\overline{p},\overline{q}\rangle = L((x_ip)^*q) = L(p^*x_iq) = \langle\overline{p}, X_i\overline{q}\rangle \quad \text{for all} \quad p, q \in \mathbb{R}\langle\mathbf{x}\rangle,$$

so the X_i are self-adjoint. Since $g \in S \cup \{1\}$ is assumed to be symmetric and $\langle \overline{p}, g(\mathbf{X})\overline{p} \rangle = \langle \overline{p}, \overline{gp} \rangle = L(p^* gp) \geq 0$ for all p we have $g(\mathbf{X}) \succeq 0$. By the Archimedean condition (4.4), there exists an $R > 0$ such that $R - \sum_{i=1}^n x_i^2 \in \mathcal{M}(S) + \mathcal{I}(T)$. Using $R - x_i^2 = (R - \sum_{j=1}^n x_j^2) + \sum_{j \neq i} x_j^2 \in \mathcal{M}(S) + \mathcal{I}(T)$ we get

$$\langle X_i \overline{p}, X_i \overline{p} \rangle = L(p^* x_i^2 p) \leq R \cdot L(p^* p) = R \langle \overline{p}, \overline{p} \rangle \quad \text{for all} \quad p \in \mathbb{R}\langle \mathbf{x} \rangle.$$

So each X_i extends to a bounded self-adjoint operator, also denoted X_i, on the Hilbert space $\mathcal{H}$ such that $g(\mathbf{X})$ is positive for all $g \in S \cup \{1\}$. Moreover, we have $\langle \overline{f}, h(\mathbf{X})\overline{1} \rangle = L(f^* h) = 0$ for all $f \in \mathbb{R}\langle \mathbf{x} \rangle, h \in T$.

The operators $X_i \in \mathcal{B}(\mathcal{H})$ extend to self-adjoint operators in $\mathcal{B}(\mathbb{C} \otimes_{\mathbb{R}} \mathcal{H})$, where $\mathbb{C} \otimes_{\mathbb{R}} \mathcal{H}$ is the complexification of $\mathcal{H}$. Let $\mathcal{A}$ be the unital C^*-algebra obtained by taking the operator norm closure of $\mathbb{R}\langle \mathbf{X} \rangle \subseteq \mathcal{B}(\mathbb{C} \otimes_{\mathbb{R}} \mathcal{H})$. It follows that $\mathbf{X} \in \mathcal{D}_{\mathcal{A}}(S) \cap \mathcal{V}_{\mathcal{A}}(T)$.

Define the state τ on $\mathcal{A}$ by $\tau(a) = \langle \overline{1}, a\overline{1} \rangle$ for $a \in \mathcal{A}$. For all $p, q \in \mathbb{R}\langle \mathbf{x} \rangle$ we have

$$\tau(p(\mathbf{X})q(\mathbf{X})) = \langle \overline{1}, p(\mathbf{X})q(\mathbf{X})\overline{1} \rangle = \langle \overline{1}, \overline{pq} \rangle = L(pq), \tag{4.11}$$

so that the restriction of τ to $\mathbb{R}\langle \mathbf{X} \rangle$ is tracial. Since $\mathbb{R}\langle \mathbf{X} \rangle$ is dense in $\mathcal{A}$ in the operator norm, this implies τ is tracial.

To conclude the proof, observe that Equation (4.9) follows from Equation (4.11) by taking $q = 1$. □

The next result can be seen as a finite-dimensional analogue of the above result, where we do not need $\mathcal{M}(S) + \mathcal{I}(T)$ to be Archimedean, but instead we assume the rank of $M(L)$ to be finite (i.e., L to be flat). In addition to the Gelfand–Naimark–Segal construction, the proof uses Artin–Wedderburn theory. For the unconstrained case the proof of this result can be found in [BK12], and in [BKP16, KP16] this result is extended to the constrained case. Recall that $\mathcal{D}(S), \mathcal{V}(T)$, and their C^*-algebraic analogues have been defined in Equations (4.7), (4.8), and Section 4.1.2 respectively.

Theorem 4.6. *For $S \subseteq \operatorname{Sym} \mathbb{R}\langle \mathbf{x} \rangle$, $T \subseteq \mathbb{R}\langle \mathbf{x} \rangle$, and $L \in \mathbb{R}\langle \mathbf{x} \rangle^*$, the following are equivalent:*

(1) L is a symmetric, tracial, linear form with $L(1) = 1$ that is nonnegative on $\mathcal{M}(S)$, zero on $\mathcal{I}(T)$, and has $\operatorname{rank}(M(L)) < \infty$;

(2) there is a finite-dimensional C^*-algebra $\mathcal{A}$ with a tracial state τ, and a tuple $\mathbf{X} \in \mathcal{D}_{\mathcal{A}}(S) \cap \mathcal{V}_{\mathcal{A}}(T)$ satisfying Equation (4.9);

(3) L is a convex combination of normalized trace evaluations at points in the set $\mathcal{D}(S) \cap \mathcal{V}(T)$.

Proof. ((1) ⇒ (2)) Here we can follow the proof of Theorem 4.5, with the extra observation that the condition $\operatorname{rank}(M(L)) < \infty$ implies that the quotient space $\mathbb{R}\langle \mathbf{x} \rangle / N(L)$ is finite-dimensional. Since $\mathbb{R}\langle \mathbf{x} \rangle / N(L)$ is finite-dimensional, the multiplication operators are bounded, and the constructed C^*-algebra $\mathcal{A}$ is finite-dimensional.

((2) ⇒ (3)) By Artin-Wedderburn theory there exists a $*$-isomorphism

$$\varphi\colon \mathcal{A} \to \bigoplus_{m=1}^{M} \mathbb{C}^{d_m \times d_m} \quad \text{for some } \; M \in \mathbb{N},\ d_1, \dots, d_M \in \mathbb{N}.$$

Define the $*$-homomorphisms $\varphi_m\colon \mathcal{A} \to \mathbb{C}^{d_m \times d_m}$ for $m \in [M]$ by $\varphi = \oplus_{m=1}^{M} \varphi_m$. Then, for each $m \in [M]$, the map $\mathbb{C}^{d_m \times d_m} \to \mathbb{C}$ defined by $X \mapsto \tau(\varphi_m^{-1}(X))$ is a positive tracial linear form, and hence it is a nonnegative multiple $\lambda_m \mathrm{tr}(\cdot)$ of the normalized matrix trace (since, for a full matrix algebra, the normalized trace is the unique tracial state). Then we have $\tau(a) = \sum_m \lambda_m \,\mathrm{tr}(\varphi_m(a))$ for all $a \in \mathcal{A}$. So $\tau(\cdot) = \sum_m \lambda_m \mathrm{tr}(\cdot)$ for nonnegative scalars λ_m with $\sum_m \lambda_m = L(1) = 1$. By defining the matrices $X_i^m = \varphi_m(X_i)$ for $m \in [M]$, we get

$$L(p) = \tau(p(X_1, \dots, X_n)) = \sum_{m=1}^{M} \lambda_m \,\mathrm{tr}(p(X_1^m, \dots, X_n^m)) \quad \text{for all} \quad p \in \mathbb{R}\langle \mathbf{x} \rangle.$$

Since φ_m is a $*$-homomorphism we have $g(X_1^m, \dots, X_n^m) \succeq 0$ for all $g \in S \cup \{1\}$ and also $h(X_1^m, \dots, X_n^m) = 0$ for all $h \in T$, which shows $(X_1^m, \dots, X_n^m) \in \mathcal{D}(S) \cap \mathcal{V}(T)$.

((3) ⇒ (1)) If L is a convex combination of normalized trace evaluations at elements from $\mathcal{D}(S) \cap \mathcal{V}(T)$, then L is symmetric, tracial, nonnegative on $\mathcal{M}(S)$, zero on $\mathcal{I}(T)$, and satisfies $\mathrm{rank}(M(L)) < \infty$ because the moment matrix of any trace evaluation has finite rank. Moreover $L(1) = 1$. □

The previous two theorems were about linear functionals defined on the full space of noncommutative polynomials. The following result claims that a *flat* linear functional on a truncated polynomial space can be extended to a flat linear functional on the full space of polynomials while preserving the same positivity properties. It is due to Curto and Fialkow [CF96] in the commutative case and extensions to the noncommutative case can be found in [PNA10] (for eigenvalue optimization) and [BK12] (for trace optimization).

Theorem 4.7. *Let $1 \leq \delta \leq t < \infty$, $S \subseteq \mathrm{Sym}\,\mathbb{R}\langle \mathbf{x} \rangle_{2\delta}$, and $T \subseteq \mathbb{R}\langle \mathbf{x} \rangle_{2\delta}$. Suppose $L \in \mathbb{R}\langle \mathbf{x} \rangle_{2t}^*$ is symmetric, tracial, δ-flat, nonnegative on $\mathcal{M}_{2t}(S)$, and zero on $\mathcal{I}_{2t}(T)$. Then L extends to a symmetric, tracial, linear form on $\mathbb{R}\langle \mathbf{x} \rangle$ that is nonnegative on $\mathcal{M}(S)$, zero on $\mathcal{I}(T)$, and whose moment matrix has finite rank.*

Proof. Let $W \subseteq \langle \mathbf{x} \rangle_{t-\delta}$ index a maximum nonsingular submatrix of $M_{t-\delta}(L)$, and let $\mathrm{span}(W)$ be the linear space spanned by W. We have the vector space direct sum

$$\mathbb{R}\langle \mathbf{x} \rangle_t = \mathrm{span}(W) \oplus N_t(L). \tag{4.12}$$

That is, for each $u \in \langle \mathbf{x} \rangle_t$ there exists a unique $r_u \in \mathrm{span}(W)$ such that $u - r_u \in N_t(L)$.

We first construct the (unique) symmetric flat extension $\hat{L} \in \mathbb{R}\langle \mathbf{x} \rangle_{2t+2}$ of L. For this we set $\hat{L}(p) = L(p)$ for $\deg(p) \leq 2t$, and we set

$$\hat{L}(u^* x_i v) = L(u^* x_i r_v) \quad \text{and} \quad \hat{L}((x_i u)^* x_j v) = L((x_i r_u)^* x_j r_v)$$

for all $i, j \in [n]$ and $u, v \in \langle \mathbf{x} \rangle$ with $|u| = |v| = t$. One can verify that $\hat{L}$ is symmetric and satisfies $x_i(u - r_u) \in N_{t+1}(\hat{L})$ for all $i \in [n]$ and $u \in \mathbb{R}\langle \mathbf{x} \rangle_t$, from which it follows that $\hat{L}$ is 2-flat.

We also have $(u - r_u)x_i \in N_{t+1}(\hat{L})$ for all $i \in [n]$ and $u \in \mathbb{R}\langle \mathbf{x} \rangle_t$: Since $\hat{L}$ is 2-flat, we have $(u - r_u)x_i \in N_{t+1}(\hat{L})$ if and only if $\hat{L}(p(u - r_u)x_i) = 0$ for all $p \in \mathbb{R}\langle \mathbf{x} \rangle_{t-1}$. By using $\deg(x_i p) \leq t$, L is tracial, and $u - r_u \in N_t(L)$, we get $\hat{L}(p(u - r_u)x_i) = L(p(u - r_u)x_i) = L(x_i p(u - r_u)) = 0$.

By consecutively using $(v - r_v)x_j \in N_{t+1}(\hat{L})$, symmetry of $\hat{L}$, $x_i(u - r_u) \in N_{t+1}(\hat{L})$, and again symmetry of $\hat{L}$, we see that

$$\begin{aligned} \hat{L}((x_i u)^* v x_j) &= \hat{L}((x_i u)^* r_v x_j) = \hat{L}((r_v x_j)^* x_i u) \\ &= \hat{L}((r_v x_j)^* x_i r_u) = \hat{L}((x_i r_u)^* r_v x_j), \end{aligned} \tag{4.13}$$

and in an analogous way one can show

$$\hat{L}((u x_i)^* x_j v) = \hat{L}((r_u x_i)^* x_j r_v). \tag{4.14}$$

We can now show that $\hat{L}$ is tracial. We do this by showing that $\hat{L}(w x_j) = \hat{L}(x_j w)$ for all w with $\deg(w) \leq 2t + 1$. Notice that when $\deg(w) \leq 2t - 1$ the statement follows from the fact that $\hat{L}$ is an extension of L. Suppose $w = u^* v$ with $\deg(u) = t + 1$ and $\deg(v) \leq t$. We write $u = x_i u'$, and we let $r_{u'}, r_v \in \mathbb{R}\langle \mathbf{x} \rangle_{t-1}$ be such that $u' - r_{u'}, v - r_v \in N_t(L)$. We then have

$$\begin{aligned} \hat{L}(w x_j) = \hat{L}(u^* v x_j) &= \hat{L}((x_i u')^* v x_j) \\ &= \hat{L}((x_i r_{u'})^* r_v x_j) && \text{by } (4.13) \\ &= L((x_i r_{u'})^* r_v x_j) && \text{since } \deg(x_i r_{u'} r_v x_j) \leq 2t \\ &= L((r_{u'} x_j)^* x_i r_v) && \text{since } L \text{ is tracial} \\ &= \hat{L}((r_{u'} x_j)^* x_i r_v) && \text{since } \deg((r_{u'} x_j)^* x_i r_v) \leq 2t \\ &= \hat{L}((u' x_j)^* x_i v) && \text{by } (4.14) \\ &= \hat{L}(x_j w). \end{aligned}$$

It follows that $\hat{L}$ is a symmetric tracial flat extension of L, and $\operatorname{rank}(M(\hat{L})) = \operatorname{rank}(M(L))$.

Next, we iterate the above procedure to extend L to a symmetric tracial linear functional $\hat{L} \in \mathbb{R}\langle \mathbf{x} \rangle^*$. It remains to show that $\hat{L}$ is nonnegative on $\mathcal{M}(S)$ and zero on $\mathcal{I}(T)$. For this we make two observations:

(i) $\mathcal{I}(N_t(L)) \subseteq N(\hat{L})$.

(ii) $\mathbb{R}\langle \mathbf{x} \rangle = \operatorname{span}(W) \oplus \mathcal{I}(N_t(L))$.

For (i) we use the (easy to check) fact that $N_t(L) = \operatorname{span}(\{u - r_u : u \in \langle \mathbf{x} \rangle_t\})$. Then it suffices to show that $w(u - r_u) \in N(\hat{L})$ for all $w \in \langle \mathbf{x} \rangle$, which can be done using induction on $|w|$. From (i) one easily deduces that $\operatorname{span}(W) \cap N(\hat{L}) = \{0\}$, so we have the direct sum $\operatorname{span}(W) \oplus \mathcal{I}(N_t(L))$. The claim (ii) follows using induction on the length of $w \in \langle \mathbf{x} \rangle$: The base case $w \in \langle \mathbf{x} \rangle_t$ follows from (4.12).

Let $w = x_i v \in \langle \mathbf{x} \rangle$ and assume $v \in \operatorname{span}(W) \oplus \mathcal{I}(N_t(L))$, that is, $v = r_v + q_v$ where $r_v \in \operatorname{span}(W)$ and $q_v \in \mathcal{I}(N_t(L))$. We have $x_i v = x_i r_v + x_i q_v$ so it suffices to show $x_i r_v, x_i q_v \in \operatorname{span}(W) \oplus \mathcal{I}(N_t(L))$. Clearly $x_i q_v \in \mathcal{I}(N_t(L))$, since $q_v \in \mathcal{I}(N_t(L))$. Also, observe that $x_i r_v \in \mathbb{R}\langle \mathbf{x} \rangle_t$ and therefore $x_i r_v \in \operatorname{span}(W) \oplus \mathcal{I}(N_t(L))$ by (4.12).

We conclude the proof by showing that $\hat{L}$ is nonnegative on $\mathcal{M}(S)$ and zero on $\mathcal{I}(T)$. Let $g \in \mathcal{M}(S)$, $h \in \mathcal{I}(T)$, and $p \in \mathbb{R}\langle \mathbf{x} \rangle$. For $p \in \mathbb{R}\langle \mathbf{x} \rangle$ we extend the definition of r_p so that $r_p \in \operatorname{span}(W)$ and $p - r_p \in \mathcal{I}(N_t(L))$, which is possible by (ii). Then,

$$\hat{L}(p^* g p) \overset{\text{(i)}}{=} \hat{L}(p^* g r_p) = \hat{L}(r_p^* g p) \overset{\text{(i)}}{=} \hat{L}(r_p^* g r_p) = L(r_p^* g r_p) \geq 0,$$

$$\hat{L}(p^* h) = \hat{L}(h^* p) \overset{\text{(i)}}{=} \hat{L}(h^* r_p) = \hat{L}(r_p h) = L(r_p h) = 0,$$

where we use $\deg(r_p^* g r_p) \leq 2(t - \delta) + 2\delta = 2t$ and $\deg(r_p h) \leq (t - \delta) + 2\delta \leq 2t$. □

Combining Theorem 4.6 and Theorem 4.7 gives the following result, which shows that a flat linear form can be extended to a conic combination of trace evaluation maps. It was first proven in [KP16, Proposition 6.1] (and in [BK12] for the unconstrained case).

Corollary 4.8. *Let $1 \leq \delta \leq t < \infty$, $S \subseteq \operatorname{Sym} \mathbb{R}\langle \mathbf{x} \rangle_{2\delta}$, and $T \subseteq \mathbb{R}\langle \mathbf{x} \rangle_{2\delta}$. If $L \in \mathbb{R}\langle \mathbf{x} \rangle_{2t}^*$ is symmetric, tracial, δ-flat, nonnegative on $\mathcal{M}_{2t}(S)$, and zero on $\mathcal{I}_{2t}(T)$, then it extends to a conic combination of trace evaluations at elements of $\mathcal{D}(S) \cap \mathcal{V}(T)$.*

4.1.4 Specialization to the commutative setting

The material from Section 4.1.3 can be adapted to the commutative setting. In the commutative setting the variables (symbols) $x_1, \ldots, x_n$ are assumed to pairwise commute. Throughout $[\mathbf{x}]$ denotes the set of monomials in $x_1, \ldots, x_n$, i.e., the commutative analogue of $\langle \mathbf{x} \rangle$. Observe that there are far fewer commutative monomials than noncommutative monomials:

$$|[\mathbf{x}]_t| = \binom{n}{\leq t} = \sum_{k=0}^{t} \binom{n}{k} \leq \frac{n^{t+1} - 1}{n - 1} = |\langle \mathbf{x} \rangle_t|.$$

The moment matrix $M_t(L)$ of a linear form $L \in \mathbb{R}[\mathbf{x}]_{2t}^*$ is now indexed by the monomials in $[\mathbf{x}]_t$, where we set $M_t(L)_{w,w'} = L(ww')$ for $w, w' \in [\mathbf{x}]_t$. Due to the commutativity of the variables, this matrix is smaller and more entries are now required to be equal. For instance, the (x_2x_1, x_3x_4)-entry of $M_2(L)$ is equal to its (x_3x_1, x_2x_4)-entry, which does not hold in general in the noncommutative case.

Given $a \in \mathbb{R}^n$, the *evaluation map* at a is the linear map $L_a \in \mathbb{R}[\mathbf{x}]^*$ defined by

$$L_a(p) = p(a_1, \ldots, a_n) \quad \text{for all} \quad p \in \mathbb{R}[\mathbf{x}]. \tag{4.15}$$

We can view L_a as a trace evaluation at 1×1 matrices. Moreover, we can view a trace evaluation map at a tuple of pairwise commuting matrices as a conic combination of evaluation maps at scalars by simultaneously diagonalizing the matrices.

The quadratic module $\mathcal{M}(S)$ and the ideal $\mathcal{I}(T)$ have immediate specializations to the commutative setting. We recall that in the commutative setting the (scalar) positivity domain and scalar variety of sets $S, T \subseteq \mathbb{R}[\mathbf{x}]$ are given by

$$D(S) = \big\{a \in \mathbb{R}^n : g(a) \geq 0 \text{ for } g \in S\big\}, \;\; V(T) = \big\{a \in \mathbb{R}^n : h(a) = 0 \text{ for } h \in T\big\}.^4 \tag{4.16}$$

We first give the commutative analogue of Theorem 4.5, where we give an additional integral representation in point (3). The equivalence of points (1) and (3) is proved in [Put93] based on Putinar's Positivstellensatz. Here we give a direct proof on the "moment side" using the Gelfand representation.

Theorem 4.9. *Let $S, T \subseteq \mathbb{R}[\mathbf{x}]$ with $\mathcal{M}(S) + \mathcal{I}(T)$ Archimedean. For $L \in \mathbb{R}[\mathbf{x}]^*$, the following are equivalent:*

(1) L is nonnegative on $\mathcal{M}(S)$, zero on $\mathcal{I}(T)$, and $L(1) = 1$;

(2) there exists a unital commutative C^*-algebra $\mathcal{A}$ with a state τ and $\mathbf{X} \in \mathcal{D}_{\mathcal{A}}(S) \cap \mathcal{V}_{\mathcal{A}}(T)$ such that $L(p) = \tau(p(\mathbf{X}))$ for all $p \in \mathbb{R}[\mathbf{x}]$;

(3) there exists a probability measure μ on $D(S) \cap V(T)$ such that

$$L(p) = \int_{D(S)\cap V(T)} p(x)\, d\mu(x) \quad \textit{for all} \quad p \in \mathbb{R}[\mathbf{x}].$$

Proof. ((1) ⇒ (2)) This is the commutative analogue of the implication (1) ⇒ (2) in Theorem 4.5 (observing in addition that the operators X_i in (4.10) pairwise commute so that the constructed C^*-algebra $\mathcal{A}$ is commutative).

((2) ⇒ (3)) Let $\widehat{\mathcal{A}}$ denote the set of unital $*$-homomorphisms $\mathcal{A} \to \mathbb{C}$, known as the *spectrum* of $\mathcal{A}$. We equip $\widehat{\mathcal{A}}$ with the weak-* topology, so that it is compact as a result of $\mathcal{A}$ being unital (see, e.g., [Bla06, II.2.1.4]). The *Gelfand representation* is the $*$-isomorphism

$$\Gamma\colon \mathcal{A} \to \mathcal{C}(\widehat{\mathcal{A}}), \quad \Gamma(a)(\phi) = \phi(a) \quad \text{for} \quad a \in \mathcal{A},\; \phi \in \widehat{\mathcal{A}},$$

where $\mathcal{C}(\widehat{\mathcal{A}})$ is the set of complex-valued continuous functions on $\widehat{\mathcal{A}}$. Since Γ is an isomorphism, the state τ on $\mathcal{A}$ induces a state τ' on $\mathcal{C}(\widehat{\mathcal{A}})$ defined by $\tau'(\Gamma(a)) = \tau(a)$ for $a \in \mathcal{A}$. By the Riesz representation theorem (see, e.g., [Rud87, Theorem 2.14]) there is a Radon measure ν on $\widehat{\mathcal{A}}$ such that

$$\tau'(\Gamma(a)) = \int_{\widehat{\mathcal{A}}} \Gamma(a)(\phi)\, d\nu(\phi) \quad \text{for all} \quad a \in \mathcal{A}.$$

We then have

$$\begin{aligned} L(p) = \tau(p(\mathbf{X})) = \tau'(\Gamma(p(\mathbf{X}))) &= \int_{\widehat{\mathcal{A}}} \Gamma(p(\mathbf{X}))(\phi)\, d\nu(\phi) = \int_{\widehat{\mathcal{A}}} \phi(p(\mathbf{X}))\, d\nu(\phi) \\ &= \int_{\widehat{\mathcal{A}}} p(\phi(X_1), \ldots, \phi(X_n))\, d\nu(\phi) = \int_{\widehat{\mathcal{A}}} p(f(\phi))\, d\nu(\phi) = \int_{\mathbb{R}^n} p(x)\, d\mu(x), \end{aligned}$$

[4] Note that in the commutative setting we could avoid using the variety since $V(T) = D(\pm T)$. However, in the noncommutative setting, the polynomials in T need not be symmetric in which case the quadratic module $\mathcal{D}(\pm T)$ would not be well defined.

where $f\colon \widehat{\mathcal{A}} \to \mathbb{R}^n$ is defined by $\phi \mapsto (\phi(X_1), \ldots, \phi(X_n))$, and where $\mu = f_*\nu$ is the pushforward measure of ν by f; that is, $\mu(B) = \nu(f^{-1}(B))$ for measurable $B \subseteq \mathbb{R}^n$.

Since $\mathbf{X} \in \mathcal{D}_{\mathcal{A}}(S)$, we have $g(\mathbf{X}) \succeq 0$ for all $g \in S$, hence $\Gamma(g(\mathbf{X}))$ is a positive element of $\mathcal{C}(\widehat{\mathcal{A}})$, implying $g(\phi(X_1), \ldots, \phi(X_n)) = \phi(g(\mathbf{X})) = \Gamma(g(\mathbf{X}))(\phi) \geq 0$. Similarly we see $h(\phi(X_1), \ldots, \phi(X_n)) = 0$ for all $h \in T$. So, the range of f is contained in $D(S) \cap V(T)$, μ is a probability measure on $D(S) \cap V(T)$ since $L(1) = 1$, and we have $L(p) = \int_{D(S)\cap V(T)} p(x)\, d\mu(x)$ for all $p \in \mathbb{R}[\mathbf{x}]$.

((3) ⇒ (1)) This is immediate. □

We point out that the more common proof for the implication (1) ⇒ (3) in Theorem 4.9 relies on Putinar's Positivstellensatz [Put93]: if L satisfies (1) then $L(p) \geq 0$ for all polynomials p nonnegative on $D(S) \cap V(T)$ (since Putinar's Positivstellensatz shows $p + \varepsilon \in \mathcal{M}(S) + \mathcal{I}(T)$ for any $\varepsilon > 0$), and thus L has a representing measure μ as in (3) by the Riesz-Haviland theorem [Hav36].

The following is the commutative analogue of Theorem 4.6.

Theorem 4.10. *For $S \subseteq \mathbb{R}[\mathbf{x}]$, $T \subseteq \mathbb{R}[\mathbf{x}]$, and $L \in \mathbb{R}[\mathbf{x}]^*$, the following are equivalent:*

(1) L is nonnegative on $\mathcal{M}(S)$, zero on $\mathcal{I}(T)$, has $\operatorname{rank}(M(L)) < \infty$, and satisfies $L(1) = 1$;

(2) there is a finite-dimensional commutative C^*-algebra $\mathcal{A}$ with a state τ, and $\mathbf{X} \in \mathcal{D}_{\mathcal{A}}(S) \cap \mathcal{V}_{\mathcal{A}}(T)$ such that $L(p) = \tau(p(\mathbf{X}))$ for all $p \in \mathbb{R}[\mathbf{x}]$;

(3) L is a convex combination of evaluations at points in $D(S) \cap V(T)$.

Proof. ((1) ⇒ (2)) We indicate how to derive this claim from its noncommutative analogue. For this denote the commutative version of $p \in \mathbb{R}\langle\mathbf{x}\rangle$ by $p^c \in \mathbb{R}[\mathbf{x}]$. For any $g \in S$ and $h \in T$, select symmetric polynomials $g', h' \in \mathbb{R}\langle\mathbf{x}\rangle$ with $(g')^c = g$ and $(h')^c = h$, and set

$$S' = \{g' : g \in S\} \subseteq \mathbb{R}\langle\mathbf{x}\rangle$$

and

$$T' = \{h' : h \in T\} \cup \{x_i x_j - x_j x_i \in \mathbb{R}\langle\mathbf{x}\rangle : i, j \in [n], i \neq j\} \subseteq \mathbb{R}\langle\mathbf{x}\rangle.$$

Define the linear form $L' \in \mathbb{R}\langle\mathbf{x}\rangle^*$ by $L'(p) = L(p^c)$ for $p \in \mathbb{R}\langle\mathbf{x}\rangle$. Then L' is symmetric, tracial, nonnegative on $\mathcal{M}(S')$, zero on $\mathcal{I}(T')$, and satisfies $\operatorname{rank} M(L') = \operatorname{rank} M(L) < \infty$. Following the proof of the implication (1) ⇒ (2) in Theorem 4.5, we see that the operators $X_1, \ldots, X_n$ pairwise commute (since $\mathbf{X} \in \mathcal{V}_{\mathcal{A}}(T')$ and T' contains all $x_i x_j - x_j x_i$) and thus the constructed C^*-algebra $\mathcal{A}$ is finite-dimensional and commutative.

((2) ⇒ (3)) Here we follow the proof of this implication in Theorem 4.6 and observe that since $\mathcal{A}$ is finite-dimensional and commutative, it is $*$-isomorphic to an algebra of diagonal matrices ($d_m = 1$ for all $m \in [M]$), which directly gives the desired result.

((3) ⇒ (1)) is easy. □

The next result, due to Curto and Fialkow [CF96], is the commutative analogue of Corollary 4.8.

Theorem 4.11. *Let $1 \leq \delta \leq t < \infty$ and $S, T \subseteq \mathbb{R}[\mathbf{x}]_{2\delta}$. If $L \in \mathbb{R}[\mathbf{x}]_{2t}^*$ is δ-flat, nonnegative on $\mathcal{M}_{2t}(S)$, and zero on $\mathcal{I}_{2t}(T)$, then L extends to a conic combination of evaluation maps at points in $D(S) \cap V(T)$.*

Proof. Here too we derive the result from its noncommutative analogue in Corollary 4.8. As in the above proof for the implication (1) $\Longrightarrow$ (2) in Theorem 4.10, define the sets $S', T' \subseteq \mathbb{R}\langle\mathbf{x}\rangle$ and the linear form $L' \in \mathbb{R}\langle\mathbf{x}\rangle_{2t}^*$ by $L'(p) = L(p^c)$ for $p \in \mathbb{R}\langle\mathbf{x}\rangle_{2t}$. Then L' is symmetric, tracial, nonnegative on $\mathcal{M}_{2t}(S')$, zero on $\mathcal{I}_{2t}(T')$, and δ-flat. By Corollary 4.8, L' is a conic combination of trace evaluation maps at elements of $\mathcal{D}(S') \cap \mathcal{V}(T')$. It suffices now to observe that such a trace evaluation $L_{\mathbf{X}}$ is a conic combination of (scalar) evaluations at elements of $D(S) \cap V(T)$. Indeed, as $\mathbf{X} \in \mathcal{V}(T')$, the matrices $X_1, \ldots, X_n$ pairwise commute and thus can be assumed to be diagonal. Since $\mathbf{X} \in \mathcal{D}(S') \cap \mathcal{V}(T')$, we have $g(\mathbf{X}) \succeq 0$ for $g' \in S'$ and $h'(\mathbf{X}) = 0$ for $h' \in T'$. This implies $g((X_1)_{jj}, \ldots, (X_n)_{jj}) \geq 0$ and $h((X_1)_{jj}, \ldots, (X_n)_{jj}) = 0$ for all $g \in S$, $h \in T$, and $j \in [d]$. Thus $L_{\mathbf{X}} = \sum_j L_{r_j}$, where $r_j = ((X_1)_{jj}, \ldots, (X_n)_{jj}) \in D(S) \cap V(T)$. □

Unlike in the noncommutative setting, here we also have the following result, which permits to express any linear functional L nonnegative on an Archimedean quadratic module as a conic combination of evaluations at points, when restricting L to polynomials of bounded degree.

Theorem 4.12. *Let $S, T \subseteq \mathbb{R}[\mathbf{x}]$ such that $\mathcal{M}(S) + \mathcal{I}(T)$ is Archimedean. If $L \in \mathbb{R}[\mathbf{x}]^*$ is nonnegative on $\mathcal{M}(S)$ and zero on $\mathcal{I}(T)$, then for any integer $k \in \mathbb{N}$ the restriction of L to $\mathbb{R}[\mathbf{x}]_k$ extends to a conic combination of evaluations at points in $D(S) \cap V(T)$.*

Proof. By Theorem 4.9 there exists a probability measure μ on $D(S)$ such that

$$L(p) = L(1) \int_{D(S) \cap V(T)} p(x)\, d\mu(x) \quad \text{for all} \quad p \in \mathbb{R}[\mathbf{x}].$$

A general version of Tchakaloff's theorem, as explained in [BT06], shows that there exist $r \in \mathbb{N}$, scalars $\lambda_1, \ldots, \lambda_r > 0$ and points $x_1, \ldots, x_r \in D(S)$ such that

$$\int_{D(S) \cap V(T)} p(x)\, d\mu(x) = \sum_{i=1}^{r} \lambda_i p(x_i) \quad \text{for all} \quad p \in \mathbb{R}[\mathbf{x}]_k.$$

Hence the restriction of L to $\mathbb{R}[\mathbf{x}]_k$ extends to a conic combination of evaluations at points in $D(S)$. □

4.2 Commutative and tracial polynomial optimization

We briefly discuss here the basic polynomial optimization problems in the commutative and tracial settings. We recall how to design hierarchies of semidefinite programming based bounds and we give their main convergence properties.

The classical commutative polynomial optimization problem asks to minimize a polynomial $f \in \mathbb{R}[\mathbf{x}]$ over a feasible region of the form $D(S)$ as defined in (4.16):

$$f_* = \inf_{a \in D(S)} f(a) = \inf\big\{f(a) : a \in \mathbb{R}^n,\ g(a) \geq 0 \text{ for } g \in S\big\}. \tag{4.17}$$

In tracial polynomial optimization, given $f \in \operatorname{Sym}\mathbb{R}\langle\mathbf{x}\rangle$, this is modified to the problem of minimizing $\operatorname{tr}(f(\mathbf{X}))$ over a feasible region of the form $\mathcal{D}(S) \cap \mathcal{V}(T)$ (as defined in (4.7) and (4.8)):

$$\begin{aligned} f_*^{\mathrm{tr}} = \inf\big\{\operatorname{tr}(f(\mathbf{X})) : d \in \mathbb{N},\ \mathbf{X} \in (\mathrm{H}^d)^n,& \\ g(\mathbf{X}) \succeq 0 \text{ for } g \in S,\ h(\mathbf{X}) = 0 \text{ for } h \in T\big\},& \end{aligned}$$

where $\operatorname{tr}(\cdot)$ is the normalized trace. Observe that the infimum does not change if we replace H^d by S^d in view of the embedding of H^d into S^{2d} that we have seen in Equation (2.1). Commutative polynomial optimization is recovered by restricting to 1 × 1 matrices.

For the commutative case, Lasserre [Las01] and Parrilo [Par00] have proposed hierarchies of semidefinite programming relaxations based on sums of squares of polynomials and the dual theory of moments. This approach has been extended to eigenvalue optimization in [PNA10, NPA12] and later to tracial optimization in [BCKP13, KP16]. The starting point in deriving these relaxations is to reformulate the above problems as minimizing $L(f)$ over all normalized trace evaluation maps L at points in $D(S)$ or $\mathcal{D}(S) \cap \mathcal{V}(T)$. We then express computationally tractable properties satisfied by such maps L and truncate to polynomials of finite degree t. Notice that in the commutative setting we do not need to work with the variety $V(T)$, since $V(T) = D(T \cup -T)$.

For a set $S \subseteq \mathbb{R}[\mathbf{x}]$ and $t \in \mathbb{N} \cup \{\infty\}$, recall the (truncated) quadratic module:

$$\mathcal{M}_{2t}(S) = \operatorname{cone}\big\{gp^2 : p \in \mathbb{R}[\mathbf{x}],\ g \in S \cup \{1\},\ \deg(gp^2) \leq 2t\big\}. \tag{4.18}$$

For a polynomial $f \in \mathbb{R}[\mathbf{x}]$ and $t \geq \lceil \deg(f)/2 \rceil$ we can use the quadratic module to formulate the following semidefinite programming lower bound on f_*:

$$f_t = \inf\big\{L(f) : L \in \mathbb{R}[\mathbf{x}]_{2t}^*,\ L(1) = 1,\ L \geq 0 \text{ on } \mathcal{M}_{2t}(S)\big\}.$$

For $t \in \mathbb{N}$ we have $f_t \leq f_\infty \leq f_*$.

In the same way, for sets $S \cup \{f\} \subseteq \operatorname{Sym}\mathbb{R}\langle\mathbf{x}\rangle$, and $T \subseteq \mathbb{R}\langle\mathbf{x}\rangle$, and $t \in \mathbb{N} \cup \{\infty\}$ such that $\lceil \deg(f)/2 \rceil \leq t \leq \infty$, we have the following semidefinite programming lower bound on f_*^{tr}:

$$\begin{aligned} f_t^{\mathrm{tr}} = \inf\big\{L(f) : L \in \mathbb{R}\langle\mathbf{x}\rangle_{2t}^* \text{ tracial and symmetric, } L(1) = 1,& \\ L \geq 0 \text{ on } \mathcal{M}_{2t}(S), L = 0 \text{ on } \mathcal{I}_{2t}(T)\big\},& \end{aligned}$$

where we now use definition (4.2) for $\mathcal{M}_{2t}(S)$ and (4.3) for $\mathcal{I}_{2t}(T)$.

The next theorem from [Las01] gives fundamental convergence properties for the commutative case; see also, e.g., [Las09, Lau09] for a detailed exposition.

Theorem 4.13. *Let $1 \leq \delta \leq t < \infty$ and $S \cup \{f\} \subseteq \mathbb{R}[\mathbf{x}]_{2\delta}$ with $D(S) \neq \emptyset$.*

- *(i)* *If $\mathcal{M}(S)$ is Archimedean, then $f_t \to f_\infty$ as $t \to \infty$, the optimal values in f_∞ and f_* are attained, and $f_\infty = f_*$.*

- *(ii)* *If f_t admits an optimal solution L that is δ-flat, then L is a convex combination of evaluation maps at global minimizers of f in $D(S)$, and $f_t = f_\infty = f_*$.*

Proof. (i) By repeating the first part of the proof of Theorem 4.14 in the commutative setting we see that $f_t \to f_\infty$ and that the optimum is attained in f_∞. Let L be optimal for f_∞ and let k be greater than $\deg(f)$ and $\deg(g)$ for $g \in S$. By Theorem 4.12, the restriction of L to $\mathbb{R}[\mathbf{x}]_k$ extends to a conic combination of evaluations at points in $D(S)$. It follows that this extension if feasible for f_* with the same objective value, which shows $f_\infty = f_*$.

(ii) This follows in the same way as the proof of Theorem 4.14(ii) below, where, instead of using Corollary 4.8, we now use its commutative analogue, Theorem 4.11. □

To discuss convergence for the tracial case we need one more optimization problem:[5]

$$\begin{aligned} f_{\mathrm{II}_1}^{\mathrm{tr}} = \inf\{\tau(f(\mathbf{X})) : \mathcal{A} \text{ is a unital } C^*\text{-algebra with tracial state } \tau, \\ \mathbf{X} \in \mathcal{D}_{\mathcal{A}}(S) \cap \mathcal{V}_{\mathcal{A}}(T)\}. \end{aligned} \tag{4.19}$$

This problem can be seen as an infinite-dimensional analogue of f_*^{tr}: if we restrict to finite-dimensional C^*-algebras in the definition of $f_{\mathrm{II}_1}^{\mathrm{tr}}$, then we recover the parameter f_*^{tr} (use Theorem 4.6 to see this). Moreover, as we see in Theorem 4.14(ii) below, equality $f_*^{\mathrm{tr}} = f_{\mathrm{II}_1}^{\mathrm{tr}}$ holds if some flatness condition is satisfied. Whether $f_{\mathrm{II}_1}^{\mathrm{tr}} = f_*^{\mathrm{tr}}$ is true in general is related to Connes' embedding conjecture (see [KS08, KP16, BKP16]).

For all $t \in \mathbb{N}$ we have

$$f_t^{\mathrm{tr}} \le f_{t+1}^{\mathrm{tr}} \le f_\infty^{\mathrm{tr}} \le f_{\mathrm{II}_1}^{\mathrm{tr}} \le f_*^{\mathrm{tr}},$$

where the last inequality follows by considering for $\mathcal{A}$ the full matrix algebra $\mathbb{C}^{d\times d}$. The next theorem from [KP16] summarizes convergence properties for these parameters, its proof uses Lemma 4.15 below.

Theorem 4.14. *Let $1 \le \delta \le t < \infty$, $S \cup \{f\} \subseteq \mathrm{Sym}\,\mathbb{R}\langle\mathbf{x}\rangle_{2\delta}$, and $T \subseteq \mathbb{R}\langle\mathbf{x}\rangle_{2\delta}$, with $\mathcal{D}(S) \cap \mathcal{V}(T) \neq \emptyset$.*

- *(i)* *If $\mathcal{M}(S) + \mathcal{I}(T)$ is Archimedean, then $f_t^{tr} \to f_\infty^{\mathrm{tr}}$ as $t \to \infty$, and the optimal values in f_∞^{tr} and $f_{\mathrm{II}_1}^{\mathrm{tr}}$ are attained and equal.*

- *(ii)* *If f_t^{tr} has an optimal solution L that is δ-flat, then L is a convex combination of normalized trace evaluations at matrix tuples in $\mathcal{D}(S) \cap \mathcal{V}(T)$, and $f_t^{tr} = f_\infty^{tr} = f_{\mathrm{II}_1}^{\mathrm{tr}} = f_*^{\mathrm{tr}}$.*

[5] The subscript II_1 in $f_{\mathrm{II}_1}^{\mathrm{tr}}$ comes from the more usual definition of this parameter using von Neumann algebras of type II_1, see [KP16]. See [GdLL19, App. A] for a short discussion on the equivalence of the two definitions.

Proof. We first show (i). As the Minkowski sum $\mathcal{M}(S)+\mathcal{I}(T)$ is Archimedean (4.4), $R-\sum_{i=1}^n x_i^2 \in \mathcal{M}_{2d}(S)\mathcal{I}_{2d}(T)$ for some $R>0$ and $d\in\mathbb{N}$. Since the bounds f_t^{tr} are monotone nondecreasing in t and upper bounded by f_∞^{tr}, the limit $\lim_{t\to\infty} f_t^{\mathrm{tr}}$ exists and it is at most f_∞^{tr}.

Fix $\varepsilon>0$. For $t\in\mathbb{N}$ let L_t be a feasible solution to the program defining f_t^{tr} with value $L_t(f)\le f_t^{\mathrm{tr}}+\varepsilon$. As $L_t(1)=1$ for all t we can apply Lemma 4.15 and conclude that the sequence $(L_t)_t$ has a convergent subsequence. Let $L\in\mathbb{R}\langle\mathbf{x}\rangle^*$ be the pointwise limit. One can easily check that L is feasible for f_∞^{tr}. Hence we have $f_\infty^{\mathrm{tr}}\le L(f)\le \lim_{t\to\infty} f_t^{\mathrm{tr}}+\varepsilon\le f_\infty^{\mathrm{tr}}+\varepsilon$. Letting $\varepsilon\to 0$ we obtain that $f_\infty^{\mathrm{tr}}=\lim_{t\to\infty} f_t^{\mathrm{tr}}$ and L is optimal for f_∞^{tr}.

Next, since L is symmetric, tracial, nonnegative on $\mathcal{M}(S)$, and zero on $\mathcal{I}(T)$, we can apply Theorem 4.5 to obtain a feasible solution $(\mathcal{A},\tau,\mathbf{X})$ to $f_{\mathrm{II}_1}^{\mathrm{tr}}$ satisfying (4.9) with objective value $L(f)$. This shows $f_\infty^{\mathrm{tr}}=f_{\mathrm{II}_1}^{\mathrm{tr}}$ and that the optima are attained in f_∞^{tr} and $f_{\mathrm{II}_1}^{\mathrm{tr}}$.

Finally, part (ii) is derived as follows. If L is an optimal solution of f_t^{tr} that is δ-flat, then, by Corollary 4.8, it has an extension $\hat{L}\in\mathbb{R}\langle\mathbf{x}\rangle^*$ that is a conic combination of trace evaluations at elements of $\mathcal{D}(S)\cap\mathcal{V}(T)$. This shows that $f_*^{\mathrm{tr}}\le\hat{L}(f)=L(f)$, and thus the chain of equalities $f_t^{\mathrm{tr}}=f_\infty^{\mathrm{tr}}=f_*^{\mathrm{tr}}=f_{\mathrm{II}_1}^{\mathrm{tr}}$ holds. □

We conclude with the following technical lemma, based on the Banach-Alaoglu theorem. It is a well-known crucial tool for proving the asymptotic convergence result from Theorem 4.14(i) and it is used at other places in this thesis.

Lemma 4.15. *Let $S\subseteq\operatorname{Sym}\mathbb{R}\langle\mathbf{x}\rangle$, $T\subseteq\mathbb{R}\langle\mathbf{x}\rangle$, and assume $R-(x_1^2+\cdots+x_n^2)\in\mathcal{M}_{2d}(S)+\mathcal{I}_{2d}(T)$ for some $d\in\mathbb{N}$ and $R>0$. For $t\in\mathbb{N}$ assume $L_t\in\mathbb{R}\langle\mathbf{x}\rangle_{2t}^*$ is tracial, nonnegative on $\mathcal{M}_{2t}(S)$, and zero on $\mathcal{I}_{2t}(T)$. Then we have*

$$|L_t(w)|\le R^{|w|/2}L_t(1)\quad \textit{for all } w\in\langle\mathbf{x}\rangle_{2t-2d+2}.$$

In addition, if $\sup_t L_t(1)<\infty$, then $\{L_t\}_t$ has a pointwise converging subsequence in $\mathbb{R}\langle\mathbf{x}\rangle^*$.

Proof. We first use induction on $|w|$ to show that $L_t(w^*w)\le R^{|w|}L_t(1)$ for all $w\in\langle\mathbf{x}\rangle_{t-d+1}$. For this, assume $L_t(w^*w)\le R^{|w|}L_t(1)$ and $|w|\le t-d$. Then we have

$$L_t((x_iw)^*x_iw)=L_t(w^*(x_i^2-R)w)+R\cdot L_t(w^*w)\le R\cdot R^{|w|}L_t(1)=R^{|x_iw|}L_t(1).$$

For the inequality we use the fact that $L_t(w^*(x_i^2-R)w)\le 0$ since $w^*(R-x_i^2)w$ can be written as the sum of a polynomial in $\mathcal{M}_{2t}(S)+\mathcal{I}_{2t}(T)$ and a sum of commutators of degree at most $2t$, which follows using the following identity: $w^*qhw=ww^*qh+[w^*qh,w]$. Next we write any $w\in\langle\mathbf{x}\rangle_{2(t-d+1)}$ as $w=w_1^*w_2$ with $w_1,w_2\in\langle\mathbf{x}\rangle_{t-d+1}$ and use the positive semidefiniteness of the principal submatrix of $M_t(L_t)$ indexed by $\{w_1,w_2\}$ to get

$$L_t(w)^2=L_t(w_1^*w_2)^2\le L_t(w_1^*w_1)L_t(w_2^*w_2)\le R^{|w_1|+|w_2|}L_t(1)^2=R^{|w|}L_t(1)^2.$$

This shows the first claim.

Suppose $c := \sup_t L_t(1) < \infty$. For each $t \in \mathbb{N}$, consider the linear functional $\hat{L}_t \in \mathbb{R}\langle \mathbf{x} \rangle^*$ defined by $\hat{L}_t(w) = L_t(w)$ if $|w| \leq 2t - 2d + 2$ and $\hat{L}_t(w) = 0$ otherwise. Then the vector $(\hat{L}_t(w)/(cR^{|w|/2}))_{w \in \langle \mathbf{x} \rangle}$ lies in the supremum norm unit ball of $\mathbb{R}^{\langle \mathbf{x} \rangle}$, which is compact in the weak$*$ topology by the Banach–Alaoglu theorem. It follows that the sequence $(\hat{L}_t)_t$ has a pointwise converging subsequence and thus the same holds for the sequence $(L_t)_t$. □

4.3 Advantages and disadvantages of (non)commutativity

We want to point out some fundamental differences between the (finite) convergence behavior of the Lasserre hierarchy for commutative and noncommutative (tracial) polynomial optimization. We do so by means of three simple settings: the feasible region $\mathcal{D}(S) \cap \mathcal{V}(T)$ is

(i) the ball: $S = \{1 - \sum_{i \in [n]} x_i^2\}$, $T = \emptyset$,

(ii) the cube: $S = \{1 - x_i^2 : i \in [n]\}$, $T = \emptyset$,

(iii) (a subset of) the hypercube: $\{1 - x_i^2 : i \in [n]\} \subseteq T$.

Note that in all three settings we have that $\mathcal{M}(S) + \mathcal{I}(T)$ is Archimedean so that the Lasserre hierarchy converges.

Let us first describe the commutative setting of minimizing a polynomial $f(x)$ over $x \in D(S) \cap V(T)$ for some finite sets of $S, T \subseteq \mathbb{R}[\mathbf{x}]$. It is well known that if $|V(T)| < \infty$, then the Lasserre hierarchy converges in finitely many steps: there is a t such that $f_t = f_*$ (see, e.g., [Lau09, Thm. 6.15]). A special case is polynomial optimization over a subset of the hypercube, in that setting we even know that $t = n$ suffices.[6] It is important to note that finite convergence does not happen in general. For example, in the very simple setting of minimizing a polynomial over the ball one can find examples where $f_t < f_*$ for all $t \in \mathbb{N}$, see, e.g., [Lau09, Ex. 6.19]. Nevertheless, examples where there is no finite convergence are rare: Nie showed that, in the Archimedean setting, one generically has finite convergence [Nie14b] (here the distribution is on the input polynomials).

As we have mentioned before, the noncommutative setting contains both tracial optimization and eigenvalue optimization. Since the convergence behavior of the Lasserre relaxations differs between these problems, we treat them separately. As we have seen, the tracial optimization problem corresponds to minimizing $\tau(f)$ over *tracial* states τ on unital (finite-dimensional) C^*-algebras. When we remove the tracial condition on τ, we obtain the eigenvalue optimization problem:

$$\inf\big\{\tau(f(\mathbf{X})) : \mathcal{A} \text{ is a unital } C^*\text{-algebra with state } \tau,\\ \mathbf{X} \in \mathcal{D}_{\mathcal{A}}(S) \cap \mathcal{V}_{\mathcal{A}}(T)\big\},$$

[6] To see why $t = 2n$ suffices, a sub-optimal result, note that $\mathbb{R}[\mathbf{x}]/\mathcal{I}(T)$ consists of multilinear polynomials, which makes any $L \in \mathbb{R}[\mathbf{x}]_{2n}^*$ that satisfies $L = 0$ on $\mathcal{I}(T)$ at least n-flat. The result then follows from Theorem 4.13.

where $\{f\} \cup S \subseteq \operatorname{Sym}\mathbb{R}\langle \mathbf{x}\rangle, T \subseteq \mathbb{R}\langle \mathbf{x}\rangle$. To explain why this is called an eigenvalue optimization problem we have to view a C^*-algebra $\mathcal{A}$ as the $*$-algebra $\mathcal{B}(\mathcal{H})$ of bounded operators on a Hilbert space (using the GNS construction, see Theorem 4.4). We see that the above problem is equivalent to

$$\begin{aligned}\inf\bigl\{\langle\psi, f(\mathbf{X})\psi\rangle : \text{Hilbert space } \mathcal{H}, \text{ unit vector } \psi \in \mathcal{H}, \mathbf{X} \in (\mathcal{B}(\mathcal{H}))^n,\\ g(\mathbf{X}) \succeq 0 \text{ for } g \in S, h(\mathbf{X}) = 0 \text{ for } h \in T\bigr\}.\end{aligned} \tag{4.20}$$

Here the Hilbert space $\mathcal{H}$, the unit vector ψ, and the tuple $\mathbf{X} \in (\mathcal{B}(\mathcal{H}))^n$ are the variables. Similarly to the commutative and tracial settings, one has a hierarchy of semidefinite programs $\{f_t^{\mathrm{nc}}\}_{t\in\mathbb{N}\cup\{\infty,*\}}$ that converges to (4.20) as $t \to \infty$ (under the assumption that $\mathcal{M}(S) + \mathcal{I}(T)$ is Archimedean) [PNA10].

We now discuss the convergence behavior of f_t^{nc}. Perhaps somewhat surprisingly, the situation is completely reversed in the eigenvalue optimization setting compared to the commutative setting. For the ball or the cube, one has finite convergence $f^{\mathrm{nc}}_{\lceil \deg(f)/2\rceil+1} = f_*^{\mathrm{nc}}$ [CKP12]. But, for eigenvalue optimization over the hypercube there is no such guarantee: one cannot find a finite order $z(n) \in \mathbb{N}$ such that $f^{\mathrm{nc}}_{z(n)} = f^{\mathrm{nc}}_\infty$ holds for all eigenvalue optimization problems in n variables that have $x_i^2 - 1 \in T$ for all $i \in [n]$. The latter is a consequence of Slofstra's work mentioned in footnote 4 in Chapter 3 and can be seen as follows.

The maximum winning probability of a nonlocal game G over strategies in $C_{qc}(\Gamma)$ can be written as an eigenvalue minimization problem [NPA08]. Let us call the corresponding eigenvalue minimization problem $f^{\mathrm{nc}}_\infty(G)$. For completeness we mention that the problems $f^{\mathrm{nc}}_\infty(G)$ include the constraints $x_i^2 - 1$ for all $i \in [n]$. Slofstra showed that there exists a certain class of games for which the problem of determining if that probability equals 1 is undecidable [Slo16]. Hence, for these games G there does not exist a function $z : \mathbb{N} \to \mathbb{N}$ such that $f^{\mathrm{nc}}_{z(n)}(G) = f^{\mathrm{nc}}_\infty(G)$ (where n is the number of variables in the game G).

Slofstra's results are not constructive in the sense that they do not provide explicit optimization problems for which there is a gap between f_t^{nc} and f_∞^{nc}. Recently, an eigenvalue optimization problem on the noncommutative hypercube was constructed for which there is an explicit gap even at the 2^{n-1}th order: in [BWHKN18] it is shown that there exists a family of nonlocal games called *Capped GHZ games*, $\{CG_n\}_{n\in\mathbb{N}}$, for which the maximum winning probability over commuting-operator strategies f_∞^{nc} is at most $1 - 1/e^n$, while the order 2^{n-1} relaxation has $f^{\mathrm{nc}}_{2^{n-1}} = 1$.[7]

What about the tracial setting? Again the situation is somewhat different. One no longer has finite convergence on the cube, as can be seen from Example 4.3 of [KS08]. That example shows that the hierarchy $\{f_t^{\mathrm{tr}}\}$ for minimizing the normalized trace $\operatorname{tr}((1-x^2)(1-y^2))$ over the cube satisfies $f_t^{\mathrm{tr}} < f_*^{\mathrm{tr}}$ for all $t \in \mathbb{N}$.

Algebraic certificates of nonnegativity. Let us now look at the dual problem of algebraically certifying nonnegativity of polynomials: Positivstellensätze. Again, there will be a difference between the commutative, the eigenvalue, and the tracial

[7] The eigenvalue optimization problem corresponding to CG_n has poly(n) variables.

settings. To observe this difference it suffices to consider polynomials on the cube, that is, we consider the problem of algebraically certifying nonnegativity of polynomials that are nonnegative whenever we evaluate them on elements in $[-1,1]^n$ or the noncommutative analogue $\mathcal{D}(\{1-x_i^2 : i \in [n]\})$.

Let us first mention the 'nice' cases where there is a characterization of nonnegative polynomials on the cube: the commutative setting and the noncommutative eigenvalue setting. The characterizations below are special cases of the Positivstellensätze of Putinar and Helton, McCullough.

Theorem 4.16 (Putinar [Put93]). *For every $f \in \mathbb{R}[\mathbf{x}]$, the following are equivalent:*

(i) $f \geq 0$ *on* $[-1,1]^n$*;*

(ii) For all $\varepsilon > 0$*, the polynomial* $f + \varepsilon$ *belongs to the quadratic module*

$$\mathcal{M}(\{1-x_i^2 : i \in [n]\}).$$

Theorem 4.17 (Helton & McCullough [HM04]). *For every $f \in \operatorname{Sym}\mathbb{R}\langle\mathbf{x}\rangle$, the following are equivalent:*

(i) $f(A_1,\dots,A_n) \succeq 0$ *for all* $s \in \mathbb{N}$ *and contractions* $A_1,\dots,A_n \in \mathrm{S}^s$*;*

(ii) For all $\varepsilon > 0$*, the polynomial* $f + \varepsilon$ *belongs to the quadratic module*

$$\mathcal{M}(\{1-x_i^2 : i \in [n]\}).$$

Based on the above two theorems one would expect that a similar result holds for certifying trace nonnegativity on the cube. However, as Klep and Schweighofer showed in [KS08], the analogous statement in the tracial setting is equivalent to Connes' embedding conjecture, a long-standing open problem in operator theory [Con76, pp. 105–107].

Conjecture 4.18 (Algebraic version of Connes' conjecture [KS08]). *Suppose $f \in \operatorname{Sym}\mathbb{R}\langle\mathbf{x}\rangle$. Then the following are equivalent:*

(i) $\mathrm{tr}(f(A_1,\dots,A_n)) \geq 0$ *for all* $s \in \mathbb{N}$ *and contractions* $A_1,\dots,A_n \in \mathrm{S}^s$*.*

(ii) For every $\varepsilon > 0$*, the polynomial* $f+\varepsilon$ *is cyclically equivalent*[8] *to an element in the quadratic module* $\mathcal{M}(\{1-x_i^2 : i \in [n]\})$*.*

Klep and Schweighofer showed that Connes' embedding conjecture holds if and only if the implication (i) ⇒ (ii) in Conjecture 4.18 holds for all $n \in \mathbb{N}$ and polynomials $f \in \operatorname{Sym}\mathbb{R}\langle\mathbf{x}\rangle$. (Note that the implication (ii) ⇒ (i) is trivial.)

[8] A polynomial $f \in \mathbb{R}\langle\mathbf{x}\rangle$ is cyclically equivalent to a polynomial $g \in \mathbb{R}\langle\mathbf{x}\rangle$ if the difference $f-g$ can be written as a sum of commutators $fg - gf$.

4.4 Summary of main results

For convenience, below we give a short summary of the main convergence results of the hierarchies $\{f_t\}$ and $\{f_t^{\mathrm{tr}}\}$, including pointers to the relevant results/notions. Let us first restate the hierarchies:

$$\begin{aligned} f_t &= \inf\big\{L(f) : L \in \mathbb{R}[\mathbf{x}]_{2t}^*,\ L(1) = 1,\ L \geq 0 \text{ on } \mathcal{M}_{2t}(S)\big\}, \\ f_t^{\mathrm{tr}} &= \inf\big\{L(f) : L \in \mathbb{R}\langle\mathbf{x}\rangle_{2t}^* \text{ tracial and symmetric, } L(1) = 1, \\ &\qquad\qquad L \geq 0 \text{ on } \mathcal{M}_{2t}(S), L = 0 \text{ on } \mathcal{I}_{2t}(T)\big\}. \end{aligned}$$

Here $t \in \mathbb{N} \cup \{\infty\}$. When $t = *$ we consider the problem corresponding to $t = \infty$ with the additional constraint that the moment matrix $M(L)$ has finite rank. The hierarchy $\{f_t\}$ lower bounds the commutative polynomial optimization problem (4.17) and similarly $\{f_t^{\mathrm{tr}}\}$ lower bounds the tracial polynomial optimization problem (4.19). The truncated quadratic module $\mathcal{M}_{2t}(S)$ is defined in the noncommutative setting in Equation (4.2), and in the commutative setting in Equation (4.18). The truncated left ideal $\mathcal{I}_{2t}(T)$ is defined in Equation (4.3).

Under the condition that $\mathcal{M}(S)+\mathcal{I}(T)$ satisfies the Archimedean condition (4.4) we have asymptotic convergence:

$$f_t \to f_\infty \qquad \text{and} \qquad f_t^{\mathrm{tr}} \to f_\infty^{\mathrm{tr}} \qquad \text{as} \qquad t \to \infty,$$

see Theorem 4.13 and Theorem 4.14. In the commutative setting one can moreover show that f_∞ is equal to the global minimum of f over the set $D(S)$ (4.17). In the noncommutative setting, the parameter f_∞^{tr} is equal to the C^*-algebraic version of the tracial optimization problem (4.19). In general this is not equal to the minimum of $\mathrm{tr}(f(\mathbf{X}))$ over $\mathbf{X}$ in the intersection of the matrix positivity domain $\mathcal{D}(S)$ (4.7) and matrix variety $\mathcal{V}(T)$ (4.8), that minimum is equal to the parameter f_*^{tr}.

For both hierarchies there is an important additional convergence result under flatness. If the program defining the bound f_t admits a sufficiently flat optimal solution, then equality holds: $f_t = f_\infty$. Similarly, if f_t^{tr} admits a sufficiently flat optimal solution, then $f_t^{\mathrm{tr}} = f_\infty^{\mathrm{tr}}$. Moreover, in this case, the parameter f_t^{tr} is equal to f_*^{tr}, the minimum value of $\mathrm{tr}(f(\mathbf{X}))$ over $\mathcal{D}(S) \cap \mathcal{V}(T)$.

Part I

Lower bounds on factorization ranks

Chapter 5

Lower bounds on matrix factorization ranks via polynomial optimization

This chapter is based on the paper "Lower bounds on matrix factorization ranks via noncommutative polynomial optimization", by S. Gribling, D. de Laat, and M. Laurent [GdLL19].

In this chapter we start our study of matrix factorization ranks. Using techniques from (noncommutative) polynomial optimization, we provide a unified approach to obtain lower bounds. This results in a hierarchy of semidefinite programming lower bounds for each of the factorization ranks. We study the convergence properties of our hierarchies and provide some (numerical) examples.

Organization. This chapter is organized as follows. We first sketch the main ideas of our approach and we relate our approach to the more classical use of polynomial optimization. We then give an overview of our results and we compare them with known lower bounds on the respective matrix factorization ranks. The main body of the chapter consists of four sections, each dealing with one of the four matrix factorization ranks: the nonnegative rank, the positive semidefinite rank, and their symmetric analogues, the completely positive rank and the completely positive semidefinite rank.

5.1 Basic approach

To explain the basic idea of how we obtain lower bounds for matrix factorization ranks we consider the case of the completely positive semidefinite rank. Given a minimal factorization $A = (\mathrm{Tr}(X_iX_j))$, with $d = \text{cpsd-rank}_{\mathbb{C}}(A)$ and $\mathbf{X} = (X_1, \ldots, X_n)$ in $(\mathrm{H}_+^d)^n$, consider the linear form $L_{\mathbf{X}}$ on $\mathbb{R}\langle \mathbf{x} \rangle$, the trace evaluation at $\mathbf{X}$, as defined in (4.6):

$$L_{\mathbf{X}}(p) = \mathrm{Re}(\mathrm{Tr}(p(X_1, \ldots, X_n))) \quad \text{for} \quad p \in \mathbb{R}\langle \mathbf{x} \rangle. \tag{4.6}$$

Then we have $A = (L_{\mathbf{X}}(x_i x_j))$ and $\text{cpsd-rank}_{\mathbb{C}}(A) = d = L_{\mathbf{X}}(1)$. To obtain lower bounds on $\text{cpsd-rank}_{\mathbb{C}}(A)$ we minimize $L(1)$ over a set of linear functionals L that satisfy certain computationally tractable properties of the above linear functional $L_{\mathbf{X}}$. Note that this idea of minimizing $L(1)$ over a suitable set of linear functionals has recently been used in the works [TS15, Nie17] in the commutative setting to derive a hierarchy of lower bounds converging to the nuclear norm of a symmetric tensor.

We now list some properties of the above linear functional $L_{\mathbf{X}}$. First, it is symmetric ($L_{\mathbf{X}}(p) = L_{\mathbf{X}}(p^*)$) and tracial ($L_{\mathbf{X}}(pq) = L_{\mathbf{X}}(qp)$). Moreover it satisfies some positivity conditions, since we have $L_{\mathbf{X}}(q) \geq 0$ whenever $q(\mathbf{X})$ is positive semidefinite. It follows that $L_{\mathbf{X}}(p^*p) \geq 0$ for all $p \in \mathbb{R}\langle \mathbf{x} \rangle$ and, as we explain later, $L_{\mathbf{X}}$ satisfies the *localizing* conditions $L_{\mathbf{X}}(p^*(\sqrt{A_{ii}}x_i - x_i^2)p) \geq 0$ for all $p \in \mathbb{R}\langle \mathbf{x} \rangle$ and $i \in [n]$. Restricting to truncated linear functionals acting on polynomials of bounded degree yields the following hierarchy of lower bounds on the cpsd-rank of A:

$$\begin{aligned} \xi_t^{\mathrm{cpsd}}(A) = \min\Big\{ L(1) : \; & L \in \mathbb{R}\langle x_1, \ldots, x_n \rangle_{2t}^* \text{ tracial and symmetric}, \\ & L(x_i x_j) = A_{ij} \quad \text{for} \quad i, j \in [n], \\ & L \geq 0 \quad \text{on} \quad \mathcal{M}_{2t}\big(\{\sqrt{A_{11}}x_1 - x_1^2, \ldots, \sqrt{A_{nn}}x_n - x_n^2\}\big) \Big\}. \end{aligned} \tag{5.1}$$

Recall that the quadratic module $\mathcal{M}_{2t}(S)$ of a set of symmetric polynomials S is defined in Equation (4.2). The bound $\xi_t^{\mathrm{cpsd}}(A)$ is computationally tractable (for small t). Indeed, as we explained in Section 4.1.1, the localizing constraint "$L \geq 0$ on $\mathcal{M}_{2t}(S)$" can be enforced by requiring certain matrices, whose entries are determined by L, to be positive semidefinite. This makes the problem defining $\xi_t^{\mathrm{cpsd}}(A)$ into a semidefinite program. The localizing conditions ensure the Archimedean property of the quadratic module, which permits to show certain convergence properties of the bounds $\xi_t^{\mathrm{cpsd}}(A)$.

The above approach extends naturally to the other matrix factorization ranks, using the following two basic ideas. First, since the cp-rank and the nonnegative rank deal with factorizations by *diagonal* matrices, we can use linear functionals acting on classical *commutative* polynomials. Second, the *asymmetric* factorization ranks (psd-rank and nonnegative rank) can be seen as analogs of the symmetric ranks in the *partial matrix* setting, where we know only the values of L on the quadratic monomials corresponding to entries in the off-diagonal blocks (this will require scaling of the factors in order to be able to define localizing constraints ensuring the Archimedean property). A main advantage of our approach is that it applies to all four matrix factorization ranks, after easy suitable adaptations.

Let us briefly explain the connection between trace evaluation functions at tuples of diagonal matrices and the more usual point evaluation maps considered in Equation (4.15). Consider a trace evaluation $L_{\mathbf{X}}$ (4.6) at a tuple of diagonal matrices $X = (\mathrm{diag}(v_1), \ldots, \mathrm{diag}(v_n))$, where $v_1, \ldots, v_n \in \mathbb{R}^d$. Then $L_{\mathbf{X}}$ is nothing else than the sum of d scalar evaluation maps:

$$L_{\mathbf{X}} = \sum_{k=1}^{d} L_{v^{(k)}}, \tag{5.2}$$

where $v^{(k)} = (v_1(k), \ldots, v_n(k)) \in \mathbb{R}^n$ for each $k \in [d]$, and the scalar evaluation maps $L_{v^{(k)}}$ are as defined in Equation (4.15), that is, $L_{v^{(k)}}(p) = p(v_1(k), \ldots, v_n(k))$ for all $p \in \mathbb{R}[\mathbf{x}]$. The vectors $v_1, \ldots, v_n \in \mathbb{R}^d$ and $v^{(1)}, \ldots, v^{(d)} \in \mathbb{R}^n$ are related as follows:

$$\big(\langle v_i, v_j\rangle\big)_{i,j\in[n]} = \sum_{k\in[d]} v^{(k)}(v^{(k)})^T.$$

In fact, this highlights the difference between two points of view on matrix factorizations: we can view a matrix either as a Gram matrix of vectors $v_1, \ldots, v_n$ or as a sum of rank-1 matrices $v^{(k)}(v^{(k)})^T$. The latter is called an 'atomic decomposition'; a rank-one matrix is called an 'atom'. Both the nonnegative rank and the completely positive rank have an atomic formulation. The correspondence between $L_{\mathbf{X}}$ and $\sum_{k\in[d]} L_{v^{(k)}}$ therefore explains why we can use techniques from commutative polynomial optimization to obtain lower bounds on the nonnegative rank and the completely positive rank. On the other hand, it is not clear how to use commutative polynomial optimization to obtain lower bounds on the psd-rank and cpsd-rank, since these factorization ranks are not known to have an 'atomic' formulation. As we explain in this chapter, noncommutative polynomial optimization offers the right framework to deal with general matrix factorizations.

5.1.1 Connection to polynomial optimization

In classical polynomial optimization the problem is to find the global minimum of a commutative polynomial f over a semialgebraic set. Tracial polynomial optimization is a noncommutative analog, where the problem is to minimize the normalized trace $\mathrm{tr}(f(\mathbf{X}))$ of a symmetric polynomial f where the tuple $\mathbf{X}$ lies in a matrix positivity domain $\mathcal{D}(S)$. Notice that the distinguishing feature here is the dimension independence: the optimization is over all possible matrix sizes. Perhaps counterintuitively, we use techniques similar to those used for the tracial polynomial optimization problem to compute lower bounds on factorization dimensions.

We briefly recall the hierarchies of semidefinite programming lower bounds for the above mentioned polynomial optimization problems; see Chapter 4 and in particular Section 4.2 for more details. For this chapter the moment formulation of the lower bounds is most relevant: For all $t \in \mathbb{N} \cup \{\infty\}$ we can define the bounds

$$\begin{aligned} f_t &= \inf\big\{L(f) : L \in \mathbb{R}[\mathbf{x}]_{2t}^*,\ L(1) = 1,\ L \geq 0 \text{ on } \mathcal{M}_{2t}(S)\big\},\\ f_t^{\mathrm{tr}}&= \inf\big\{L(f) : L \in \mathbb{R}\langle\mathbf{x}\rangle_{2t}^* \text{ tracial and symmetric},\ L(1) = 1,\ L \geq 0 \text{ on } \mathcal{M}_{2t}(S)\big\}, \end{aligned}$$

where f_t (resp., f_t^{tr}) lower bounds the (tracial) polynomial optimization problem.

The connection between the parameters $\xi_t^{\mathrm{cpsd}}(A)$ and f_t^{tr} is now clear: in the former we do not have the normalization property "$L(1) = 1$" but we do have the additional affine constraints "$L(x_ix_j) = A_{ij}$". This close relation to (tracial) polynomial optimization allows us to use that theory to understand the convergence properties of our bounds.

5.2 New and known results

Here we give an overview of the results in this chapter and we put them in perspective by explaining their relation to existing lower bounds.

5.2.1 Our results

For the nonnegative rank and the completely positive rank the best known generic lower bounds are due to Fawzi and Parrilo [FP15, FP16]. Based on the 'atomic' formulation of rank_+ and cp-rank they define the parameters $\tau_+(A)$ and $\tau_{\text{cp}}(A)$ in [FP16]. These parameters, respectively, lower bound the nonnegative rank and the cp-rank and are defined as follows:

$$\tau_+(A) = \min\Big\{\alpha : \alpha \geq 0,\, A \in \alpha \cdot \mathcal{R}_+\Big\},$$
$$\tau_{\text{cp}}(A) = \min\Big\{\alpha : \alpha \geq 0,\, A \in \alpha \cdot \mathcal{R}_{\text{cp}}\Big\},$$

where $\mathcal{R}_+$ and $\mathcal{R}_{\text{cp}}$ correspond to the convex hulls of the 'atoms' in the 'atomic formulation' of the respective factorization ranks:

$$\mathcal{R}_+ = \text{conv}\big\{R \in \mathbb{R}^{m\times n} : 0 \leq R \leq A,\, \text{rank}(R) \leq 1\big\},$$
$$\mathcal{R}_{\text{cp}} = \text{conv}\big\{R \in \mathrm{S}^n : 0 \leq R \leq A,\, R \preceq A,\, \text{rank}(R) \leq 1\big\}.$$

Note that τ_+ and τ_{cp} are the *Minkowski functionals* associated to the convex bodies $\mathcal{R}_+$ and $\mathcal{R}_{\text{cp}}$.

As the psd-rank and cpsd-rank are not known to admit atomic formulations, the techniques from [FP16] do not extend directly to these factorization ranks. As we have seen before in Equation (5.2), the atomic formulation corresponds to Gram factorizations by diagonal matrices. This suggests a way to obtain lower bounds on each of the four factorization ranks using (noncommutative) polynomial optimization. In particular, we show how the polynomial optimization perspective permits to obtain analogues of $\tau_+(\cdot)$ and $\tau_{\text{cp}}(\cdot)$ for the psd-rank and cpsd-rank. Namely, we will later introduce the parameters $\xi_*^{\text{psd}}(A)$ and $\xi_*^{\text{cpsd}}(A)$, which, respectively, lower bound psd-rank(A) and cpsd-rank(A). These bounds are defined as limiting objects (with an additional finiteness condition) of noncommutative polynomial optimization hierarchies. However, the results from Propositions 5.2 and 5.29 show that they can be interpreted as follows:

$$\xi_*^{\text{psd}}(A) = \inf\Big\{\alpha : \alpha \geq 0,\, A \in \alpha \cdot \mathcal{R}_{\text{psd}}\Big\},$$
$$\xi_*^{\text{cpsd}}(A) = \inf\Big\{\alpha : \alpha \geq 0,\, A \in \alpha \cdot \mathcal{R}_{\text{cpsd}}\Big\},$$

where

$$\begin{aligned}\mathcal{R}_{\rm psd} = \Big\{ \big(\mathrm{tr}(X_i Y_j)\big) \in \mathbb{R}^{m\times n} : \ & X_i \succeq 0 \ (i \in [m]), \ \textstyle\sum_{i=1}^m X_i = I, \\ & \textstyle\sum_{i=1}^m A_{ij} I \succeq Y_j \succeq 0 \ (j \in [n]) \quad \Big\}, \\ \mathcal{R}_{\rm cpsd} = \Big\{ \big(\mathrm{tr}(X_i X_j)\big) \in \mathbb{R}^{n\times n} : \ & \sqrt{A_{ii}} I \succeq X_i \succeq 0 \ (i \in [n]) \Big\}.\end{aligned}$$

The sets $\mathcal{R}_{\rm psd}$ and $\mathcal{R}_{\rm cpsd}$ can be viewed as noncommutative analogues of the sets $\mathcal{R}_+$ and $\mathcal{R}_{\rm cp}$.

The polynomial optimization perspective provides a hierarchy $\xi_t^{\mathtt{x}}(A)$ of semidefinite programming lower bounds for each of these factorization ranks, where $t \in \mathbb{N} \cup \{\infty, *\}$ and $\mathtt{x} \in \{+, \mathrm{cp}, \mathrm{psd}, \mathrm{cpsd}\}$, with the property that

$$\xi_t^{\mathtt{x}}(A) \leq \xi_{t+1}^{\mathtt{x}}(A) \leq \xi_\infty^{\mathtt{x}}(A) \leq \xi_*^{\mathtt{x}}(A) \quad \text{ for } t \in \mathbb{N},$$

and that $\xi_*^{\mathtt{x}}(A)$ is a lower bound on the respective factorization rank. The parameter $\xi_*^{\mathtt{x}}(A)$ can be obtained from the parameter $\xi_\infty^{\mathtt{x}}(A)$ by adding a finiteness condition on the rank of the associated moment matrix. We show that these hierarchies converge: $\xi_t^{\mathtt{x}}(A) \to \xi_\infty^{\mathtt{x}}(A)$ as $t \to \infty$, and that for the nonnegative rank we have $\xi_\infty^+(A) = \xi_*^+(A) = \tau_+(A)$. The basic hierarchy $\{\xi_t^{\rm cp}(A)\}$ for the cp-rank does not converge to $\tau_{\rm cp}(A)$ in general, but we provide two types of additional constraints that can be added to the program defining $\xi_t^{\rm cp}(A)$ to ensure convergence to $\tau_{\rm cp}(A)$. Therefore, our approach provides a computational scheme for approximating the parameters $\tau_+(\cdot)$ and $\tau_{\rm cp}(\cdot)$ considered in [FP16] (whereas they only provided a single semidefinite programming lower bound on these lower bounds). In Sections 5.4.5 and 5.5.2 we give some numerical examples where our bounds $\xi_t^{\mathtt{x}}(\cdot)$ improve on the SDP relaxations of $\tau_+(\cdot)$ and $\tau_{\rm cp}(\cdot)$ of [FP16]. Below we give a more detailed comparison between our bounds $\xi_t^{\mathtt{x}}(\cdot)$ and existing bounds.

5.2.2 Relation to existing bounds

Completely positive semidefinite rank. In the literature not much is known about lower bounds for the cpsd-rank. The inequality $\mathrm{rank}(A) \leq \text{cpsd-rank}_{\mathbb{C}}(A)^2$ is known (see Equation (2.4)), which follows by viewing a Hermitian $d \times d$ matrix as a d^2-dimensional real vector, and an analytic lower bound is given in [PSVW18]. In Section 5.3 we show that the new parameter $\xi_1^{\rm cpsd}(A)$ is at least as good as this analytic lower bound and we give a small example where a strengthening of $\xi_2^{\rm cpsd}(A)$ is strictly better than both above-mentioned generic lower bounds. Currently we lack evidence that the lower bounds $\xi_t^{\rm cpsd}(A)$ can be larger than, for example, the matrix size, but this could be because small matrices with large cpsd-rank are hard to construct (or might even not exist for $n < 10$). We also introduce several ideas leading to strengthenings of the basic bounds $\xi_t^{\rm cpsd}(A)$.

Nonnegative rank. In [FP16] it is shown that $\tau_+(A)$ is at least as good as certain norm-based lower bounds. In particular, $\tau_+(\cdot)$ is at least as good as the ℓ_∞-norm-based lower bound, which was used by Rothvoß [Rot17] to show that the matching polytope has exponential linear extension complexity. In [FP15] it is shown that for the Frobenius norm, the square of the norm-based bound is still a lower bound on the nonnegative rank, but it is not known how this lower bound compares to $\tau_+(\cdot)$. Fawzi and Parrilo [FP16] also defined an SDP relaxation $\tau_+^{\mathrm{sos}}(A)$ of $\tau_+(A)$. In Section 5.5 we show how a natural strengthening of our bound of order $t = 2$ is at least as strong as $\tau_+^{\mathrm{sos}}(A)$, and we give an example where this strengthening is strictly stronger for $t = 3$.

Completely positive rank. As we said before, the basic hierarchy $\{\xi_t^{\mathrm{cp}}(A)\}$ for the cp-rank does not converge to $\tau_{\mathrm{cp}}(A)$ in general, but we provide two types of additional constraints that can be added to the program defining $\xi_t^{\mathrm{cp}}(A)$ to ensure convergence to $\tau_{\mathrm{cp}}(A)$. First, we show how a generalization of the tensor constraints that are used in the definition of the parameter $\tau_{\mathrm{cp}}^{\mathrm{sos}}(A)$ can be used for this, and we also give a more efficient (using smaller matrix blocks) description of these constraints. This strengthening of $\xi_2^{\mathrm{cp}}(A)$ is then at least as strong as $\tau_{\mathrm{cp}}^{\mathrm{sos}}(A)$, but requires matrix variables of roughly half the size. Alternatively, we show that for every $\varepsilon > 0$ there is a finite number of additional linear constraints that can be added to the basic hierarchy $\{\xi_t^{\mathrm{cp}}(A)\}$ so that the limit of the sequence of these new lower bounds $\xi_t^{+}(A)$ is at least $\tau_{\mathrm{cp}}(A) - \varepsilon$. We give numerical results on small matrices studied in the literature, which show that $\xi_3^{+}(A)$ can improve over $\tau_+^{\mathrm{sos}}(A)$.

Positive semidefinite rank. In Section 5.6 we compare the new bounds $\xi_t^{\mathrm{psd}}(A)$ to a bound from [LWdW17] and we provide some numerical examples illustrating their performance.

5.3 The completely positive semidefinite rank

Let A be a completely positive semidefinite $n \times n$ matrix. Throughout we assume that $A_{ii} > 0$ for all $i \in [n]$.[1] For $t \in \mathbb{N} \cup \{\infty\}$ we consider the following semidefinite program, which, as we see below, lower bounds the complex completely positive semidefinite rank of A:

$$\begin{aligned}\xi_t^{\mathrm{cpsd}}(A) = \min\big\{L(1) : \; & L \in \mathbb{R}\langle x_1, \ldots, x_n\rangle_{2t}^* \text{ tracial and symmetric},\\ & L(x_ix_j) = A_{ij} \quad \text{for} \quad i,j \in [n],\\ & L \geq 0 \quad \text{on} \quad \mathcal{M}_{2t}(S_A^{\mathrm{cpsd}})\big\},\end{aligned}$$

where we set

$$S_A^{\mathrm{cpsd}} = \big\{\sqrt{A_{11}}x_1 - x_1^2, \ldots, \sqrt{A_{nn}}x_n - x_n^2\big\}. \tag{5.3}$$

Additionally, we define the parameter $\xi_*^{\mathrm{cpsd}}(A)$, obtained by adding the rank constraint $\mathrm{rank}(M(L)) < \infty$ to the program defining $\xi_\infty^{\mathrm{cpsd}}(A)$, where we consider the infimum instead of the minimum since we do not know whether the infimum is always attained. (In Proposition 5.1 we show the infimum is attained in $\xi_t^{\mathrm{cpsd}}(A)$

[1] We can do so without loss of generality: if $A_{ii} = 0$ for $i \in [n]$ then $A_{ij} = 0$ for all $j \in [n]$, and the cpsd-rank of A equals the cpsd-rank of the submatrix of A whose rows and columns are indexed by $[n] \setminus \{i\}$.

for $t \in \mathbb{N} \cup \{\infty\}$.) This gives a hierarchy of monotone nondecreasing lower bounds on the completely positive semidefinite rank:

$$\xi_1^{\text{cpsd}}(A) \le \ldots \le \xi_t^{\text{cpsd}}(A) \le \ldots \le \xi_\infty^{\text{cpsd}}(A) \le \xi_*^{\text{cpsd}}(A) \le \text{cpsd-rank}_{\mathbb{C}}(A).$$

The inequality $\xi_\infty^{\text{cpsd}}(A) \le \xi_*^{\text{cpsd}}(A)$ is clear, and monotonicity as well: If L is feasible for $\xi_k^{\text{cpsd}}(A)$ with $t \le k \le \infty$, then its restriction to $\mathbb{R}\langle \mathbf{x} \rangle_{2t}$ is feasible for $\xi_t^{\text{cpsd}}(A)$.

The following notion of *localizing* polynomials will be useful. A set $S \subseteq \mathbb{R}\langle \mathbf{x} \rangle$ is said to be *localizing* at a matrix tuple $\mathbf{X}$ if $\mathbf{X} \in \mathcal{D}(S)$ (i.e., $g(\mathbf{X}) \succeq 0$ for all polynomials $g \in S$) and we say that S is *localizing for A* if S is localizing at some factorization $\mathbf{X} \in (\mathrm{H}_+^d)^n$ of A with $d = \text{cpsd-rank}_{\mathbb{C}}(A)$. The set S_A^{cpsd} as defined in (5.3) is localizing for A, and, in fact, it is localizing at *any* factorization $\mathbf{X}$ of A by Hermitian positive semidefinite matrices. Indeed, since

$$A_{ii} = \mathrm{Tr}(X_i^2) \ge \lambda_{\max}(X_i^2) = \lambda_{\max}(X_i)^2$$

we have $\sqrt{A_{ii}} X_i - X_i^2 \succeq 0$ for all $i \in [n]$.

We can now use this to show the inequality $\xi_*^{\text{cpsd}}(A) \le \text{cpsd-rank}_{\mathbb{C}}(A)$. For this set $d = \text{cpsd-rank}_{\mathbb{C}}(A)$, let $\mathbf{X} \in (\mathrm{H}_+^d)^n$ be a Gram factorization of A, and consider the linear form $L_{\mathbf{X}} \in \mathbb{R}\langle \mathbf{x} \rangle^*$ defined by

$$L_{\mathbf{X}}(p) = \mathrm{Re}(\mathrm{Tr}(p(\mathbf{X}))) \quad \text{for all} \quad p \in \mathbb{R}\langle \mathbf{x} \rangle.$$

By construction $L_{\mathbf{X}}$ is symmetric and tracial, and we have $A = (L(x_i x_j))$. Moreover, since the set of polynomials S_A^{cpsd} is localizing for A, the linear form $L_{\mathbf{X}}$ is nonnegative on $\mathcal{M}(S_A^{\text{cpsd}})$. Finally, we have $\mathrm{rank}(M(L_{\mathbf{X}})) < \infty$, since the algebra generated by $X_1, \ldots, X_n$ is finite-dimensional. Hence, $L_{\mathbf{X}}$ is feasible for $\xi_*^{\text{cpsd}}(A)$ with $L_{\mathbf{X}}(1) = \mathrm{Re}(\mathrm{Tr}(I_d)) = d$, which shows $\xi_*^{\text{cpsd}}(A) \le \text{cpsd-rank}_{\mathbb{C}}(A)$.

The inclusions in (5.4) below show that the quadratic module $\mathcal{M}(S_A^{\text{cpsd}})$ is Archimedean (recall the definition in (4.4)). Moreover, although there are other possible choices for the localizing polynomials to use in S_A^{cpsd}, these inclusions also show that the choice made in (5.3) leads to the largest truncated quadratic module and thus to the best bound. For any scalar $c > 0$, we have the inclusions

$$\mathcal{M}_{2t}(x, c - x) \subseteq \mathcal{M}_{2t}(x, c^2 - x^2) \subseteq \mathcal{M}_{2t}(cx - x^2) \subseteq \mathcal{M}_{2t+2}(x, c - x), \tag{5.4}$$

which hold in light of the following identities:

$$c - x = \big((c - x)^2 + c^2 - x^2\big)/(2c), \tag{5.5}$$

$$c^2 - x^2 = (c - x)^2 + 2(cx - x^2), \tag{5.6}$$

$$cx - x^2 = \big((c - x)x(c - x) + x(c - x)x\big)/c, \tag{5.7}$$

$$x = \big((cx - x^2) + x^2\big)/c. \tag{5.8}$$

In the rest of this section we investigate properties of the hierarchy $\{\xi_t^{\text{cpsd}}(A)\}$ as well as some variations on it. We discuss convergence properties, asymptotically and under flatness, and we give another formulation for the parameter $\xi_*^{\text{cpsd}}(A)$. Moreover, as the inequality $\xi_*^{\text{cpsd}}(A) \le \text{cpsd-rank}_{\mathbb{C}}(A)$ is typically strict, we present an approach to strengthen the bounds in order to go beyond $\xi_*^{\text{cpsd}}(A)$. Then we propose some techniques to simplify the computation of the bounds, and we illustrate the behaviour of the bounds on some examples.

5.3.1 The parameters $\xi_\infty^{\mathrm{cpsd}}(A)$ and $\xi_*^{\mathrm{cpsd}}(A)$

In this section we consider convergence properties of the hierarchy $\xi_t^{\mathrm{cpsd}}(\cdot)$, both asymptotically and under flatness. We also give equivalent reformulations of the limiting parameters $\xi_\infty^{\mathrm{cpsd}}(A)$ and $\xi_*^{\mathrm{cpsd}}(A)$ in terms of C^*-algebras with a tracial state, which we will use in Sections 5.3.3 and 5.3.4 to show properties of these parameters.

Proposition 5.1. *Let $A \in \mathrm{CS}_+^n$. For $t \in \mathbb{N} \cup \{\infty\}$ the optimum in $\xi_t^{\mathrm{cpsd}}(A)$ is attained, and*

$$\lim_{t\to\infty} \xi_t^{\mathrm{cpsd}}(A) = \xi_\infty^{\mathrm{cpsd}}(A).$$

Moreover, $\xi_\infty^{\mathrm{cpsd}}(A)$ is equal to the smallest scalar $\alpha \geq 0$ for which there exists a unital C^*-algebra $\mathcal{A}$ with tracial state τ and $(X_1,\dots,X_n) \in \mathcal{D}_{\mathcal{A}}(S_A^{\mathrm{cpsd}})$ such that $A = \alpha \cdot (\tau(X_iX_j))$.

Proof. The sequence $(\xi_t^{\mathrm{cpsd}}(A))_t$ is monotonically nondecreasing and upper bounded by $\xi_\infty^{\mathrm{cpsd}}(A) < \infty$, which implies its limit exists and is at most $\xi_\infty^{\mathrm{cpsd}}(A)$.

As $\xi_t^{\mathrm{cpsd}}(A) \leq \xi_\infty^{\mathrm{cpsd}}(A)$, we may add the redundant constraint $L(1) \leq \xi_\infty^{\mathrm{cpsd}}(A)$ to the problem $\xi_t^{\mathrm{cpsd}}(A)$ for every $t \in \mathbb{N}$. By (5.6) we have $\mathrm{Tr}(A) - \sum_i x_i^2 \in \mathcal{M}_2(S_A^{\mathrm{cpsd}})$. Hence, using the result of Lemma 4.15, the feasible region of $\xi_t^{\mathrm{cpsd}}(A)$ is compact, and thus it has an optimal solution L_t. Again by Lemma 4.15, the sequence (L_t) has a pointwise converging subsequence with limit $L \in \mathbb{R}\langle \mathbf{x}\rangle^*$. This pointwise limit L is symmetric, tracial, satisfies $(L(x_ix_j)) = A$, and is nonnegative on $\mathcal{M}(S_A^{\mathrm{cpsd}})$. Hence L is feasible for $\xi_\infty^{\mathrm{cpsd}}(A)$. This implies that L is optimal for $\xi_\infty^{\mathrm{cpsd}}(A)$ and we have $\lim_{t\to\infty} \xi_t^{\mathrm{cpsd}}(A) = \xi_\infty^{\mathrm{cpsd}}(A)$.

The reformulation of $\xi_\infty^{\mathrm{cpsd}}(A)$ in terms of C^*-algebras with a tracial state follows directly using Theorem 4.5. □

Next we give an equivalent reformulation of the parameter $\xi_*^{\mathrm{cpsd}}(A)$, which follows as a direct application of Theorem 4.6. In general we do not know whether the infimum in $\xi_*^{\mathrm{cpsd}}(A)$ is attained. However, as an application of Corollary 4.8, we see that this infimum is attained if there is an integer $t \in \mathbb{N}$ for which $\xi_t^{\mathrm{cpsd}}(A)$ admits a flat optimal solution.

Proposition 5.2. *Let $A \in \mathrm{CS}_+^n$. The parameter $\xi_*^{\mathrm{cpsd}}(A)$ is given by the infimum of $L(1)$ taken over all conic combinations L of trace evaluations at elements in $\mathcal{D}_{\mathcal{A}}(S_A^{\mathrm{cpsd}})$ for which $A = (L(x_ix_j))$. The parameter $\xi_*^{\mathrm{cpsd}}(A)$ is also equal to the infimum over all $\alpha \geq 0$ for which there exist a finite-dimensional C^*-algebra $\mathcal{A}$ with tracial state τ and $(X_1,\dots,X_n) \in \mathcal{D}_{\mathcal{A}}(S_A^{\mathrm{cpsd}})$ such that $A = \alpha \cdot (\tau(X_iX_j))$.*

In addition, if $\xi_t^{\mathrm{cpsd}}(A)$ admits a flat optimal solution, then $\xi_t^{\mathrm{cpsd}}(A) = \xi_*^{\mathrm{cpsd}}(A)$.

We can strengthen the above when A lies on an extreme ray of the cone CS_+^n.

Proposition 5.3. *If A lies on an extreme ray of the cone CS_+^n, then*

$$\xi_*^{\mathrm{cpsd}}(A) = \inf\Big\{ d \cdot \max_{i\in[n]} \frac{\|X_i\|^2}{A_{ii}} : d \in \mathbb{N}, X_1,\dots,X_n \in \mathrm{H}_+^d,\ A = \mathrm{Gram}\big(X_1,\dots,X_n\big) \Big\}.$$

Moreover, if $X_1, \dots, X_n$ is a Gram decomposition of A providing an optimal solution to this reformulation of $\xi_*^{\text{cpsd}}(A)$ and some X_i has rank 1, then

$$\xi_*^{\text{cpsd}}(A) = \text{cpsd-rank}_{\mathbb{C}}(A).$$

Proof. Let γ be the value of the infimum on the right-hand side. First observe that $\xi_*^{\text{cpsd}}(A) \leq \gamma$ since for any $\mathbf{X} \in (\mathrm{H}_+^d)^n$ with $\mathrm{Gram}(\mathbf{X}) = A$ the linear functional $L(p) = \lambda \mathrm{Tr}(p(\mathbf{X}/\sqrt{\lambda}))$, where $\lambda = \max_{i\in[n]} \frac{\|X_i\|^2}{A_{ii}}$, is feasible for $\xi_*^{\text{cpsd}}(A)$ with objective value $\lambda \cdot d$.

For the reverse inequality, consider a feasible L to $\xi_*^{\text{cpsd}}(A)$. Then, in view of the first claim in Proposition 5.2, we have $L = \sum_j \alpha_j L^{(j)}$ for some $\alpha_j > 0$ and trace evaluations $L^{(j)}$ at elements $\mathbf{X}^{(j)} \in \mathcal{D}(S_A^{\text{cpsd}}) \cap (\mathrm{H}_+^{d_j})^n$ such that $L(1) = \sum_j \alpha_j d_j$. We may assume that the α_j's and $\mathbf{X}^{(j)}$'s are such that $\max_{i\in[n]} \frac{\|X_i^{(j)}\|^2}{A_{ii}} = 1$ for all j. Since A lies on an extreme ray of CS_+^n, for each j there exists a $\beta_j > 0$ such that $\beta_j A = \alpha_j \big(L^{(j)}(x_i x_k)\big)$. It follows that $A = \mathrm{Gram}(\sqrt{\alpha_j/\beta_j}\mathbf{X}^{(j)})$ and therefore $\gamma \leq d_j \cdot \frac{\alpha_j}{\beta_j}$ for all j. We then obtain the inequality

$$\gamma \leq \min_j \frac{\alpha_j d_j}{\beta_j} \leq \frac{\sum_j \alpha_j d_j}{\sum_j \beta_j} = \sum_j \alpha_j d_j = L(1),$$

where the first equality uses $\sum_j \beta_j = 1$.

Finally, assume that $X_1, \dots, X_n \in \mathrm{H}_+^d$ is a Gram decomposition of A with $d \cdot \max_{i\in[n]} \frac{\|X_i\|^2}{A_{ii}} = \xi_*^{\text{cpsd}}(A)$ and that one of the X_i has rank one. Then $\max_{i\in[n]} \frac{\|X_i\|^2}{A_{ii}} = 1$ since for a rank-one psd matrix X_i we have $\|X_i\|^2 = \mathrm{Tr}(X_i^2) = A_{ii}$. It follows that $d = \xi_*^{\text{cpsd}}(A) \leq \text{cpsd-rank}_{\mathbb{C}}(A) \leq d$ and thus $\xi_*^{\text{cpsd}}(A) = \text{cpsd-rank}_{\mathbb{C}}(A)$. □

5.3.2 Adding constraints to improve on $\xi_*^{\text{cpsd}}(A)$

In order to strengthen the bounds we may require nonnegativity over a (truncated) quadratic module generated by a larger set of localizing polynomials for A. The following lemma gives one such approach.

Lemma 5.4. *Let $A \in \mathrm{CS}_+^n$. For $v \in \mathbb{R}^n$ and $g_v = v^T A v - \big(\sum_{i=1}^n v_i x_i\big)^2$, the set $\{g_v\}$ is localizing for every Gram factorization by Hermitian positive semidefinite matrices of A (in particular, $\{g_v\}$ is localizing for A).*

Proof. If $X_1, \dots, X_n$ is a Gram decomposition of A by Hermitian positive semidefinite matrices, then

$$v^T A v = \mathrm{Tr}\,\Big(\Big(\sum_{i=1}^n v_i X_i\Big)^2\Big) \geq \lambda_{\max}\Big(\Big(\sum_{i=1}^n v_i X_i\Big)^2\Big),$$

hence $v^T A v I - (\sum_{i=1}^n v_i X_i)^2 \succeq 0$. □

Given a set $V \subseteq \mathbb{R}^n$, we consider the larger set

$$S_{A,V}^{\mathrm{cpsd}} = S_A^{\mathrm{cpsd}} \cup \{g_v : v \in V\}$$

of localizing polynomials for A. For $t \in \mathbb{N} \cup \{\infty, *\}$, denote by $\xi_{t,V}^{\mathrm{cpsd}}(A)$ the parameter obtained by replacing in $\xi_t^{\mathrm{cpsd}}(A)$ the nonnegativity constraint on $\mathcal{M}_{2t}(S_A^{\mathrm{cpsd}})$ by nonnegativity on the larger set $\mathcal{M}_{2t}(S_{A,V}^{\mathrm{cpsd}})$. We have $\xi_{t,\emptyset}^{\mathrm{cpsd}}(A) = \xi_t^{\mathrm{cpsd}}(A)$ and

$$\xi_t^{\mathrm{cpsd}}(A) \leq \xi_{t,V}^{\mathrm{cpsd}}(A) \leq \text{cpsd-rank}_{\mathbb{C}}(A) \quad \text{for all} \quad V \subseteq \mathbb{R}^n.$$

By scaling invariance, we can add the above constraints for all $v \in \mathbb{R}^n$ by setting V to be the unit sphere $\mathbb{S}^{n-1}$. Since $\mathbb{S}^{n-1}$ is a compact metric space, there exists a sequence $V_1 \subseteq V_2 \subseteq \ldots \subseteq \mathbb{S}^{n-1}$ of finite subsets such that $\bigcup_{k \geq 1} V_k$ is dense in $\mathbb{S}^{n-1}$. Each of the parameters $\xi_{t,V_k}^{\mathrm{cpsd}}(A)$ involves finitely many localizing constraints, and, as we now show, they converge to the parameter $\xi_{t,\mathbb{S}^{n-1}}^{\mathrm{cpsd}}(A)$.

Proposition 5.5. *Consider a matrix $A \in \mathrm{CS}_+^n$. For $t \in \{\infty, *\}$, we have*

$$\lim_{k \to \infty} \xi_{t,V_k}^{\mathrm{cpsd}}(A) = \xi_{t,\mathbb{S}^{n-1}}^{\mathrm{cpsd}}(A).$$

Proof. Let $\varepsilon > 0$. Since $\bigcup_k V_k$ is dense in $\mathbb{S}^{n-1}$, there is an integer $k \geq 1$ so that for every $u \in \mathbb{S}^{n-1}$ there exists a vector $v \in V_k$ satisfying

$$\|u - v\|_1 \leq \frac{\varepsilon \lambda_{\min}(A)}{4\sqrt{n} \max_i A_{ii}} \quad \text{and} \quad \|u - v\|_2 \leq \frac{\varepsilon \lambda_{\min}(A)}{4 \mathrm{Tr}(A^2)^{1/2}}. \tag{5.9}$$

The above Proposition 5.1 and Proposition 5.2 have natural analogues for the programs $\xi_{t,V}^{\mathrm{cpsd}}(A)$. These analogues show that for $t = \infty$ (resp. $t = *$) the parameter $\xi_{t,V_k}^{\mathrm{cpsd}}(A)$ is the infimum over all $\alpha \geq 0$ for which there exist a (resp. finite-dimensional) unital C^*-algebra $\mathcal{A}$ with tracial state τ and $\mathbf{X} \in \mathcal{D}_{\mathcal{A}}(S_{A,V_k}^{\mathrm{cpsd}})$ such that $A = \alpha \cdot (\tau(X_i X_j))$.

Below we will show that $\mathbf{X}' = \sqrt{1-\varepsilon}\mathbf{X} \in \mathcal{D}_{\mathcal{A}}(S_{A,\mathbb{S}^{n-1}}^{\mathrm{cpsd}})$. This implies that the linear form $L \in \mathbb{R}\langle \mathbf{x} \rangle^*$ defined by $L(p) = \alpha/(1-\varepsilon)\tau(p(\mathbf{X}'))$ is feasible for $\xi_{t,\mathbb{S}^{n-1}}^{\mathrm{cpsd}}(A)$ with objective value $L(1) = \alpha/(1-\varepsilon)$. This shows

$$\xi_{t,\mathbb{S}^{n-1}}^{\mathrm{cpsd}}(A) \leq \frac{1}{1-\varepsilon}\, \xi_{t,V_k}^{\mathrm{cpsd}}(A) \leq \frac{1}{1-\varepsilon} \lim_{k \to \infty} \xi_{t,V_k}^{\mathrm{cpsd}}(A).$$

Since $\varepsilon > 0$ was arbitrary, letting ε tend to 0 completes the proof.

We now show $\mathbf{X}' = \sqrt{1-\varepsilon}\mathbf{X} \in \mathcal{D}_{\mathcal{A}}(S_{A,\mathbb{S}^{n-1}}^{\mathrm{cpsd}})$. For this consider the map

$$f_{\mathbf{X}} \colon \mathbb{S}^{n-1} \to \mathbb{R},\, v \mapsto \Big\| \sum_{i=1}^n v_i X_i \Big\|^2,$$

where $\|\cdot\|$ denotes the C^*-algebra norm of $\mathcal{A}$. For $\alpha \in \mathbb{R}_+$ and $a \in \mathcal{A}$ with $a^* = a$, we have $\alpha \geq \|a\|$ if and only if $\alpha - a \succeq 0$ in $\mathcal{A}$, or, equivalently, $\alpha^2 - a^2 \succeq 0$ in $\mathcal{A}$. Since $\mathbf{X} \in \mathcal{D}_{\mathcal{A}}(S_{A,V_k}^{\mathrm{cpsd}})$ we have $v^T A v - f_{\mathbf{X}}(v) \geq 0$ for all $v \in V_k$, and hence

$$v^T A v - f_{\mathbf{X}'}(v) = v^T A v\Big(1 - (1-\varepsilon)\frac{f_{\mathbf{X}}(v)}{v^T A v}\Big) \geq v^T A v\big(1 - (1-\varepsilon)\big) = \varepsilon v^T A v \geq \varepsilon \lambda_{\min}(A).$$

Let $u \in \mathbb{S}^{n-1}$ and let $v \in V_k$ be such that (5.9) holds. Using the Cauchy-Schwarz inequality we have

$$\begin{aligned}|u^T Au - v^T Av| &= |(u-v)^T A(u+v)| = |\langle A, (u-v)(u+v)^T\rangle| \\ &\leq \sqrt{\mathrm{Tr}(A^2)}\sqrt{\mathrm{Tr}((u+v)(u-v)^T(u-v)(u+v)^T)} \\ &\leq \sqrt{\mathrm{Tr}(A^2)}\|u-v\|_2\|u+v\|_2 \leq 2\sqrt{\mathrm{Tr}(A^2)}\|u-v\|_2 \\ &\leq 2\sqrt{\mathrm{Tr}(A^2)}\frac{\varepsilon\lambda_{\min}(A)}{4\sqrt{\mathrm{Tr}(A^2)}} = \frac{\varepsilon\lambda_{\min}(A)}{2}.\end{aligned}$$

Since $\sqrt{A_{ii}}X_i - X_i^2$ is positive in $\mathcal{A}$, we have that $\sqrt{A_{ii}} - X_i$ is positive in $\mathcal{A}$ by (5.5) and (5.6), which implies $\|X_i\| \leq \sqrt{A_{ii}}$. By the reverse triangle inequality we then have

$$\begin{aligned}|f_{\mathbf{X}'}(u) - f_{\mathbf{X}'}(v)| &= \Big|\|\sum_{i=1}^n u_i X_i'\| - \|\sum_{i=1}^n v_i X_i'\|\Big|\Big(\|\sum_{i=1}^n u_i X_i'\| + \|\sum_{i=1}^n v_i X_i'\|\Big) \\ &\leq \|\sum_{i=1}^n (v_i - u_i)X_i'\| 2\sqrt{n}\,\max_i\sqrt{A_{ii}} \\ &\leq \Big(\sum_{i=1}^n |v_i - u_i|\|X_i'\|\Big)2\sqrt{n}\,\max_i\sqrt{A_{ii}} \\ &\leq \|u-v\|_1 2\sqrt{n}\,\max_i A_{ii} \\ &\leq \frac{\varepsilon\lambda_{\min}(A)}{4\sqrt{n}\,\max_i A_{ii}}2\sqrt{n}\,\max_i A_{ii} = \frac{\varepsilon\lambda_{\min}(A)}{2}.\end{aligned}$$

Combining the above inequalities we obtain that $u^T Au - f_{\mathbf{X}'}(u) \geq 0$ for all $\mathbb{S}^{n-1}$, and hence $u^T Au - \big(\sum_{i=1}^n u_i X_i'\big)^2$ is positive in $\mathcal{A}$. Thus we have $\mathbf{X}' \in \mathcal{D}_{\mathcal{A}}(S^{\mathrm{cpsd}}_{A,\mathbb{S}^{n-1}})$. □

We now discuss two examples where the bounds $\xi^{\mathrm{cpsd}}_{*,V}(A)$ go beyond $\xi^{\mathrm{cpsd}}_{*}(A)$.

Example 5.6. Consider the matrix

$$A = \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1\end{pmatrix} = \mathrm{Gram}\left(\begin{pmatrix} 1 & 0 \\ 0 & 0\end{pmatrix}, \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2\end{pmatrix}\right), \tag{5.10}$$

with cpsd-rank$_{\mathbb{C}}(A) = 2$. We can also write $A = \mathrm{Gram}(Y_1, Y_2)$, where

$$Y_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0\end{pmatrix}, \quad Y_2 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1\end{pmatrix}.$$

With $X_i = \sqrt{2}\,Y_i$ we have $I - X_i^2 \succeq 0$ for $i = 1, 2$. Hence the linear form $L = L_{\mathbf{X}}/2$ is feasible for $\xi^{\mathrm{cpsd}}_{*}(A)$, which shows that $\xi^{\mathrm{cpsd}}_{*}(A) \leq L(1) = 3/2$. In fact, this form L gives an flat solution to $\xi^{\mathrm{cpsd}}_{2}(A)$, and as we can check using a semidefinite programming solver it is also optimal, so $\xi^{\mathrm{cpsd}}_{*}(A) = 3/2$. In passing, we observe that $\xi_1^{\mathrm{cpsd}}(A) = 4/3$, which coincides with the analytic lower bound (5.11) (see also Lemma 5.11 below).

For $e = (1,1) \in \mathbb{R}^2$ and $V = \{e\}$, this form L is not feasible for $\xi_{*,V}^{\mathrm{cpsd}}(A)$, because for the polynomial $p = 1 - 3x_1 - 3x_2$ we have $L(p^* g_e p) = -9/2 < 0$. This means that the localizing constraint $L(p^* g_e p) \geq 0$ is not redundant: For $t \geq 2$ it cuts off part of the feasibility region of $\xi_t^{\mathrm{cpsd}}(A)$. Indeed, using a semidefinite programming solver we find an optimal flat solution of $\xi_{3,V}^{\mathrm{cpsd}}(A)$ with objective value approximately 1.633, hence

$$\xi_{*,V}^{\mathrm{cpsd}}(A) \approx 1.633 > 3/2 = \xi_*^{\mathrm{cpsd}}(A). \qquad \triangle$$

Example 5.7. Consider the symmetric circulant matrices

$$M(\alpha) = \begin{pmatrix} 1 & \alpha & 0 & 0 & \alpha \\ \alpha & 1 & \alpha & 0 & 0 \\ 0 & \alpha & 1 & \alpha & 0 \\ 0 & 0 & \alpha & 1 & \alpha \\ \alpha & 0 & 0 & \alpha & 1 \end{pmatrix} \quad \text{for} \quad \alpha \in \mathbb{R}_+.$$

For $0 \leq \alpha \leq 1/2$ we have $M(\alpha) \in \mathrm{CS}_+^5$ with $\text{cpsd-rank}_{\mathbb{C}}(M(\alpha)) \leq 5$. To see this we set $\beta = (1 + \sqrt{1-4\alpha^2})/2$, so that $\sqrt{\beta}\sqrt{1-\beta} = \alpha$, and observe that the matrices

$$X_i = \mathrm{Diag}(\sqrt{\beta}\, e_i + \sqrt{1-\beta}\, e_{i+1}) \in \mathrm{S}_+^5, \quad i \in [5], \quad (\text{with } e_6 := e_1),$$

provide a factorization of $M(\alpha)$.

Notice that for $\alpha = 1/2$ we have that $\beta = 1/2 = 1 - \beta$. That is,

$$M(1/2) = \frac{1}{2}\,\mathrm{Gram}(\{\mathrm{Diag}(e_i + e_{i+1}) : i \in [5], e_6 := e_1\}).$$

The tuple $(\mathrm{Diag}(e_1 + e_2), \ldots, \mathrm{Diag}(e_5 + e_1))$ belongs to $\mathcal{D}(S_{M(1/2)}^{\mathrm{cpsd}})$, and therefore, using the second part of Proposition 5.2, we find that $\xi_*^{\mathrm{cpsd}}(M(1/2)) \leq 5/2$. For $\alpha = 1/2$ this gives an upper bound of 5/2 on the value of $\xi_t^{\mathrm{cpsd}}(M(1/2))$ for all t. However, using a semidefinite programming solver we see that

$$\xi_{2,V}^{\mathrm{cpsd}}(M(1/2)) = 5,$$

where V is the set containing the vector $(1,-1,1,-1,1)$ and its cyclic shifts. Hence the bound $\xi_{2,V}^{\mathrm{cpsd}}(M(1/2))$ is tight: It certifies $\text{cpsd-rank}_{\mathbb{C}}(M(1/2)) = 5$, while the other two known bounds (the rank bound $\sqrt{\mathrm{rank}(A)}$ and the analytic bound (5.11)) only give $\text{cpsd-rank}_{\mathbb{C}}(A) \geq 3$.

We now observe that there exist $0 < \varepsilon, \delta < 1/2$ such that $\text{cpsd-rank}_{\mathbb{C}}(M(\alpha)) = 5$ for all $\alpha \in [0,\varepsilon] \cup [\delta, 1/2]$. Indeed, this follows from the fact that $\xi_1^{\mathrm{cpsd}}(M(0)) = 5$ (by Lemma 5.11), the above result that $\xi_{2,V}^{\mathrm{cpsd}}(M(1/2)) = 5$, and the lower semicontinuity of $\alpha \mapsto \xi_{2,V}^{\mathrm{cpsd}}(M(\alpha))$, which is shown in Lemma 5.12 below.

As the matrices $M(\alpha)$ are nonsingular, the above factorization shows that their cp-rank is equal to 5 for all $\alpha \in [0, 1/2]$; whether they all have cpsd-rank equal to 5 is not known. $\triangle$

5.3.3 Boosting the bounds

In this section we propose some additional constraints that can be added in order to strengthen the bounds $\xi_{t,V}^{\mathrm{cpsd}}(A)$. These constraints may shrink the feasible region of $\xi_{t,V}^{\mathrm{cpsd}}(A)$ for $t \in \mathbb{N}$, but they are redundant for $t \in \{\infty, *\}$. The latter is shown using the reformulation of the parameters $\xi_{\infty,V}^{\mathrm{cpsd}}(A)$ and $\xi_{*,V}^{\mathrm{cpsd}}(A)$ in terms of C^*-algebras.

We first mention how to construct localizing constraints of "bilinear type", inspired by the work of Berta, Fawzi and Scholz [BFS16], see Lemma 5.8 part 1. As for localizing constraints, these bilinear constraints can be modeled as semidefinite constraints. Second, we show how to use zero entries in A and vectors in the kernel of A to enforce new constraints on $\xi_{t,V}^{\mathrm{cpsd}}(A)$, see Lemma 5.8 part 2.

Lemma 5.8. *Let $A \in \mathrm{CS}_+^n$ and $t \in \mathbb{N} \cup \{\infty, *\}$. Then the following types of constraints can be used to strengthen $\xi_{t,V}^{\mathrm{cpsd}}(A)$ while still obtaining a lower bound on* $\text{cpsd-rank}_{\mathbb{C}}(A)$*:*

1. *$L(p^*gpg') \geq 0$ for all g, g' localizing for A and $p \in \mathbb{R}\langle \mathbf{x} \rangle$ with $\deg(p^*gpg') \leq 2t$.*
2. *$L = 0$ on $\mathcal{I}_{2t}\big(\big\{\sum_{i=1}^n v_i x_i : v \in \ker A\big\} \cup \big\{x_i x_j : A_{ij} = 0\big\}\big)$.*

Moreover, the first type of constraints is redundant for the programs $\xi_{\infty}^{\mathrm{cpsd}}(A)$ and $\xi_{*}^{\mathrm{cpsd}}(A)$ when $g, g' \in \mathcal{M}(S_A^{\mathrm{cpsd}})$, and the second type of constraints is always redundant for $\xi_{\infty}^{\mathrm{cpsd}}(A)$ and $\xi_{*}^{\mathrm{cpsd}}(A)$.

Proof. Let $\mathbf{X} \in (\mathrm{H}_+^d)^n$ be a Gram decomposition of A, and let $L = L_{\mathbf{X}}$ be the real part of the trace evaluation at $\mathbf{X}$. Then, since $p(\mathbf{X})^* g(\mathbf{X}) p(\mathbf{X}) \succeq 0$ and $g'(\mathbf{X}) \succeq 0$, we have $L(p^*gpg') = \mathrm{Re}(\mathrm{Tr}(p(\mathbf{X})^* g(\mathbf{X}) p(\mathbf{X}) g'(\mathbf{X}))) \geq 0$. Moreover, from $0 = v^T A v = \mathrm{Tr}\left(\left(\sum_{i=1}^n v_i X_i\right)^2\right)$ and $0 = A_{ij} = \mathrm{Tr}(X_i X_j)$ it follows that $\sum_{i=1}^n v_i X_i = 0$ and $X_i X_j = 0$. Therefore $L = 0$ on $\mathcal{I}_{2t}\big(\big\{\sum_{i=1}^n v_i x_i : v \in \ker A\big\} \cup \big\{x_i x_j : A_{ij} = 0\big\}\big)$.

Now suppose that $t \in \{\infty, *\}$ and let L be feasible for $\xi_t^{\mathrm{cpsd}}(A)$. Then, by Theorem 4.5 there exists a unital C^*-algebra $\mathcal{A}$ with tracial state τ and an element $\mathbf{X} \in \mathcal{D}(S_{A,V}^{\mathrm{cpsd}})$ such that $L(p) = L(1)\tau(p(\mathbf{X}))$ for all $p \in \mathbb{R}\langle \mathbf{x} \rangle$. Then, for polynomials $g, g' \in \mathcal{M}(S_{A,V}^{\mathrm{cpsd}})$ we know that $g(\mathbf{X}), g'(\mathbf{X})$ are positive elements in $\mathcal{A}$. So $g(\mathbf{X}) = a^*a$ and $g'(\mathbf{X}) = b^*b$ for some $a, b \in \mathcal{A}$. Then we have

$$\begin{aligned} L(p^*gpg) &= L(1)\, \tau(p^*(\mathbf{X})\, g(\mathbf{X})\, p(\mathbf{X})\, g'(\mathbf{X})) \\ &= L(1)\, \tau(p^*(\mathbf{X})\, a^* a\, p(\mathbf{X})\, b^* b) \\ &= L(1)\, \tau((a\, p(\mathbf{X})\, b^*)^* a\, p(\mathbf{X})\, b^*) \geq 0, \end{aligned}$$

where we use that τ is a positive tracial state on $\mathcal{A}$. To show that $L = 0$ on the ideal mentioned in the lemma we will use that we may assume that τ is faithful (see [GdLL19, Lem. 12]). For a vector v in the kernel of A we have $0 = v^T A v = L((\sum_i v_i x_i)^2) = L(1)\tau((\sum_i v_i X_i)^2)$, and hence, since τ is faithful, $\sum_i v_i X_i = 0$ in $\mathcal{A}$. Analogously, if $A_{ij} = 0$, then $L(x_i x_j) = 0$ implies $\tau(X_i X_j) = 0$ and thus $X_i X_j = 0$, since X_i, X_j are positive in $\mathcal{A}$ and τ is faithful and tracial. Together this implies that $L = 0$ on $\mathcal{I}_{2t}\big(\big\{\sum_{i=1}^n v_i x_i : v \in \ker A\big\} \cup \big\{x_i x_j : A_{ij} = 0\big\}\big)$. □

Note that for $v \in \ker(A)$ and $p \in \mathbb{R}\langle \mathbf{x} \rangle_t$ the constraints $L(p\,(\sum_{i=1}^n v_i x_i)) = 0$ are in fact redundant: if $v \in \ker(A)$, then the vector obtained by extending v with zeros belongs to $\ker(M_t(L))$, since $M_t(L) \succeq 0$, from which it follows that $L(p\,(\sum_{i=1}^n v_i x_i)) = 0$. Also, for an implementation of $\xi_t^{\mathrm{cpsd}}(A)$ with the additional constraints of Lemma 5.8, it is more efficient to index the moment matrices with a basis for $\mathbb{R}\langle \mathbf{x} \rangle_t$ modulo the ideal $\mathcal{I}_t\big(\{\sum_i v_i x_i : v \in \ker(A)\} \cup \{x_i x_j : A_{ij} = 0\}\big)$.

5.3.4 Additional properties of the bounds

Here we list some additional properties of the parameters $\xi_t^{\mathrm{cpsd}}(A)$ for $t \in \mathbb{N} \cup \{\infty, *\}$. First we state some properties for which the proofs are immediate and thus omitted.

Lemma 5.9. *Suppose $A \in \mathrm{CS}_+^n$ and $t \in \mathbb{N} \cup \{\infty, *\}$.*

(1) If P is a permutation matrix, then $\xi_t^{\mathrm{cpsd}}(A) = \xi_t^{\mathrm{cpsd}}(P^T A P)$.

(2) If B is a principal submatrix of A, then $\xi_t^{\mathrm{cpsd}}(B) \leq \xi_t^{\mathrm{cpsd}}(A)$.

(3) If D is a positive definite diagonal matrix, then $\xi_t^{\mathrm{cpsd}}(A) = \xi_t^{\mathrm{cpsd}}(DAD)$.

We also have the following direct sum property, where the equality follows using the C^*-algebra reformulations as given in Proposition 5.1 and Proposition 5.2.

Lemma 5.10. *If $A \in \mathrm{CS}_+^n$ and $B \in \mathrm{CS}_+^m$, then $\xi_t^{\mathrm{cpsd}}(A \oplus B) \leq \xi_t^{\mathrm{cpsd}}(A) + \xi_t^{\mathrm{cpsd}}(B)$, where equality holds for $t \in \{\infty, *\}$.*

Proof. To prove the inequality we take L_A and L_B feasible for $\xi_t^{\mathrm{cpsd}}(A)$ and $\xi_t^{\mathrm{cpsd}}(B)$, and construct a feasible L for $\xi_t^{\mathrm{cpsd}}(A \oplus B)$ by setting

$$L(p(\mathbf{x}, \mathbf{y})) = L_A(p(\mathbf{x}, \mathbf{0})) + L_B(p(\mathbf{0}, \mathbf{y})).$$

Now we show equality for $t = \infty$. By Proposition 5.1, $\xi_t^{\mathrm{cpsd}}(A \oplus B)$ is equal to the infimum over all $\alpha \geq 0$ for which there exists a unital C^*-algebra $\mathcal{A}$ with tracial state τ and $(\mathbf{X}, \mathbf{Y}) \in \mathcal{D}_{\mathcal{A}}(S_{A \oplus B}^{\mathrm{cpsd}})$ such that $A = \alpha \cdot (\tau(X_i X_j))$, $B = \alpha \cdot (\tau(Y_i Y_j))$ and $(\tau(X_i Y_j)) = 0$. This implies $\mathbf{X} \in \mathcal{D}_{\mathcal{A}}(S_A^{\mathrm{cpsd}})$ and $\mathbf{Y} \in \mathcal{D}_{\mathcal{A}}(S_B^{\mathrm{cpsd}})$. Let P_A be the projection onto the space $\sum_i \mathrm{Im}(X_i)$ and define the linear form $L_A \in \mathbb{R}\langle \mathbf{x} \rangle^*$ by $L_A(p) = \alpha \cdot \tau(p(\mathbf{X}) P_A)$. It follows that L_A is nonnegative on $\mathcal{M}(S_A^{\mathrm{cpsd}})$, and

$$L_A(x_i x_j) = \alpha\, \tau(x_i x_j P_A) = \alpha\, \tau(x_i x_j) = A_{ij},$$

so L_A is feasible for $\xi_\infty^{\mathrm{cpsd}}(A)$ with $L_A(1) = \alpha \tau(P_A)$. In the same way we consider the projection P_B onto the space $\sum_j \mathrm{Im}(Y_j)$ and define a feasible solution L_B for $\xi_t^{\mathrm{cpsd}}(B)$ with $L_B(1) = \alpha \tau(P_B)$. One can show that we may assume τ to be faithful (see [GdLL19, Lem. 12]), so that positivity of X_i and Y_j together with $\tau(X_i Y_j) = 0$ implies $X_i Y_j = 0$ for all i and j, and thus $\sum_i \mathrm{Im}(X_i) \perp \sum_j \mathrm{Im}(Y_j)$. This implies $I \succeq P_A + P_B$ and thus $\tau(P_A + P_B) \leq \tau(1) = 1$. We have

$$L_A(1) + L_B(1) = \alpha\, \tau(P_A) + \alpha \tau(P_B) \leq \alpha\, \tau(1) = \alpha,$$

so $\xi_\infty^{\mathrm{cpsd}}(A) + \xi_\infty^{\mathrm{cpsd}}(B) \leq L_A(1) + L_B(1) \leq \alpha$. The case $t = *$ follows similarly, now using Proposition 5.2 instead of Proposition 5.1, completing the proof. □

Note that the cpsd-rank of a matrix satisfies the same properties as those mentioned in the above two lemmas, where the inequality in Lemma 5.10 is always an equality: $\text{cpsd-rank}_{\mathbb{C}}(A \oplus B) = \text{cpsd-rank}_{\mathbb{C}}(A) + \text{cpsd-rank}_{\mathbb{C}}(B)$ [PSVW18, GdLL17].

The following lemma shows that the first level of our hierarchy is at least as good as the analytic lower bound (5.11) on the cpsd-rank derived in [PSVW18, Thm. 10].

Lemma 5.11. *For any non-zero matrix $A \in \mathrm{CS}_+^n$ we have*

$$\xi_1^{\mathrm{cpsd}}(A) \geq \frac{\left(\sum_{i=1}^n \sqrt{A_{ii}}\right)^2}{\sum_{i,j=1}^n A_{ij}}. \tag{5.11}$$

Proof. Let L be feasible for $\xi_1^{\mathrm{cpsd}}(A)$. Since L is nonnegative on $\mathcal{M}_2(S_A^{\mathrm{cpsd}})$, it follows that $L(\sqrt{A_{ii}}x_i - x_i^2) \geq 0$, implying $\sqrt{A_{ii}}L(x_i) \geq L(x_i^2) = A_{ii}$ and thus $L(x_i) \geq \sqrt{A_{ii}}$. Moreover, the matrix $M_1(L)$ is positive semidefinite. By taking the Schur complement with respect to its upper left corner (indexed by 1) it follows that the matrix $L(1) \cdot A - (L(x_i)L(x_j))$ is positive semidefinite. Hence the sum of its entries is nonnegative, which gives $L(1)(\sum_{i,j} A_{ij}) \geq (\sum_i L(x_i))^2 \geq (\sum_i \sqrt{A_{ii}})^2$ and shows the desired inequality. □

As an application of Lemma 5.11, the first bound ξ_1^{cpsd} is exact for the $k \times k$ identity matrix: $\xi_1^{\mathrm{cpsd}}(I_k) = \text{cpsd-rank}_{\mathbb{C}}(I_k) = k$. Moreover, by combining this with Lemma 5.9, it follows that $\xi_1^{\mathrm{cpsd}}(A) \geq k$ if A contains a diagonal positive definite $k \times k$ principal submatrix. A slightly more involved example is given by the 5×5 circulant matrix A whose entries are given by $A_{ij} = (\cos((i-j)4\pi/5))^2$ $(i, j \in [5])$; this matrix was used in [FGP+15] to show a separation between the completely positive semidefinite cone and the completely positive cone, and it was shown that $\text{cpsd-rank}_{\mathbb{C}}(A) = 2$. The analytic lower bound of [PSVW18] also evaluates to 2, hence Lemma 5.11 shows that our bound is tight on this example.

We now examine further analytic properties of the parameters $\xi_t^{\mathrm{cpsd}}(\cdot)$. For each $r \in \mathbb{N}$, the set of matrices $A \in \mathrm{CS}_+^n$ with $\text{cpsd-rank}_{\mathbb{C}}(A) \leq r$ is closed, which shows that the function $A \mapsto \text{cpsd-rank}_{\mathbb{C}}(A)$ is lower semicontinuous. We now show that the functions $A \mapsto \xi_t^{\mathrm{cpsd}}(A)$ have the same property. The other bounds defined in the following sections are also lower semicontinuous, with a similar proof.

Lemma 5.12. *For every $t \in \mathbb{N} \cup \{\infty\}$ and $V \subseteq \mathbb{R}^n$, the function*

$$\mathrm{S}^n \to \mathbb{R} \cup \{\infty\},\ A \mapsto \xi_{t,V}^{\mathrm{cpsd}}(A)$$

is lower semicontinuous.

Proof. It suffices to show the result for $t \in \mathbb{N}$, because $\xi_{\infty,V}^{\mathrm{cpsd}}(A) = \sup_t \xi_{t,V}^{\mathrm{cpsd}}(A)$, and the pointwise supremum of lower semicontinuous functions is lower semicontinuous. We show that the level sets $\{A \in \mathrm{S}^n : \xi_{t,V}^{\mathrm{cpsd}}(A) \leq r\}$ are closed. For this we consider a sequence $(A_k)_{k \in \mathbb{N}}$ in S^n converging to $A \in \mathrm{S}^n$ such that $\xi_{t,V}^{\mathrm{cpsd}}(A_k) \leq r$ for all k. We show that $\xi_{t,V}^{\mathrm{cpsd}}(A) \leq r$. Let $L_k \in \mathbb{R}\langle \mathbf{x} \rangle_{2t}^*$ be an optimal solution to $\xi_{t,V}^{\mathrm{cpsd}}(A_k)$. As $L_k(1) \leq r$ for all k, it follows from Lemma 4.15 that there is a pointwise converging subsequence of $(L_k)_k$, still denoted $(L_k)_k$ for simplicity, that has a limit $L \in \mathbb{R}\langle \mathbf{x}\rangle_{2t}^*$ with $L(1) \leq r$. To complete the proof we show that L is feasible for $\xi_{t,V}^{\mathrm{cpsd}}(A)$. By the pointwise convergence of L_k to L, for every $\varepsilon > 0$, $p \in \mathbb{R}\langle \mathbf{x}\rangle$, and $i \in [n]$, there exists a $K \in \mathbb{N}$ such that for all $k \geq K$ we have

$$|L(p^* x_i p) - L_k(p^* x_i p)| < \min\{1, \frac{\varepsilon}{\sqrt{A_{ii}}}\}, \qquad |L(p^* x_i^2 p) - L_k(p^* x_i^2 p)| < \varepsilon,$$
$$|\sqrt{A_{ii}} - \sqrt{(A_k)_{ii}}| < \frac{\varepsilon}{L(p^* x_i p) + 1}.$$

Hence we have

$$\begin{aligned} L(p^*(\sqrt{A_{ii}}x_i - x_i^2)p) &= \sqrt{A_{ii}}\Big(L(p^* x_i p) - L_k(p^* x_i p) + L_k(p^* x_i p)\Big) \\ &\quad - \Big(L(p^* x_i^2 p) - L_k(p^* x_i^2 p) + L_k(p^* x_i^2 p)\Big) \\ &\geq -2\varepsilon + \sqrt{A_{ii}}\, L_k(p^* x_i p) - L_k(p^* x_i^2 p) \\ &\geq -3\varepsilon + \sqrt{(A_k)_{ii}}\, L_k(p^* x_i p) - L_k(p^* x_i^2 p) \\ &= -3\varepsilon + L_k(p^*(\sqrt{(A_k)_{ii}}\, x_i - x_i^2)p) \geq -3\varepsilon, \end{aligned}$$

where in the second inequality we use that $0 \leq L_k(p^* x_i p) \leq L(p^* x_i p) + 1$. Letting $\varepsilon \to 0$ gives $L(p^*(\sqrt{A_{ii}}x_i - x_i^2)p) \geq 0$.

Similarly one can show $L(p^*(v^T A v - (\sum_i v_i x_i)^2)p) \geq 0$ for $v \in V$, $p \in \mathbb{R}\langle \mathbf{x}\rangle$. □

If we restrict to completely positive semidefinite matrices with an all-ones diagonal, that is, to $\mathrm{CS}_+^n \cap \mathcal{E}_n$, we can show an even stronger property. Here $\mathcal{E}_n$ is the *elliptope*, which is the set of $n \times n$ positive semidefinite matrices with an all-ones diagonal.

Lemma 5.13. *For every $t \in \mathbb{N} \cup \{\infty\}$, the function*

$$\mathrm{CS}_+^n \cap \mathcal{E}_n \to \mathbb{R},\ A \mapsto \xi_t^{\mathrm{cpsd}}(A)$$

is convex, and hence continuous on the interior of its domain.

Proof. Let $A, B \in \mathrm{CS}_+^n \cap \mathcal{E}_n$ and $0 < \lambda < 1$. Let L_A and L_B be optimal solutions for $\xi_t^{\mathrm{cpsd}}(A)$ and $\xi_t^{\mathrm{cpsd}}(B)$. Since the diagonals of A and B are the same, we have $S_A^{\mathrm{cpsd}} = S_B^{\mathrm{cpsd}}$. So $L = \lambda L_A + (1-\lambda)L_B$ is feasible for $\xi_t^{\mathrm{cpsd}}(\lambda A + (1-\lambda)B)$, hence $\xi_t^{\mathrm{cpsd}}(\lambda A + (1-\lambda)B) \leq \lambda L_A(1) + (1-\lambda)L_B(1) = \lambda \xi_t^{\mathrm{cpsd}}(A) + (1-\lambda)\xi_t^{\mathrm{cpsd}}(B)$. □

Example 5.14. Here we show that for $t \geq 1$, the function

$$\mathrm{CS}_+^n \to \mathbb{R},\ A \mapsto \xi_t^{\mathrm{cpsd}}(A)$$

is not continuous. For this we consider the matrices

$$A_k = \begin{pmatrix} 1/k & 0 \\ 0 & 1 \end{pmatrix} \in \mathrm{CS}_+^2,$$

with $\text{cpsd-rank}_{\mathbb{C}}(A_k) = 2$ for all $k \geq 1$. As A_k is diagonal positive definite we have $\xi_t^{\mathrm{cpsd}}(A_k) = 2$ for all $t, k \geq 1$, while $\xi_t^{\mathrm{cpsd}}(\lim_{k\to\infty} A_k) = 1$. This argument extends to CS_+^n with $n > 2$. This example also shows that the first level of the hierarchy $\xi_1^{\mathrm{cpsd}}(\cdot)$ can be strictly better than the analytic lower bound (5.11) of [PSVW18]. △

Example 5.15. In this example we determine $\xi_t^{\mathrm{cpsd}}(A)$ for all $t \geq 1$ and $A \in \mathrm{CS}_+^2$. In view of Lemma 5.9(3) we only need to find $\xi_t^{\mathrm{cpsd}}(A(\alpha))$ for $0 \leq \alpha \leq 1$, where $A(\alpha) = \left(\begin{smallmatrix} 1 & \alpha \\ \alpha & 1 \end{smallmatrix}\right)$.

The first bound $\xi_1^{\mathrm{cpsd}}(A(\alpha))$ is equal to the analytic bound $2/(\alpha+1)$ from Equation (5.11), where the equality follows from the fact that the truncated linear functional L given by $L(x_i x_j) = A(\alpha)_{ij}$, $L(x_1) = L(x_2) = 1$ and $L(1) = 2/(\alpha+1)$ is feasible for $\xi_1^{\mathrm{cpsd}}(A(\alpha))$.

For $t \geq 2$ we show that $\xi_t^{\mathrm{cpsd}}(A(\alpha)) = 2 - \alpha$. By the above this is true for $\alpha = 0$ and $\alpha = 1$, and in Example 5.6 we show $\xi_t^{\mathrm{cpsd}}(A(1/2)) = 3/2$ for $t \geq 2$. The claim then follows since the function $\alpha \mapsto \xi_t^{\mathrm{cpsd}}(A(\alpha))$ is convex by Lemma 5.13. △

5.4 The completely positive rank

The best current approach for lower bounding the completely positive rank of a matrix is due to Fawzi and Parrilo [FP16]. Their approach relies on the atomicity of the completely positive rank, that is, the fact that cp-rank$(A) \leq r$ if and only if A has an atomic decomposition $A = \sum_{k=1}^r v_k v_k^T$ for nonnegative vectors v_k. In other words, if cp-rank$(A) = r$, then A/r can be written as a convex combination of r rank-one positive semidefinite matrices $v_k v_k^T$ that satisfy $0 \leq v_k v_k^T \leq A$ and $v_k v_k^T \preceq A$. Based on this observation Fawzi and Parrilo define the parameter

$$\tau_{\mathrm{cp}}(A) = \min\Big\{\alpha : \alpha \geq 0,\, A \in \alpha{\cdot}\mathrm{conv}\big\{R \in \mathrm{S}^n : 0 \leq R \leq A,\, R \preceq A,\, \mathrm{rank}(R) \leq 1\big\}\Big\},$$

as lower bound for cp-rank(A). They also define the semidefinite programming parameter

$$\begin{aligned} \tau_{\mathrm{cp}}^{\mathrm{sos}}(A) = \min\big\{\alpha : \;& \alpha \in \mathbb{R},\, X \in \mathrm{S}^{n^2}, \\ & \begin{pmatrix} \alpha & \mathrm{vec}(A)^T \\ \mathrm{vec}(A) & X \end{pmatrix} \succeq 0, \\ & X_{(i,j),(i,j)} \leq A_{ij}^2 \quad \text{for} \quad 1 \leq i, j \leq n, \\ & X_{(i,j),(k,\ell)} = X_{(i,\ell),(k,j)} \quad \text{for} \quad 1 \leq i < k \leq n,\ 1 \leq j < \ell \leq n, \\ & X \preceq A \otimes A\big\}, \end{aligned}$$

as an efficiently computable relaxation of $\tau_{\mathrm{cp}}(A)$, and they show rank$(A) \leq \tau_{\mathrm{cp}}^{\mathrm{sos}}(A)$. Therefore we have

$$\mathrm{rank}(A) \leq \tau_{\mathrm{cp}}^{\mathrm{sos}}(A) \leq \tau_{\mathrm{cp}}(A) \leq \text{cp-rank}(A).$$

Instead of the atomic point of view, here we take the Gram factorization perspective, which allows us to obtain bounds by adapting the techniques from Section 5.3 to the commutative setting. Indeed, we may view a factorization $A = (a_i^T a_j)$ by nonnegative vectors as a factorization by diagonal (and thus pairwise commuting) positive semidefinite matrices.

Before presenting the details of our hierarchy of lower bounds, we mention some of our results in order to make the link to the parameters $\tau_{\mathrm{cp}}^{\mathrm{sos}}(A)$ and $\tau_{\mathrm{cp}}(A)$. The direct analogue of $\{\xi_t^{\mathrm{cpsd}}(A)\}$ in the commutative setting leads to a hierarchy that does not converge to $\tau_{\mathrm{cp}}(A)$, but we provide two approaches to strengthen it that do converge to $\tau_{\mathrm{cp}}(A)$. The first approach is based on a generalization of the tensor constraints in $\tau_{\mathrm{cp}}^{\mathrm{sos}}(A)$. We also provide a computationally more efficient version of these tensor constraints, leading to a hierarchy whose second level is at least as good as $\tau_{\mathrm{cp}}^{\mathrm{sos}}(A)$ while being defined by a smaller semidefinite program. The second approach relies on adding localizing constraints for vectors in the unit sphere as in Section 5.3.2.

The following hierarchy is a commutative analogue of the hierarchy from Section 5.3, where we may now add the localizing polynomials $A_{ij} - x_i x_j$ for the pairs $1 \leq i < j \leq n$, which was not possible in the noncommutative setting of the completely positive semidefinite rank. For each $t \in \mathbb{N} \cup \{\infty\}$ we consider the semidefinite program

$$\begin{aligned}\xi_t^{\mathrm{cp}}(A) = \min\big\{L(1) : \;& L \in \mathbb{R}[x_1, \ldots, x_n]_{2t}^*,\\ & L(x_i x_j) = A_{ij} \quad \text{for} \quad i,j \in [n],\\ & L \geq 0 \quad \text{on} \quad \mathcal{M}_{2t}(S_A^{\mathrm{cp}})\big\},\end{aligned}$$

where we set

$$S_A^{\mathrm{cp}} = \big\{\sqrt{A_{ii}}x_i - x_i^2 : i \in [n]\big\} \cup \big\{A_{ij} - x_i x_j : 1 \leq i < j \leq n\big\}.$$

We additionally define $\xi_*^{\mathrm{cp}}(A)$ by adding the constraint $\mathrm{rank}(M(L)) < \infty$ to $\xi_\infty^{\mathrm{cp}}(A)$. We also consider the strengthening $\xi_{t,\dagger}^{\mathrm{cp}}(A)$, where we add to $\xi_t^{\mathrm{cp}}(A)$ the positivity constraints

$$L(gu) \geq 0 \quad \text{for} \quad g \in \{1\} \cup S_A^{\mathrm{cp}} \quad \text{and} \quad u \in [\mathbf{x}]_{2t-\deg(g)} \tag{5.12}$$

and the tensor constraints

$$(L((ww')^c))_{w,w' \in \langle \mathbf{x} \rangle_{=\ell}} \preceq A^{\otimes \ell} \quad \text{for all integers} \quad 2 \leq \ell \leq t, \tag{5.13}$$

which generalize the case $\ell = 2$ used in the relaxation $\tau_{\mathrm{cp}}^{\mathrm{sos}}(A)$. Here, for a word $w \in \langle \mathbf{x} \rangle$, we denote by w^c the corresponding (commutative) monomial in $[\mathbf{x}]$. The tensor constraints (5.13) involve matrices indexed by the *noncommutative* words of length exactly ℓ. In Section 5.4.4 we show a more economical way to rewrite these constraints as $(L(mm'))_{m,m' \in [\mathbf{x}]_{=\ell}} \preceq Q_\ell A^{\otimes \ell} Q_\ell^T$, thus involving smaller matrices indexed by *commutative* monomials of degree ℓ.

Note that, as before, we can strengthen the bounds by adding other localizing polynomials to the set S_A^{cp}. In particular, we can follow the approach of Section 5.3.2. Another possibility is to add localizing constraints specific to the commutative setting: we can add each monomial $u \in [\mathbf{x}]$ to S_A^{cp} (see Section 5.4.5 for an example).

The bounds $\xi_t^{\mathrm{cp}}(A)$ and $\xi_{t,\dagger}^{\mathrm{cp}}(A)$ are monotonically nondecreasing in t and they are invariant under simultaneously permuting the rows and columns of A and under scaling a row and column of A by a positive number. In Propositions 5.16 and 5.17 we show

$$\tau_{\mathrm{cp}}^{\mathrm{sos}}(A) \leq \xi_{t,\dagger}^{\mathrm{cp}}(A) \leq \tau_{\mathrm{cp}}(A) \quad \text{for} \quad t \geq 2,$$

and in Proposition 5.20 we show the equality $\xi_{*,\dagger}^{\mathrm{cp}}(A) = \tau_{\mathrm{cp}}(A)$.

5.4.1 Comparison to $\tau_{\mathrm{cp}}^{\mathrm{sos}}(A)$

We first show that the semidefinite programs defining $\xi_{t,\dagger}^{\mathrm{cp}}(A)$ are valid relaxations for the completely positive rank. More precisely, we show that they lower bound $\tau_{\mathrm{cp}}(A)$.

Proposition 5.16. *For $A \in \mathrm{CP}^n$ and $t \in \mathbb{N} \cup \{\infty, *\}$ we have $\xi_{t,\dagger}^{\mathrm{cp}}(A) \leq \tau_{\mathrm{cp}}(A)$.*

Proof. It suffices to show the inequality for $t = *$. For this consider a decomposition $A = \alpha \sum_{k=1}^r \lambda_k R_k$, where $\alpha \geq 1$, $\lambda_k > 0$, $\sum_{k=1}^r \lambda_k = 1$, $0 \leq R_k \leq A$, $R_k \preceq A$, and $\operatorname{rank} R_k = 1$. There are nonnegative vectors v_k such that $R_k = v_k v_k^T$. Define the linear map $L \in \mathbb{R}[\mathbf{x}]^*$ by $L = \alpha \sum_{k=1}^r \lambda_k L_{v_k}$, where L_{v_k} is the evaluation at v_k mapping any polynomial $p \in \mathbb{R}[\mathbf{x}]$ to $p(v_k)$ (see Equation (4.15) for the definition of a scalar evaluation functional).

The equality $(L(x_i x_j)) = A$ follows from the identity $A = \alpha \sum_{k=1}^r \lambda_k R_k$. The constraints $L((\sqrt{A_{ii}} x_i - x_i^2) p^2) \geq 0$ follow because

$$L_{v_k}(\sqrt{A_{ii}} x_i - x_i^2) p^2) = (\sqrt{A_{ii}} (v_k)_i - (v_k)_i^2) p(v_k)^2 \geq 0,$$

where we use that $(v_k)_i \geq 0$ and $(v_k)_i^2 = (R_k)_{ii} \leq A_{ii}$ implies $(v_k)_i^2 \leq (v_k)_i \sqrt{A_{ii}}$. The constraints $L((A_{ij} - x_i x_j) p^2) \geq 0$ and

$$L(gu) \geq 0 \quad \text{for} \quad g \in \{1\} \cup S_A^{\mathrm{cp}} \quad \text{and} \quad u \in [\mathbf{x}]$$

follow in a similar way.

It remains to show that $X_\ell \preceq A^{\otimes \ell}$ holds for all $\ell \in \mathbb{N}$, where we set $X_\ell = (L(uv))_{u,v \in \langle \mathbf{x} \rangle_{=\ell}}$. Note that $X_1 = A$. We adapt the argument used in [FP16] to show $X_\ell \preceq A^{\otimes \ell}$ using induction on $\ell \geq 2$. Suppose $A^{\otimes(\ell-1)} \succeq X_{\ell-1}$. Combining $A - R_k \succeq 0$ and $R_k \succeq 0$ gives $(A - R_k) \otimes R_k^{\otimes(\ell-1)} \succeq 0$ and thus $A \otimes R_k^{\otimes(\ell-1)} \succeq R_k^{\otimes \ell}$ for each k. Scaling the above by a factor $\alpha\lambda_k$ and summing over k gives

$$A \otimes X_{\ell-1} = \sum_k \alpha \lambda_k A \otimes R_k^{\otimes(\ell-1)} \succeq \sum_k \alpha \lambda_k R_k^{\otimes \ell} = X_\ell.$$

Finally, combining with $A^{\otimes(\ell-1)} - X_{\ell-1} \succeq 0$ and $A \succeq 0$, we obtain

$$A^{\otimes \ell} = A \otimes (A^{\otimes(\ell-1)} - X_{\ell-1}) + A \otimes X_{\ell-1} \succeq A \otimes X_{\ell-1} \succeq X_\ell. \qquad \square$$

Now we show that the new parameter $\xi_{2,\dagger}^{\mathrm{cp}}(A)$ is at least as good as $\tau_{\mathrm{cp}}^{\mathrm{sos}}(A)$. Later in Section 5.4.5 we will give an example where the inequality is strict.

Proposition 5.17. *For $A \in \mathrm{CP}^n$ we have $\tau_{\mathrm{cp}}^{\mathrm{sos}}(A) \leq \xi_{2,\dagger}^{\mathrm{cp}}(A)$.*

Proof. Let L be feasible for $\xi_{2,\dagger}^{\mathrm{cp}}(A)$. We will construct a feasible solution to the program defining $\tau_{\mathrm{cp}}^{\mathrm{sos}}(A)$ with objective value $L(1)$, which implies $\tau_{\mathrm{cp}}^{\mathrm{sos}}(A) \leq L(1)$ and thus the desired inequality. For this set $\alpha = L(1)$ and define the symmetric $n^2 \times n^2$ matrix X by $X_{(i,j),(k,\ell)} = L(x_i x_j x_k x_\ell)$ for $i, j, k, \ell \in [n]$. Then the matrix

$$M := \begin{pmatrix} \alpha & \operatorname{vec}(A)^T \\ \operatorname{vec}(A) & X \end{pmatrix}$$

is positive semidefinite. This follows because M is obtained from the principal submatrix of $M_2(L)$ indexed by the monomials 1 and x_ix_j $(1 \leq i \leq j \leq n)$ where the rows/columns indexed by x_jx_i with $1 \leq i < j \leq n$ are duplicates of the rows/columns indexed by x_ix_j.

We have $L((A_{ij} - x_ix_j)x_ix_j) \geq 0$ for all i, j: For $i \neq j$ this follows using the constraint $L((A_{ij} - x_ix_j)u) \geq 0$ with $u = x_ix_j$ (from (5.12)), and for $i = j$ this follows from

$$L((A_{ii} - x_i^2)x_i^2) = L((\sqrt{A_{ii}} - x_i)^2 + 2(\sqrt{A_{ii}}x_i - x_i^2)) \geq 0,$$

which holds because of (5.6), the constraint $L(p^2) \geq 0$ for $\deg(p) \leq 2$, and the constraint $L(\sqrt{A_{ii}}x_i - x_i^2) \geq 0$. Using that $L(x_ix_j) = A_{ij}$, we get the inequality $X_{(i,j),(i,j)} = L(x_i^2x_j^2) \leq A_{ij}^2$. Furthermore, we have the identity

$$X_{(i,j),(k,\ell)} = L(x_ix_jx_kx_\ell) = L(x_ix_\ell x_kx_j) = X_{(i,\ell),(k,j)},$$

and the constraint $(L(uv))_{u,v\in\langle \mathbf{x}\rangle_{=2}} \preceq A^{\otimes 2}$ implies $X \preceq A \otimes A$. Together this shows that M is feasible for $\tau_{\mathrm{cp}}^{\mathrm{sos}}(A)$. □

5.4.2 Convergence of the basic hierarchy

We first summarize convergence properties of the hierarchy $\xi_t^{\mathrm{cp}}(A)$. Note that unlike in Section 5.3 where we can only claim the inequality $\xi_\infty^{\mathrm{cpsd}}(A) \leq \xi_*^{\mathrm{cpsd}}(A)$, here we can show the equality $\xi_\infty^{\mathrm{cp}}(A) = \xi_*^{\mathrm{cp}}(A)$. This is because we can use Theorem 4.12, which permits to represent certain truncated linear functionals by finite atomic measures.

Proposition 5.18. *Let $A \in \mathrm{CP}^n$. For every $t \in \mathbb{N} \cup \{\infty, *\}$ the optimum in $\xi_t^{\mathrm{cp}}(A)$ is attained, and $\xi_t^{\mathrm{cp}}(A) \to \xi_\infty^{\mathrm{cp}}(A) = \xi_*^{\mathrm{cp}}(A)$ as $t \to \infty$. If $\xi_t^{\mathrm{cp}}(A)$ admits a flat optimal solution, then $\xi_t^{\mathrm{cp}}(A) = \xi_\infty^{\mathrm{cp}}(A)$. Moreover, $\xi_\infty^{\mathrm{cp}}(A) = \xi_*^{\mathrm{cp}}(A)$ is the minimum value of $L(1)$ taken over all linear functionals L that satisfy $A = (L(x_ix_j))$ and that are conic combinations of evaluations at elements of $D(S_A^{\mathrm{cp}})$.*

Proof. We may assume $A \neq 0$. Since $\sqrt{A_{ii}}x_i - x_i^2 \in S_A^{\mathrm{cp}}$ for all i, using (5.6) we obtain that $\mathrm{Tr}(A) - \sum_i x_i^2 \in \mathcal{M}_2(S_A^{\mathrm{cp}})$. By adapting the proof of Proposition 5.1 to the commutative setting, we see that the optimum in $\xi_t^{\mathrm{cp}}(A)$ is attained for $t \in \mathbb{N} \cup \{\infty\}$, and $\xi_t^{\mathrm{cp}}(A) \to \xi_\infty^{\mathrm{cp}}(A)$ as $t \to \infty$.

We now show the inequality $\xi_*^{\mathrm{cp}}(A) \leq \xi_\infty^{\mathrm{cp}}(A)$, which implies that equality holds. For this, let L be optimal for $\xi_\infty^{\mathrm{cp}}(A)$. By Theorem 4.12, the restriction of L to $\mathbb{R}[\mathbf{x}]_2$ extends to a conic combination of evaluations at points in $D(S_A^{\mathrm{cp}})$. It follows that this extension is feasible for $\xi_*^{\mathrm{cp}}(A)$ with the same objective value. This shows that $\xi_*^{\mathrm{cp}}(A) \leq \xi_\infty^{\mathrm{cp}}(A)$, that the optimum in $\xi_*^{\mathrm{cp}}(A)$ is attained, and that $\xi_*^{\mathrm{cp}}(A)$ is the minimum of $L(1)$ over all conic combinations L of evaluations at elements of $D(S_A^{\mathrm{cp}})$ such that $A = (L(x_ix_j))$. Finally, by Theorem 4.11 we have $\xi_t^{\mathrm{cp}}(A) = \xi_\infty^{\mathrm{cp}}(A)$ if $\xi_t^{\mathrm{cp}}(A)$ admits a flat optimal solution. □

Next, we give a reformulation for the parameter $\xi_*^{\mathrm{cp}}(A)$, which is similar to the formulation of $\tau_{\mathrm{cp}}(A)$, although it lacks the constraint $R \preceq A$ which is present in $\tau_{\mathrm{cp}}(A)$.

Proposition 5.19. *We have*

$$\xi_*^{\mathrm{cp}}(A) = \min\Big\{\alpha : \alpha \geq 0,\, A \in \alpha \cdot \mathrm{conv}\big\{R \in \mathrm{S}^n : 0 \leq R \leq A,\, \mathrm{rank}(R) \leq 1\big\}\Big\}.$$

Proof. This follows directly from the reformulation of $\xi_*^{\mathrm{cp}}(A)$ in Proposition 5.18 in terms of conic combinations of evaluations at points in $D(S_A^{\mathrm{cp}})$, after observing that, for $v \in \mathbb{R}^n$, we have $v \in D(S_A^{\mathrm{cp}})$ if and only if the matrix $R = vv^T$ satisfies $0 \leq R \leq A$. □

5.4.3 Additional constraints and convergence to $\tau_{\mathrm{cp}}(A)$

The reformulation of the parameter $\xi_*^{\mathrm{cp}}(A)$ in Proposition 5.19 differs from $\tau_{\mathrm{cp}}(A)$ in that the constraint $R \preceq A$ is missing. In order to have a hierarchy converging to $\tau_{\mathrm{cp}}(A)$ we need to add constraints to enforce that L can be decomposed as a conic combination of evaluation maps at nonnegative vectors v satisfying $vv^T \preceq A$. Here we present two ways to achieve this goal.

First we show that the tensor constraints (5.13) suffice in the sense that $\xi_{*,\dagger}^{\mathrm{cp}}(A) = \tau_{\mathrm{cp}}(A)$ (note that the constraints (5.12) are not needed for this result). However, because of the special form of the tensor constraints we do not know whether $\xi_{t,\dagger}^{\mathrm{cp}}(A)$ admitting a flat optimal solution implies $\xi_{t,\dagger}^{\mathrm{cp}}(A) = \xi_{*,\dagger}^{\mathrm{cp}}(A)$, and we do not know whether $\xi_{\infty,\dagger}^{\mathrm{cp}}(A) = \xi_{*,\dagger}^{\mathrm{cp}}(A)$.

Second, we adapt the approach of adding additional localizing constraints from Section 5.3.2 to the commutative setting, where we do show

$$\xi_{\infty,\mathbb{S}^{n-1}}^{\mathrm{cp}}(A) = \xi_{*,\mathbb{S}^{n-1}}^{\mathrm{cp}}(A) = \tau_{\mathrm{cp}}(A).$$

This yields a doubly indexed sequence of semidefinite programs whose optimal values converge to $\tau_{\mathrm{cp}}(A)$.

Proposition 5.20. *Let $A \in \mathrm{CP}^n$. For every $t \in \mathbb{N} \cup \{\infty\}$ the optimum in $\xi_{t,\dagger}^{\mathrm{cp}}(A)$ is attained. We have $\xi_{t,\dagger}^{\mathrm{cp}}(A) \to \xi_{\infty,\dagger}^{\mathrm{cp}}(A)$ as $t \to \infty$ and $\xi_{*,\dagger}^{\mathrm{cp}}(A) = \tau_{\mathrm{cp}}(A)$.*

Proof. The attainment of the optima in $\xi_{t,\dagger}^{\mathrm{cp}}(A)$ for $t \in \mathbb{N} \cup \{\infty\}$ and the convergence of $\xi_{t,\dagger}^{\mathrm{cp}}(A)$ to $\xi_{\infty,\dagger}^{\mathrm{cp}}(A)$ can be shown in the same way as the analogous statements for $\xi_t^{\mathrm{cp}}(A)$ in Proposition 5.18.

We have seen the inequality $\xi_{*,\dagger}^{\mathrm{cp}}(A) \leq \tau_{\mathrm{cp}}(A)$ in Proposition 5.16. Now we show the reverse inequality. Let L be feasible for $\xi_{*,\dagger}^{\mathrm{cp}}(A)$. We will show that L is feasible for $\tau_{\mathrm{cp}}(A)$, which implies $\tau_{\mathrm{cp}}(A) \leq L(1)$ and thus $\tau_{\mathrm{cp}}(A) \leq \xi_{*,\dagger}^{\mathrm{cp}}(A)$.

By Proposition 5.17 and the fact that $\mathrm{rank}(A) \leq \tau_{\mathrm{cp}}^{\mathrm{sos}}(A)$ we have $L(1) > 0$ (where we assume $A \neq 0$). By Theorem 4.10, we may write

$$L = L(1) \sum_{k=1}^{K} \lambda_k L_{v_k},$$

where $\lambda_k > 0$, $\sum_k \lambda_k = 1$, and L_{v_k} is an evaluation map at a point $v_k \in D(S_A^{\mathrm{cp}})$. We define the matrices $R_k = v_k v_k^T$, so that $A = L(1) \sum_{k=1}^K R_k$. The matrices R_k satisfy $0 \leq R_k \leq A$ since $v_k \in D(S_A^{\mathrm{cp}})$. Clearly also $R_k \succeq 0$. It remains to show that $R_k \preceq A$. For this we use the tensor constraints (5.13). Using that L is a conic combination of evaluation maps, we may rewrite these constraints as

$$L(1) \sum_{k=1}^{K} \lambda_k R_k^{\otimes \ell} \preceq A^{\otimes \ell},$$

from which it follows that $L(1)\lambda_k R_k^{\otimes \ell} \preceq A^{\otimes \ell}$ for all $k \in [K]$. Therefore, for all $k \in [K]$ and all vectors v with $v^T R_k v > 0$ we have

$$L(1)\lambda_k \leq \left(\frac{v^T A v}{v^T R_k v}\right)^{\ell} \quad \text{for all} \quad \ell \in \mathbb{N}.$$

Suppose there is a k such that $R_k \not\preceq A$. Then there exists a v such that we have $v^T R_k v > v^T A v$. As $(v^T A v)/(v^T R_k v) < 1$, letting $\ell \to \infty$ we obtain $L(1)\lambda_k = 0$, reaching a contradiction. It follows that $R_k \preceq A$ for all $k \in [K]$. □

The second approach for reaching $\tau_{\mathrm{cp}}(A)$ is based on using the extra localizing constraints from Section 5.3.2. For a subset $V \subseteq \mathbb{S}^{n-1}$, define $\xi_{t,V}^{\mathrm{cp}}(A)$ by replacing the truncated quadratic module $\mathcal{M}_{2t}(S_A^{\mathrm{cp}})$ in $\xi_t^{\mathrm{cp}}(A)$ by $\mathcal{M}_{2t}(S_{A,V}^{\mathrm{cp}})$, where

$$S_{A,V}^{\mathrm{cp}} = S_A^{\mathrm{cp}} \cup \Big\{ v^T A v - \Big(\sum_{i=1}^{n} v_i x_i \Big)^2 : v \in V \Big\}.$$

Proposition 5.5 can be adapted to the completely positive setting, so that we have a sequence of finite subsets $V_1 \subseteq V_2 \subseteq \ldots \subseteq \mathbb{S}^{n-1}$ with $\xi_{*,V_k}^{\mathrm{cp}}(A) \to \xi_{*,\mathbb{S}^{n-1}}^{\mathrm{cp}}(A)$ as $k \to \infty$. Proposition 5.18 still holds when adding extra localizing constraints, so that for any $k \geq 1$ we have

$$\lim_{t \to \infty} \xi_{t,V_k}^{\mathrm{cp}}(A) = \xi_{*,V_k}^{\mathrm{cp}}(A).$$

Combined with Proposition 5.21 this shows that we have a doubly indexed sequence $\xi_{t,V_k}^{\mathrm{cp}}(A)$ of semidefinite programs that converges to $\tau_{\mathrm{cp}}(A)$ as $t \to \infty$ and $k \to \infty$.

Proposition 5.21. *For $A \in \mathrm{CP}^n$ we have $\xi_{*,\mathbb{S}^{n-1}}^{\mathrm{cp}}(A) = \tau_{\mathrm{cp}}(A)$.*

Proof. The proof is the same as the proof of Proposition 5.19, with the following additional observation: Given a vector $u \in \mathbb{R}^n$, we have $u \in D(S_{A,\mathbb{S}^{n-1}}^{\mathrm{cp}})$ only if $uu^T \preceq A$. The latter follows from the additional localizing constraints: for each $v \in \mathbb{R}^n$ we have

$$0 \leq v^T A v - \Big(\sum_i v_i u_i \Big)^2 = v^T (A - uu^T) v.$$ □

5.4.4 More efficient tensor constraints

Here we show that for any integer $\ell \geq 2$ the tensor constraint

$$A^{\otimes \ell} - (L((ww')^c))_{w,w' \in \langle \mathbf{x} \rangle_{=\ell}} \succeq 0,$$

used in the definition of $\xi^{\mathrm{cp}}_{t,+}(A)$, can be reformulated in a more economical way using matrices indexed by *commutative* monomials in $[\mathbf{x}]_{=\ell}$ instead of noncommutative words in $\langle \mathbf{x}\rangle_{=\ell}$. For this we exploit the symmetry in the matrices $A^{\otimes \ell}$ and $(L((ww')^c))_{w,w'\in\langle \mathbf{x}\rangle_{=\ell}}$ for $L \in \mathbb{R}[\mathbf{x}]^*_{2\ell}$. Recall that for a word $w \in \langle \mathbf{x}\rangle$, we let w^c denote the corresponding (commutative) monomial in $[\mathbf{x}]$.

Define the matrix $Q_\ell \in \mathbb{R}^{[\mathbf{x}]_{=\ell}\times\langle \mathbf{x}\rangle_{=\ell}}$ by

$$(Q_\ell)_{m,w} = \begin{cases} 1/d_m & \text{if } w^c = m, \\ 0 & \text{otherwise,} \end{cases} \tag{5.14}$$

where, for $m = x_1^{\alpha_1}\cdots x_n^{\alpha_n} \in [\mathbf{x}]_{=\ell}$, we define the multinomial coefficient

$$d_m = \left|\{w \in \langle \mathbf{x}\rangle_{=\ell} : w^c = m\}\right| = \frac{\ell!}{\alpha_1!\cdots\alpha_n!}. \tag{5.15}$$

Lemma 5.22. *For $L \in \mathbb{R}[\mathbf{x}]^*_{2\ell}$ we have*

$$Q_\ell (L((ww')^c))_{w,w'\in\langle \mathbf{x}\rangle_{=\ell}} Q_\ell^T = (L(mm'))_{m,m'\in[\mathbf{x}]_{=\ell}}.$$

Proof. For $m, m' \in [\mathbf{x}]_\ell$, the (m,m')-entry of the left-hand side is equal to

$$\sum_{w,w'\in\langle \mathbf{x}\rangle_{=\ell}} Q_{mw}Q_{m'w'}L((ww')^c) = \sum_{\substack{w\in\langle \mathbf{x}\rangle_{=\ell}\\ w^c=m}} \sum_{\substack{w'\in\langle \mathbf{x}\rangle_{=\ell}\\ (w')^c=m'}} \frac{L((ww')^c)}{d_m d_{m'}} = L(mm'). \qquad \square$$

The group $\mathrm{Sym}(\ell)$ of permutations of $[\ell]$ acts on $\langle \mathbf{x}\rangle_{=\ell}$ by $(x_{i_1}\cdots x_{i_\ell})^\sigma = x_{i_{\sigma(1)}}\cdots x_{i_{\sigma(\ell)}}$ for $\sigma \in \mathrm{Sym}(\ell)$. Let

$$P = \frac{1}{\ell!}\sum_{\sigma\in\mathrm{Sym}(\ell)} P_\sigma, \tag{5.16}$$

where, for any $\sigma \in \mathrm{Sym}(\ell)$, $P_\sigma \in \mathbb{R}^{\langle \mathbf{x}\rangle_{=\ell}\times\langle \mathbf{x}\rangle_{=\ell}}$ is the permutation matrix defined by

$$(P_\sigma)_{w,w'} = \begin{cases} 1 & \text{if } w^\sigma = w', \\ 0 & \text{otherwise.} \end{cases}$$

A matrix $M \in \mathbb{R}^{\langle \mathbf{x}\rangle_{=\ell}\times\langle \mathbf{x}\rangle_{=\ell}}$ is said to be $\mathrm{Sym}(\ell)$-*invariant* if $P^\sigma M = MP^\sigma$ for all $\sigma \in \mathrm{Sym}(\ell)$.

Lemma 5.23. *Let Q_ℓ be the matrix from* (5.14)*. If $M \in \mathbb{R}^{\langle \mathbf{x}\rangle_{=\ell}\times\langle \mathbf{x}\rangle_{=\ell}}$ is symmetric and* $\mathrm{Sym}(\ell)$*-invariant, then*

$$M \succeq 0 \iff Q_\ell M Q_\ell^T \succeq 0.$$

Proof. The implication $M \succeq 0 \Longrightarrow Q_\ell M Q_\ell^T \succeq 0$ is immediate. For the other implication we need a preliminary fact. Consider the diagonal matrix $D \in \mathbb{R}^{[\mathbf{x}]_{=\ell}\times[\mathbf{x}]_{=\ell}}$

with $D_{mm} = d_m$ for $m \in [\mathbf{x}]_{=\ell}$. We claim that $Q_\ell^T D Q_\ell = P$, the matrix in (5.16). Indeed, for any $w, w' \in \langle \mathbf{x} \rangle_{=\ell}$, we have

$$\begin{aligned}(Q_\ell^T D Q_\ell)_{ww'} &= \sum_{m \in [\mathbf{x}]_{=\ell}} (Q_\ell)_{mw} (Q_\ell)_{mw'} D_{mm} = \begin{cases} 1/d_m & \text{if } w^c = (w')^c = m, \\ 0 & \text{otherwise} \end{cases} \\ &= \frac{|\{\sigma \in \mathrm{Sym}(\ell) : w^\sigma = w'\}|}{\ell!} = P_{ww'}.\end{aligned}$$

Suppose $Q_\ell M Q_\ell^T \succeq 0$, and let λ be an eigenvalue of M with eigenvector z. Since $MP = PM$, we may assume $Pz = z$, for otherwise we can replace z by Pz, which is still an eigenvector of M with eigenvalue λ. We may also assume z to be a unit vector. Then $\lambda \geq 0$ can be shown using the identity $Q_\ell^T D Q_\ell = P$ as follows:

$$\begin{aligned}\lambda &= z^T M z \\ &= z^T PMPz \\ &= z^T (Q_\ell^T D Q_\ell) M (Q_\ell^T D Q_\ell) z \\ &= (DQ_\ell z)^T (Q_\ell M Q_\ell^T) DQ_\ell z \geq 0.\end{aligned}$$ □

We can now derive our symmetry reduction result:

Proposition 5.24. *For $L \in \mathbb{R}[\mathbf{x}]_{2\ell}^*$ we have*

$$A^{\otimes \ell} - (L((ww')^c))_{w,w' \in \langle \mathbf{x} \rangle_{=\ell}} \succeq 0 \quad \Longleftrightarrow \quad Q_\ell A^{\otimes \ell} Q_\ell^T - (L(mm'))_{m,m' \in [\mathbf{x}]_{=\ell}} \succeq 0.$$

Proof. For any $w, w' \in \langle \mathbf{x} \rangle_{=\ell}$ we have $(P_\sigma A^{\otimes \ell} P_\sigma^T)_{w,w'} = A^{\otimes \ell}_{w^\sigma, (w')^\sigma} = A^{\otimes \ell}_{w,w'}$ and

$$(P_\sigma (L((uu')^c))_{u,u' \in \langle \mathbf{x} \rangle_{=\ell}} P_\sigma^*)_{w,w'} = L((w^\sigma (w')^\sigma)^c) = L((ww')^c).$$

This shows that the matrix $A^{\otimes \ell} - (L((ww')^c))_{w,w' \in \langle \mathbf{x} \rangle_{=\ell}}$ is $\mathrm{Sym}(\ell)$-invariant. Hence the claimed result follows by using Lemma 5.22 and Lemma 5.23. □

5.4.5 Computational examples

Bipartite matrices

Consider the $(p+q) \times (p+q)$ matrices

$$P(a,b) = \begin{pmatrix} (a+q)I_p & J_{p,q} \\ J_{q,p} & (b+p)I_q \end{pmatrix}, \quad a, b \in \mathbb{R}_+,$$

where $J_{p,q}$ denotes the all-ones matrix of size $p \times q$. We have $P(a,b) = P(0,0) + D$ for some nonnegative diagonal matrix D. As can be easily verified, $P(0,0)$ is completely positive with $\text{cp-rank}(P(0,0)) = pq$, so $P(a,b)$ is completely positive with $pq \leq \text{cp-rank}(P(a,b)) \leq pq + p + q$.

For $p = 2$ and $q = 3$ we have $\text{cp-rank}(P(a,b)) = 6$ for all $a, b \geq 0$, which follows from the fact that 5×5 completely positive matrices with at least one zero entry have cp-rank at most 6; see [BSM03, Theorem 3.12]. Fawzi and Parrilo [FP16] show that $\tau_{\mathrm{cp}}^{\mathrm{sos}}(P(0,0)) = 6$, and give a subregion of pairs $(a,b) \in [0,1]^2$ where $5 < \tau_{\mathrm{cp}}^{\mathrm{sos}}(P(a,b)) < 6$. The next lemma shows the bound $\xi_{2,\dagger}^{\mathrm{cp}}(P(a,b))$ is tight for all $a, b \geq 0$ and therefore strictly improves on $\tau_{\mathrm{cp}}^{\mathrm{sos}}$ in this region.

Lemma 5.25. *For $a, b \geq 0$ we have $\xi_{2,\dagger}^{\mathrm{cp}}(P(a,b)) \geq pq$.*

Proof. Let L be feasible for $\xi_{2,\dagger}^{\mathrm{cp}}(P(a,b))$ and let

$$B = \begin{pmatrix} \alpha & c^T \\ c & X \end{pmatrix}$$

be the principal submatrix of $M_2(L)$ where the rows and columns are indexed by

$$\{1\} \cup \{x_i x_j : 1 \leq i \leq p,\, p+1 \leq j \in p+q\}.$$

It follows that c is the all-ones vector $c = \mathbf{1}$. Moreover, if $P(a,b)_{ij} = 0$ for some $i \neq j$, then the constraints $L(x_i x_j u) \geq 0$ and $L((P(a,b)_{ij} - x_i x_j)u) \geq 0$ imply $L(x_i x_j u) = 0$ for all $u \in [\mathbf{x}]_2$. Hence, $X_{x_i x_j, x_k x_\ell} = L(x_i x_j x_k x_\ell) = 0$ whenever $x_i x_j \neq x_k x_\ell$. It follows that X is a diagonal matrix. We write

$$B = \begin{pmatrix} \alpha & \mathbf{1}^T \\ \mathbf{1} & \mathrm{Diag}(z_1, \ldots, z_{pq}) \end{pmatrix}.$$

Since $\begin{pmatrix} 1 & -\mathbf{1}^T \\ -\mathbf{1} & J \end{pmatrix} \succeq 0$ we have

$$0 \leq \mathrm{Tr}\left(\begin{pmatrix} \alpha & \mathbf{1}^T \\ \mathbf{1} & \mathrm{Diag}(z_1, \ldots, z_{pq}) \end{pmatrix} \begin{pmatrix} 1 & -\mathbf{1}^T \\ -\mathbf{1} & J \end{pmatrix} \right) = \alpha - 2pq + \sum_{k=1}^{pq} z_k.$$

Finally, by the constraints $L((P(a,b)_{ij} - x_i x_j)u) \geq 0$ (with $i \in [p], j \in p + [q]$ and $u = x_i x_j$) and $L(x_i x_j) = P(a,b)_{ij}$ we obtain $z_k \leq 1$ for all $k \in [pq]$. Combined with the above inequality, it follows that

$$L(1) = \alpha \geq 2pq - \sum_{k=1}^{pq} z_k \geq pq,$$

and hence $\xi_{2,\dagger}^{\mathrm{cp}}(P(a,b)) \geq pq$. □

Examples related to the DJL-conjecture

The Drew-Johnson-Loewy conjecture [DJL94] states that the maximal cp-rank of an $n \times n$ completely positive matrix is equal to $\lfloor n^2/4 \rfloor$. Recently this conjecture has been disproven for $n = 7, 8, 9, 10, 11$ in [BSU14] and for all $n \geq 12$ in [BSU15] (interestingly, it remains open for $n = 6$). In Table 5.1 we provide the values of some of our bounds on the examples of [BSU14]. Although our bounds are not tight for the cp-rank, they are non-trivial and as such may be of interest for future comparisons. For numerical stability reasons we have evaluated our bounds on scaled versions of the matrices from [BSU14], so that the diagonal entries become equal to 1. The matrices $\tilde{M}_7$, $\tilde{M}_8$ and $\tilde{M}_9$ correspond to the matrices $\tilde{M}$ in Examples 1, 2, 3 of [BSU14], and M_7, M_{11} correspond to the matrices M in Examples 1 and 4. The column $\xi_{2,\dagger}^{\mathrm{cp}} + x_i x_j$ corresponds to the bound $\xi_{2,\dagger}^{\mathrm{cp}}$ where we replace S_A^{cp} by $S_A^{\mathrm{cp}} \cup \{x_i x_j : 1 \leq i < j \leq n\}$.

Table 5.1: Examples from [BSU14] with various bounds on their cp-rank

Example	cp-rank(·)	$\lfloor \frac{n^2}{4} \rfloor$	rank	ξ_1^{cp}	ξ_2^{cp}	$\xi_{2,\dagger}^{\text{cp}}$	$\xi_{2,\dagger}^{\text{cp}}$ $+x_ix_j$	$\xi_{3,\dagger}^{\text{cp}}$
M_7	14	12	7	2.64	4.21	7.21	9.75	10.50
$\widetilde{M_7}$	14	12	7	2.58	4.66	8.43	9.53	10.50
$\widetilde{M_8}$	18	16	8	3.23	5.45	8.74	10.41	13.82
$\widetilde{M_9}$	26	20	9	3.39	5.71	11.60	13.74	17.74
M_{11}	32	30	11	4.32	7.46	20.76	21.84	–

5.5 The nonnegative rank

In this section we adapt the techniques for the cp-rank from Section 5.4 to the asymmetric setting of the nonnegative rank. We now view a factorization $A = (a_i^T b_j)_{i\in[m],j\in[n]}$ by nonnegative vectors as a factorization by positive semidefinite diagonal matrices. That is, we write $A_{ij} = \mathrm{Tr}(X_iX_{m+j})$, with $X_i = \mathrm{Diag}(a_i)$ and $X_{m+j} = \mathrm{Diag}(b_j)$. Note that we can view this as a "partial matrix" setting, where for the symmetric matrix $(\mathrm{Tr}(X_iX_k))_{i,k\in[m+n]}$ of size $m+n$, only the off-diagonal entries at the positions $(i, m+j)$ for $i \in [m], j \in [n]$ are specified.

This asymmetry requires rescaling the factors in order to get upper bounds on their maximal eigenvalues, which is needed to ensure the Archimedean property for the selected localizing polynomials. For this we use the well-known fact that for any $A \in \mathbb{R}_+^{m\times n}$ there exists a factorization $A = (\mathrm{Tr}(X_iX_{m+j}))$ by diagonal nonnegative matrices of size $\mathrm{rank}_+(A)$, such that

$$\lambda_{\max}(X_i), \lambda_{\max}(X_{m+j}) \le \sqrt{A_{\max}} \quad \text{for all} \quad i \in [m], j \in [n],$$

where $A_{\max} := \max_{i,j} A_{ij}$. To see this, observe that for a rank-one matrix $R = uv^T$ with $0 \le R \le A$, one may assume $0 \le u_i, v_j \le \sqrt{A_{\max}}$ for all i, j. Hence, the set

$$S_A^+ = \big\{\sqrt{A_{\max}}x_i - x_i^2 : i \in [m+n]\big\} \cup \big\{A_{ij} - x_ix_{m+j} : i \in [m], j \in [n]\big\}$$

is localizing for A; that is, there exists a minimal factorization $\mathbf{X} \in \mathcal{D}(S_A^+)$ of A.

Given $A \in \mathbb{R}_{\ge 0}^{m\times n}$, for each $t \in \mathbb{N} \cup \{\infty\}$ we consider the semidefinite program

$$\begin{aligned}\xi_t^+(A) = \min\big\{L(1) : {} & L \in \mathbb{R}[x_1, \dots, x_{m+n}]_{2t}^*,\\ & L(x_ix_{m+j}) = A_{ij} \quad \text{for} \quad i \in [m], j \in [n],\\ & L \ge 0 \quad \text{on} \quad \mathcal{M}_{2t}(S_A^+)\big\}.\end{aligned}$$

Moreover, define $\xi_*^+(A)$ by adding the constraint $\mathrm{rank}(M(L)) < \infty$ to the program defining $\xi_\infty^+(A)$. It it easy to check that $\xi_t^+(A) \le \xi_\infty^+(A) \le \xi_*^+(A) \le \mathrm{rank}_+(A)$ for $t \in \mathbb{N}$.

Denote by $\xi_{t,\dagger}^+(A)$ the strengthening of $\xi_t^+(A)$ where we add the positivity constraints

$$L(gu) \ge 0 \quad \text{for} \quad g \in \{1\} \cup S_A^+ \quad \text{and} \quad u \in [\mathbf{x}]_{2t-\deg(g)}. \tag{5.17}$$

Note that these extra constraints can help for finite t, but that they are redundant for $t \in \{\infty, *\}$.

5.5.1 Comparison to other bounds

As in the previous section, we compare our bounds to the bounds by Fawzi and Parrilo [FP16]. They introduce the following parameter $\tau_+(A)$ as an analogue of the bound $\tau_{\mathrm{cp}}(A)$ for the nonnegative rank:

$$\tau_+(A) = \min\Big\{\alpha : \alpha \geq 0,\ A \in \alpha \cdot \mathrm{conv}\big\{R \in \mathbb{R}^{m\times n} : 0 \leq R \leq A,\ \mathrm{rank}(R) \leq 1\big\}\Big\},$$

and the analogue $\tau_+^{\mathrm{sos}}(A)$ of the bound $\tau_{\mathrm{cp}}^{\mathrm{sos}}(A)$ for the nonnegative rank:

$$\begin{aligned}\tau_+^{\mathrm{sos}}(A) = \inf\big\{\alpha : \ & X \in \mathbb{R}^{mn\times mn},\ \alpha \in \mathbb{R},\\ & \begin{pmatrix} \alpha & \mathrm{vec}(A)^T \\ \mathrm{vec}(A) & X \end{pmatrix} \succeq 0,\\ & X_{(i,j),(i,j)} \leq A_{ij}^2 \quad \text{for} \quad 1 \leq i \leq m, 1 \leq j \leq n,\\ & X_{(i,j),(k,\ell)} = X_{(i,\ell),(k,j)} \quad \text{for} \quad 1 \leq i < k \leq m,\ 1 \leq j < \ell \leq n\big\}.\end{aligned}$$

First we give the analogue of Proposition 5.18, whose proof we omit since it is very similar.

Proposition 5.26. *Let $A \in \mathbb{R}_+^{m\times n}$. For every $t \in \mathbb{N} \cup \{\infty, *\}$ the optimum in $\xi_t^+(A)$ is attained, and $\xi_t^+(A) \to \xi_\infty^+(A) = \xi_*^+(A)$ as $t \to \infty$. If $\xi_t^+(A)$ admits a flat optimal solution, then $\xi_t^+(A) = \xi_*^+(A)$. Moreover, $\xi_\infty^+(A) = \xi_*^+(A)$ is the minimum value of $L(1)$ taken over all linear functionals L that satisfy $A = (L(x_i x_{m+j}))$ and that are conic combinations L of trace evaluations at elements of $D(S_A^+)$.*

Now we observe that the parameters $\xi_\infty^+(A)$ and $\xi_*^+(A)$ coincide with $\tau_+(A)$, so that we have a sequence of semidefinite programs converging to $\tau_+(A)$.

Proposition 5.27. *For any $A \in \mathbb{R}_{\geq 0}^{m\times n}$, we have $\xi_\infty^+(A) = \xi_*^+(A) = \tau_+(A)$.*

Proof. The discussion at the beginning of Section 5.5 shows that for any rank-one matrix R satisfying $0 \leq R \leq A$ we may assume that $R = uv^T$ with $(u, v) \in \mathbb{R}_+^m \times \mathbb{R}_+^n$ and $u_i, v_j \leq \sqrt{A_{\max}}$ for $i \in [m], j \in [n]$. Hence, $\tau_+(A)$ can be written as

$$\begin{aligned}&\min\Big\{\alpha : \alpha \geq 0,\ A \in \alpha \cdot \mathrm{conv}\big\{uv^T : (u, v) \in \Big[0, \sqrt{A_{\max}}\Big]^{m+n},\ uv^T \leq A\big\}\Big\}\\ &\qquad = \min\Big\{\alpha : \alpha \geq 0,\ A \in \alpha \cdot \mathrm{conv}\big\{uv^T : (u, v) \in D(S_A^+)\big\}\Big\}.\end{aligned}$$

The equality $\xi_\infty^+(A) = \xi_*^+(A) = \tau_+(A)$ now follows from the reformulation of $\xi_*^+(A)$ in Proposition 5.26 in terms of conic evaluations, after noting that for (u, v) in $\mathbb{R}^m \times \mathbb{R}^n$ we have $(u, v) \in D(S_A^+)$ if and only if the matrix $R = uv^T$ satisfies $0 \leq R \leq A$. □

Analogously to the case of the completely positive rank we have the following proposition. The proof is similar to that of Proposition 4.2, considering now for M the principal submatrix of $M_2(L)$ indexed by the monomials 1 and $x_i x_{m+j}$ for $i \in [m]$ and $j \in [n]$.

Proposition 5.28. *If A is a nonnegative matrix, then $\xi^{+}_{2,\dagger}(A) \geq \tau_+^{\mathrm{sos}}(A)$.*

In the remainder of this section we recall how $\tau_+(A)$ and $\tau_+^{\mathrm{sos}}(A)$ compare to other bounds in the literature. These bounds can be divided into two categories: combinatorial lower bounds and norm-based lower bounds. The following diagram from [FP16] summarizes how $\tau_+^{\mathrm{sos}}(A)$ and $\tau_+(A)$ relate to the combinatorial lower bounds

$$\begin{array}{ccccccc} & & \tau_+^{\mathrm{sos}}(A) & \leq & \tau_+(A) & \leq & \mathrm{rank}_+(A) \\ & & \rotatebox{90}{$\leq$} & & \rotatebox{90}{$\leq$} & & \rotatebox{90}{$\leq$} \\ \omega(\mathrm{RG}(A)) & \leq & \overline{\vartheta}(\mathrm{RG}(A)) & \leq & \chi_{\mathrm{frac}}(\mathrm{RG}(A)) & \leq & \chi(\mathrm{RG}(A)) = \mathrm{rank}_B(A). \end{array}$$

Here $\mathrm{RG}(A)$ is the *rectangular graph*, with $V = \{(i,j) \in [m] \times [n] : A_{ij} > 0\}$ as vertex set and $E = \{((i,j),(k,\ell)) : A_{i\ell}A_{kj} = 0\}$ as edge set. The coloring number of $\mathrm{RG}(A)$ coincides with the well known *rectangle covering number* (also denoted $\mathrm{rank}_B(A)$), which was used, e.g., in [FMP$^+$15] to show that the extension complexity of the correlation polytope is exponential. The clique number of $\mathrm{RG}(A)$ is also known as the *fooling set number* (see, e.g., [FKPT13]). Observe that the above combinatorial lower bounds only depend on the sparsity pattern of the matrix A, and that they are all equal to one for a strictly positive matrix.

Fawzi and Parrilo [FP16] have furthermore shown that the bound $\tau_+(A)$ is at least as good as norm-based lower bounds:

$$\tau_+(A) = \sup_{\substack{\mathcal{N} \text{ monotone and} \\ \text{positively homogeneous}}} \frac{\mathcal{N}^*(A)}{\mathcal{N}(A)}.$$

Here, a function $\mathcal{N} : \mathbb{R}^{m\times n}_+ \to \mathbb{R}_+$ is *positively homogeneous* if $\mathcal{N}(\lambda A) = \lambda \mathcal{N}(A)$ for all $\lambda \geq 0$ and *monotone* if $\mathcal{N}(A) \leq \mathcal{N}(B)$ for $A \leq B$, and $\mathcal{N}^*(A)$ is defined as

$$\begin{gathered} \mathcal{N}^*(A) = \max\{L(A) : L : \mathbb{R}^{m\times n} \to \mathbb{R} \text{ linear and } L(X) \leq 1 \text{ for all } X \in \mathbb{R}^{m\times n}_+ \\ \text{with } \mathrm{rank}(X) \leq 1 \text{ and } \mathcal{N}(X) \leq 1\}. \end{gathered}$$

These bounds are called *norm-based* since norms often provide valid functions $\mathcal{N}$. For example, when $\mathcal{N}$ is the ℓ_∞-norm, Rothvoß [Rot17] used the corresponding lower bound $\mathcal{N}^*(A)/\mathcal{N}(A)$ to show that the matching polytope has exponential extension complexity.

When $\mathcal{N}$ is the Frobenius norm: $\mathcal{N}(A) = (\sum_{i,j} A_{ij}^2)^{1/2}$, the parameter $\mathcal{N}^*(A)$ is known as the *nonnegative nuclear norm*. In [FP15] it is denoted by $\nu_+(A)$ and it is shown to satisfy $\mathrm{rank}_+(A) \geq (\nu_+(A)/\|A\|_F)^2$. Moreover, it is reformulated as

$$\begin{aligned} \nu_+(A) &= \min\Big\{ \sum_i \lambda_i : A = \sum_i \lambda_i u_i v_i^T,\ (\lambda_i, u_i, v_i) \in \mathbb{R}^{1+m+n}_+,\ \|u_i\|_2 = \|v_i\|_2 = 1 \Big\} \\ & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (5.18) \\ &= \max\big\{ \langle A, W\rangle : W \in \mathbb{R}^{m\times n},\ \big(\begin{smallmatrix} I & -W \\ -W^T & I \end{smallmatrix}\big) \text{ is copositive} \big\}, \qquad (5.19) \end{aligned}$$

where the cone of copositive matrices is the dual of the cone of completely positive matrices. Fawzi and Parrilo [FP15] use the copositive formulation (5.19) to provide bounds $\nu_+^{[k]}(A)$ $(k \geq 0)$, based on inner approximations of the copositive cone from [Par00], which converge to $\nu_+(A)$ from below. We now observe that by Theorem 4.12 the atomic formulation of $\nu_+(A)$ from (5.18) can be seen as a moment optimization problem:

$$\nu_+(A) = \min \int_{V(S)} 1 \, d\mu(x) \ \text{ s.t. } \ A_{ij} = \int_{V(S)} x_i x_{m+j} \, d\mu(x) \quad \text{for} \quad i \in [m], j \in [n].$$

Here, the optimization variable μ is required to be a Borel measure on the variety $V(S)$, where

$$S = \{\textstyle\sum_{i=1}^m x_i^2 - 1, \ \sum_{j=1}^n x_{m+j}^2 - 1\}.$$

(The same observation is made in [TS15] for the real nuclear norm of a symmetric 3-tensor and in [Nie17] for symmetric odd-dimensional tensors.) For $t \in \mathbb{N} \cup \{\infty\}$, let $\mu_t(A)$ denote the parameter defined analogously to $\xi_t^+(A)$, where we replace the condition $L \geq 0$ on $\mathcal{M}_{2t}(S_A^+)$ by $L \geq 0$ on $\mathcal{M}_{2t}(\{x_1, \ldots, x_{m+n}\})$ and $L = 0$ on $\mathcal{I}_{2t}(S)$, and let $\mu_*(A)$ be obtained by adding the constraint $\operatorname{rank}(M(L)) < \infty$ to $\mu_\infty(A)$. We have $\mu_t(A) \to \mu_\infty(A) = \mu_*(A) = \nu_+(A)$ by Theorem 4.12 and (a non-normalized analogue of) Theorem 4.13. One can show that $\mu_1(A)$ with the additional constraints $L(u) \geq 0$ for all $u \in [\mathbf{x}]_2$, is at least as good as $\nu_+^{[0]}(A)$. It is not clear how the hierarchies $\mu_t(A)$ and $\nu_+^{[k]}(A)$ compare in general.

5.5.2 Computational examples

We illustrate the performance of our approach by comparing our lower bounds $\xi_{2,\dagger}^+$ and $\xi_{3,\dagger}^+$ to the lower bounds τ_+ and τ_+^{sos} on the two examples considered in [FP16].

All nonnegative 2×2 matrices

For $A(\alpha) = \left(\begin{smallmatrix} 1 & 1 \\ 1 & \alpha \end{smallmatrix}\right)$, Fawzi and Parrilo [FP16] show that

$$\tau_+(A(\alpha)) = 2 - \alpha \quad \text{and} \quad \tau_+^{\text{sos}}(A(\alpha)) = \frac{2}{1+\alpha} \quad \text{for all} \quad 0 \leq \alpha \leq 1.$$

Since the parameters $\tau_+(A)$ and $\tau_+^{\text{sos}}(A)$ are invariant under scaling and permuting rows and columns of A, one can use the identity

$$\begin{pmatrix} 1 & 1 \\ 1 & \alpha \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \alpha \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 1/\alpha \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

to see that their result describes the parameters for all nonnegative 2×2 matrices. By using a semidefinite programming solver for $\alpha = k/100$, $k \in [100]$, we see that $\xi_2^+(A(\alpha))$ coincides with $\tau_+(A(\alpha))$.

The nested rectangles problem

Here we consider the nested rectangles problem as described in [FP16, Section 2.7.2] (see also [MSvS03]). This problem asks for which $a, b \in [-1, 1]$ there exists a triangle T such that $R(a, b) \subseteq T \subseteq P$, where $R(a, b) = [-a, a] \times [-b, b]$ and $P = [-1, 1]^2$, see Figure 5.1 for an illustration.

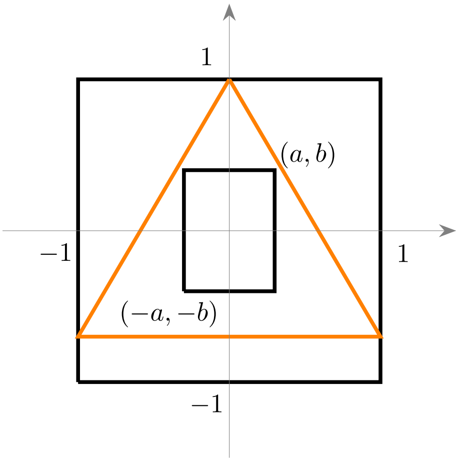


Figure 5.1: An example of the nested rectangles problem where a triangle exists.

In Chapter 2 we have seen how the nonnegative rank relates to the extension complexity of a polytope. In fact, it also relates to extended formulations of nested pairs of polytopes [BFPS15, GG12]. An *extended formulation* of a pair of polytopes $P_1 \subseteq P_2 \subseteq \mathbb{R}^d$ is a (possibly) higher-dimensional polytope K and a projection π such that $\pi(K)$ is nested between P_1 and P_2. Let us suppose $\pi(K) = \{x \in \mathbb{R}^d : y \in \mathbb{R}^k_+, (x, y) \in K\}$ and $K = \{(x, y) : Ex + Fy = g, \; y \in \mathbb{R}^k_+\}$, then k is the *size of the extended formulation*, and the smallest such k is called the *extension complexity* of the pair (P_1, P_2). It is known (cf. [BFPS15, Theorem 1]) that the extension complexity of the pair (P_1, P_2), where

$$P_1 = \mathrm{conv}(\{v_1, \ldots, v_n\}) \quad \text{and} \quad P_2 = \{x : a_i^T x \leq b_i \text{ for } i \in [m]\},$$

is equal to the nonnegative rank of the *generalized slack matrix* $S_{P_1,P_2} \in \mathbb{R}^{m \times n}$, defined by

$$(S_{P_1,P_2})_{ij} = b_j - a_j^T v_i \quad \text{for} \quad i \in [m], j \in [n].$$

It is known that any nonnegative matrix is the slack matrix of some nested pair of polytopes [GPT13, Lemma 4.1] (see also [GG12]).

Applying this to the pair $(R(a, b), P)$, one immediately sees that there exists a polytope K with at most three facets whose projection $T = \pi(K) \subseteq \mathbb{R}^2$ satisfies $R(a, b) \subseteq T \subseteq P$ if and only if the pair $(R(a, b), P)$ admits an extended formulation of size 3. For $a, b > 0$, the polytope T has to be 2-dimensional, therefore K has to be at least 2-dimensional as well; it follows that K and T have to be triangles. Hence there exists a triangle T such that $R(a,b) \subseteq T \subseteq P$ if and only if the nonnegative rank of the slack matrix $S(a,b) := S_{R(a,b),P}$ is equal to 3. One can verify that

$$S(a,b) = \begin{pmatrix} 1-a & 1+a & 1-b & 1+b \\ 1+a & 1-a & 1-b & 1+b \\ 1+a & 1-a & 1+b & 1-b \\ 1-a & 1+a & 1+b & 1-b \end{pmatrix}.$$

Such a triangle exists if and only if $(1+a)(1+b) \le 2$ (see [FP16, Proposition 4] for a proof sketch). To test the quality of their bound, Fawzi and Parrilo [FP16] compute $\tau_+^{\text{sos}}(S(a,b))$ for different values of a and b. In doing so they determine the region where $\tau_+^{\text{sos}}(S(a,b)) > 3$. We do the same for the bounds $\xi_{1,\dagger}^+(S(a,b)), \xi_{2,\dagger}^+(S(a,b))$ and $\xi_{3,\dagger}^+(S(a,b))$, see Figure 5.2. The results show that $\xi_{2,\dagger}^+(S(a,b))$ strictly improves upon the bound $\tau_+^{\text{sos}}(S(a,b))$, and that $\xi_{3,\dagger}^+(S(a,b))$ is again a strict improvement over $\xi_{2,\dagger}^+(S(a,b))$.

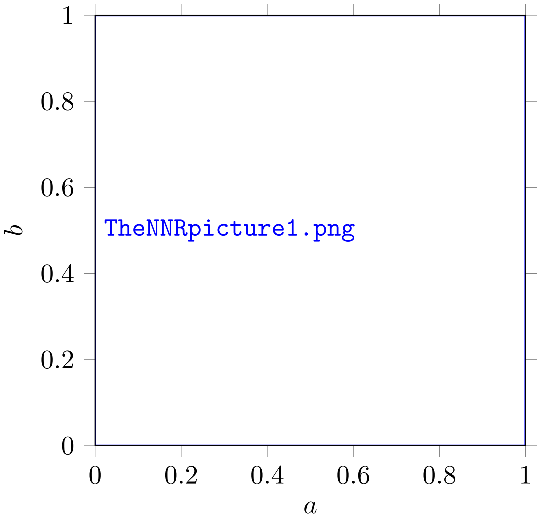


Figure 5.2: The colored region corresponds to $\text{rank}_+(S(a,b)) = 4$. The top right region (black) corresponds to $\xi_{1,\dagger}^+(S(a,b)) > 3$, the two top right regions (black and red) together correspond to $\tau_+^{\text{sos}}(S(a,b)) > 3$, the three top right regions (black, red and yellow) to $\xi_{2,\dagger}^+(S(a,b)) > 3$, and the four top right regions (black, red, yellow, and green) to $\xi_{3,\dagger}^+(S(a,b)) > 3$

5.6 The positive semidefinite rank

The positive semidefinite rank can be seen as an *asymmetric* version of the completely positive semidefinite rank. Hence, as was the case in the previous section for the nonnegative rank, we need to select suitable factors in a minimal factorization in order to be able to bound their maximum eigenvalues and obtain a localizing set of polynomials leading to an Archimedean quadratic module.

For this we can follow, e.g., the approach in [LWdW17, Lemma 5] to rescale a factorization and claim that, for any $A \in \mathbb{R}_+^{m\times n}$ with $\text{psd-rank}_{\mathbb{C}}(A) = d$, there exists a factorization $A = (\langle X_i, X_{m+j}\rangle)$ by matrices $X_1, \ldots, X_{m+n} \in \mathrm{H}_+^d$ such that $\sum_{i=1}^m X_i = I$ and $\mathrm{Tr}(X_{m+j}) = \sum_i A_{ij}$ for all $j \in [n]$. Indeed, starting from any factorization X_i, X_{m+j} in H_+^d of A, we may replace X_i by $X^{-1/2}X_iX^{-1/2}$ and X_{m+j} by $X^{1/2}X_{m+j}X^{1/2}$, where $X := \sum_{i=1}^m X_i$ is positive definite (by minimality of d). This argument shows that the set of polynomials

$$S_A^{\text{psd}} = \big\{x_i - x_i^2 : i \in [m]\big\} \cup \Big\{\Big(\sum_{i=1}^m A_{ij}\Big)x_{m+j} - x_{m+j}^2 : j \in [n]\Big\}$$

is localizing for A; that is, there is *at least one* minimal factorization $\mathbf{X}$ of A such that $g(\mathbf{X}) \succeq 0$ for all polynomials $g \in S_A^{\text{psd}}$. Moreover, for the same minimal factorization $\mathbf{X}$ of A we have $p(\mathbf{X})(1 - \sum_{i=1}^m X_i) = 0$ for all $p \in \mathbb{R}\langle \mathbf{x}\rangle$.

Given $A \in \mathbb{R}_+^{m\times n}$, for each $t \in \mathbb{N} \cup \{\infty\}$ we consider the semidefinite program

$$\begin{aligned}\xi_t^{\text{psd}}(A) = \min\big\{L(1) : \;& L \in \mathbb{R}\langle x_1, \ldots, x_{m+n}\rangle_{2t}^*,\\ & L(x_i x_{m+j}) = A_{ij} \quad \text{for} \quad i \in [m], j \in [n],\\ & L \geq 0 \quad \text{on} \quad \mathcal{M}_{2t}(S_A^{\text{psd}}),\\ & L = 0 \quad \text{on} \quad \mathcal{I}_{2t}(1 - \textstyle\sum_{i=1}^m x_i)\big\}.\end{aligned}$$

We additionally define $\xi_*^{\text{psd}}(A)$ by adding the constraint $\mathrm{rank}(M(L)) < \infty$ to the program defining $\xi_\infty^{\text{psd}}(A)$ (and considering the infimum over $L \in \mathbb{R}\langle \mathbf{x}\rangle^*$ instead of the minimum, since we do not know if the infimum is attained in $\xi_*^{\text{psd}}(A)$). By the above discussion it follows that the parameter $\xi_*^{\text{psd}}(A)$ is a lower bound on $\text{psd-rank}_{\mathbb{C}}(A)$ and we have

$$\xi_1^{\text{psd}}(A) \leq \ldots \leq \xi_t^{\text{psd}}(A) \leq \ldots \leq \xi_\infty^{\text{psd}}(A) \leq \xi_*^{\text{psd}}(A) \leq \text{psd-rank}_{\mathbb{C}}(A).$$

Note that, in contrast to the previous bounds, the parameter $\xi_t^{\text{psd}}(A)$ is not invariant under rescaling the rows of A or under taking the transpose of A (see Section 5.6.2).

It follows from the construction of S_A^{psd} and Equation (5.6) that the quadratic module $\mathcal{M}(S_A^{\text{psd}})$ is Archimedean, and hence the following analogue of Proposition 5.1 can be shown.

Proposition 5.29. *Let $A \in \mathbb{R}_+^{m\times n}$. For each $t \in \mathbb{N}\cup\{\infty\}$, the optimum in $\xi_t^{\text{psd}}(A)$ is attained, and we have*

$$\lim_{t\to\infty} \xi_t^{\text{psd}}(A) = \xi_\infty^{\text{psd}}(A).$$

Moreover, $\xi_\infty^{\text{psd}}(A)$ (resp. $\xi_*^{\text{psd}}(A)$) is equal to the infimum over all $\alpha \geq 0$ for which there exists a unital (resp. finite-dimensional) C^*-algebra $\mathcal{A}$ with tracial state τ and $\mathbf{X} \in \mathcal{D}_{\mathcal{A}}(S_A^{\text{psd}}) \cap \mathcal{V}_{\mathcal{A}}(1 - \sum_{i=1}^m x_i)$ such that $A = \alpha \cdot (\tau(X_iX_{m+j}))_{i\in[m],j\in[n]}$.

5.6.1 Comparison to other bounds

In [LWdW17] the following bound on the complex positive semidefinite rank was derived:

$$\text{psd-rank}_{\mathbb{C}}(A) \geq \sum_{i=1}^{m} \max_{j\in[n]} \frac{A_{ij}}{\sum_i A_{ij}}. \tag{5.20}$$

If a feasible linear form L to $\xi_t^{\text{psd}}(A)$ satisfies the inequalities

$$L(x_i(\sum_i A_{ij} - x_{m+j})) \;\geq\; 0 \qquad \text{for all } i \in [m], j \in [n],$$

then $L(1)$ is at least the above lower bound. Indeed, the inequalities give

$$L(x_i) \geq \max_{j\in[n]} \frac{L(x_i x_{m+j})}{\sum_i A_{ij}} = \max_{j\in[n]} \frac{A_{ij}}{\sum_i A_{ij}}.$$

and hence

$$L(1) = \sum_{i=1}^{m} L(x_i) \geq \sum_{i=1}^{m} \max_{j\in[n]} \frac{A_{ij}}{\sum_i A_{ij}}.$$

The inequalities $L(x_i(\sum_i A_{ij} - x_{m+j})) \geq 0$ are easily seen to be valid for trace evaluations at points of $\mathcal{D}(S_A^{\text{psd}})$. More importantly, as in Lemma 5.8, these inequalities are satisfied by feasible linear forms to the programs $\xi_\infty^{\text{psd}}(A)$ and $\xi_*^{\text{psd}}(A)$. Hence, $\xi_\infty^{\text{psd}}(A)$ and $\xi_*^{\text{psd}}(A)$ are at least as good as the lower bound (5.20).

In [LWdW17] two other fidelity-based lower bounds on the psd-rank were defined; we do not know how they compare to $\xi_t^{\text{psd}}(A)$.

5.6.2 Computational examples

In this section we apply our bounds to some (small) examples taken from the literature, namely 3 × 3 circulant matrices and slack matrices of small polygons.

Nonnegative circulant matrices of size 3

We consider the nonnegative circulant matrices of size 3 which are, up to scaling, of the form

$$M(b,c) = \begin{pmatrix} 1 & b & c \\ c & 1 & b \\ b & c & 1 \end{pmatrix} \quad \text{with} \quad b,c \geq 0.$$

If $b = 1 = c$, then $\text{rank}(M(b,c)) = \text{psd-rank}_{\mathbb{R}}(M(b,c)) = \text{psd-rank}_{\mathbb{C}}(M(b,c)) = 1$. Otherwise we have $\text{rank}(M(b,c)) \geq 2$, which implies $\text{psd-rank}_{\mathbb{K}}(M(b,c)) \geq 2$ for $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$. In [FGP$^+$15, Example 2.7] it is shown that

$$\text{psd-rank}_{\mathbb{R}}(M(b,c)) \leq 2 \quad \Longleftrightarrow \quad 1 + b^2 + c^2 \leq 2(b + c + bc).$$

Hence, if b and c do not satisfy the above inequality then $\text{psd-rank}_{\mathbb{R}}(M(b,c)) = 3$.

To see how good our lower bounds are for this example, we use a semidefinite programming solver to compute $\xi_2^{\text{psd}}(M(b,c))$ for $(b,c) \in [0,4]^2$ (we discretize the region with stepsize 0.01). In Figure 5.3 we see that the bound $\xi_2^{\text{psd}}(M(b,c))$ certifies that $\text{psd-rank}_{\mathbb{R}}(M(b,c)) = \text{psd-rank}_{\mathbb{C}}(M(b,c)) = 3$ for most values of (b,c) for which $\text{psd-rank}_{\mathbb{R}}(M(b,c)) = 3$.

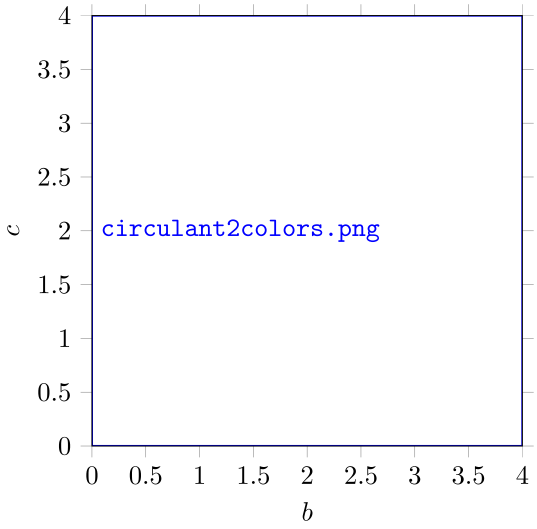


Figure 5.3: The colored region corresponds to the values of (b,c) for which $\text{psd-rank}_{\mathbb{R}}(M(b,c)) = 3$; the outer region (yellow) shows the values of (b,c) for which $\xi_2^{\text{psd}}(M(b,c)) > 2$.

Polygons

Here we consider the slack matrices of two polygons in the plane, where the bounds are sharp (after rounding) and we illustrate the dependence on scaling the rows or taking the transpose. We consider the quadrilateral Q with vertices $(0,0)$, $(0,1)$, $(1,0)$, and $(2,2)$, and the regular hexagon H, whose slack matrices are given by

$$S_Q = \begin{pmatrix} 0 & 0 & 2 & 2 \\ 1 & 0 & 0 & 3 \\ 0 & 1 & 3 & 0 \\ 2 & 2 & 0 & 0 \end{pmatrix}, \qquad S_H = \begin{pmatrix} 0 & 1 & 2 & 2 & 1 & 0 \\ 0 & 0 & 1 & 2 & 2 & 1 \\ 1 & 0 & 0 & 1 & 2 & 2 \\ 2 & 1 & 0 & 0 & 1 & 2 \\ 2 & 2 & 1 & 0 & 0 & 1 \\ 1 & 2 & 2 & 1 & 0 & 0 \end{pmatrix}.$$

Our lower bounds on $\text{psd-rank}_{\mathbb{C}}$ are not invariant under taking the transpose, indeed numerically we have $\xi_2^{\text{psd}}(S_Q) \approx 2.266$ and $\xi_2^{\text{psd}}(S_Q^T) \approx 2.5$. The slack matrix S_Q has $\text{psd-rank}_{\mathbb{R}}(S_Q) = 3$ (a corollary of [GRT13, Theorem 4.3]) and therefore both bounds certify $\text{psd-rank}_{\mathbb{C}}(S_Q) = 3 = \text{psd-rank}_{\mathbb{R}}(S_Q)$.

Secondly, our bounds are not invariant under rescaling the rows of a nonnegative matrix. Numerically we have $\xi_2^{\text{psd}}(S_H) \approx 1.99$ while $\xi_2^{\text{psd}}(DS_H) \approx 2.12$, where $D = \text{Diag}(2,2,1,1,1,1)$. The bound $\xi_2^{\text{psd}}(DS_H)$ is in fact tight (after rounding) for the complex positive semidefinite rank of DS_H (and hence of S_H): in [GGS17] it is shown that $\text{psd-rank}_{\mathbb{C}}(S_H) = 3$.

5.7 Concluding remarks

We provide a Matlab implementation of all the lower bounds introduced in this chapter, at the arXiv submission of the paper on which this chapter is based [GdLL19]. The implementation uses the CVX package [GB14] and supports various semidefinite programming solvers; for our numerical examples we used Mosek [ApS17].

We now mention some corollaries of the results of this chapter and open problems.

Testing membership in the completely positive cone and the completely positive semidefinite cone is another important problem to which our hierarchies can also be applied. It follows from the proof of Proposition 5.18 that if A is not completely positive then, for some order t, the program $\xi_t^{\text{cp}}(A)$ is infeasible or its optimum value is larger than the Carathéodory bound on the cp-rank (which is similar to an earlier result in [Nie14a]). In the noncommutative setting the situation is more complicated: If $\xi_*^{\text{cpsd}}(A)$ is feasible, then $A \in \text{CS}_+$, and if $A \notin \text{CS}_{+,\text{vN}}^n$, then $\xi_\infty^{\text{cpsd}}(A)$ is infeasible (Propositions 5.1 and 5.2). Here $\text{CS}_{+,\text{vN}}^n$ is the cone defined in [BLP17] consisting of the matrices admitting a factorization by positive elements in a von Neumann algebra with a trace. This cone can equivalently be characterized as the set of matrices of the form $\alpha\,(\tau(a_i a_j))$ where $\alpha \in \mathbb{R}_+$, and τ is a tracial state on a C^*-algebra $\mathcal{A}$ and $a_1, \ldots, a_n$ are positive elements in $\mathcal{A}$.

Our lower bounds are on the *complex* version of the (completely) positive semidefinite rank. As far as we are aware, the existing generic lower bounds (except for the dimension-counting rank lower bound) are also on the complex (completely) positive semidefinite rank. It would be interesting to find a lower bound on the *real* (completely) positive semidefinite rank that can go beyond the complex (completely) positive semidefinite rank. In [LWdW17] an ad-hoc argument is given to separate the complex positive semidefinite rank from the real positive semidefinite rank for a specific matrix.

Finally we mention that our approach applies more generally, for instance to the nonnegative tensor rank; see [GdLL19, Sec. 6] for more details.

Chapter 6

Matrices with high completely positive semidefinite rank

This chapter is based on the paper “Matrices with high completely positive semidefinite rank”, by S. Gribling, D. de Laat, and M. Laurent [GdLL17].

As we have seen in Chapter 2, the nonnegative rank, the positive semidefinite rank, and the completely positive rank can all be upper bounded by a function of the matrix size. In fact, the square of the matrix size is a (loose) upper bound on all three. In this chapter we study the question of whether the completely positive semidefinite rank can be upper bounded in terms of the matrix size n. We give an explicit construction of completely positive semidefinite matrices of size $4k^2+2k+2$ with complex completely positive semidefinite rank 2^k for any positive integer k. This shows that if such an upper bound would exist, it has to be at least exponential in the matrix size. For this we exploit connections to quantum information theory and we construct extremal bipartite correlation matrices of large rank.

The main motivation for the above question is to decide whether the completely positive semidefinite cone is closed. Indeed, if an upper bound on $\text{cpsd-rank}_{\mathbb{C}}$ that only depends on the matrix size exists, then a compactness argument shows that the cone is closed. If the cone is closed, that would immediately imply that affine slices of the cone are closed. In Section 3.4 we have seen that an important affine slice of the completely positive semidefinite cone is the set of bipartite quantum correlations. After completion of the work in this chapter, Slofstra [Slo19] showed that the set of bipartite quantum correaltions is not closed. Using the above mentioned connection, this implies that the cone CS_+^n is not closed, for n large enough; see Section 3.2 for a discussion. Hence, no upper bound exists on $\text{cpsd-rank}_{\mathbb{C}}$ that only depends on the matrix size. Nevertheless, it remains challenging to construct explicit classes of completely positive semidefinite matrices with large cpsd-rank.

6.1 Introduction

In this chapter we study the completely positive semidefinite rank, one of the two symmetric matrix factorization ranks that we have seen in the previous chapter. Recall the inclusions

$$\mathrm{CP}^n \subseteq \mathrm{CS}_+^n \subseteq \mathrm{S}_+^n \cap \mathbb{R}_+^{n\times n},$$

where S_+^n is the cone of (real) positive semidefinite $n \times n$ matrices. The three cones coincide for $n \leq 4$ (since doubly nonnegative matrices of size $n \leq 4$ are completely positive), but both inclusions are strict for $n \geq 5$ (see [LP15] for details). By Carathéodory's theorem, the completely positive rank of a matrix in CP^n is at most $\binom{n+1}{2} + 1$. As we now know, due to Slofstra's work [Slo19], there does not exist an upper bound on the cpsd-rank that only depends on the matrix size n. It remains a challenging task to construct explicit families of completely positive semidefinite matrices whose cpsd-rank is large.

In this chapter we construct an explicit family of matrices whose cpsd-rank grows exponentially in the matrix size n. Our main result is the following:

Theorem 6.1. *For each positive integer k, there exists a completely positive semidefinite matrix M of size $4k^2 + 2k + 2$ with* $\text{cpsd-rank}_{\mathbb{C}}(M) = 2^k$.

The proof of this result relies on a connection with quantum information theory and geometric properties of (bipartite) correlation matrices. A first basic ingredient is the fact from [SV17] that a quantum correlation P can be realized in local dimension d if and only if there exists a certain completely positive semidefinite matrix M with $\text{cpsd-rank}_{\mathbb{C}}(M)$ at most d (see Section 3.4). Then, the key idea is to construct a class of quantum correlations P that need large local dimension. In Chapter 7 we will revisit the topic of quantifying the amount of entanglement needed to realize a quantum correlation. There we will not focus on explicit examples, rather we will propose a new measure for the amount of entanglement of a quantum correlation. Specifically, our new measure will differ from the "minimal local dimension" measure by assuming that access to shared randomness is free.

The papers [VP09, Slo11, Ji13] each use different techniques to show the existence of different quantum correlations that require large local dimension. Our main contribution is to provide a unified, explicit construction of the quantum correlations from [VP09] and [Slo11], which uses the seminal work of Tsirelson [Tsi87, Tsi93] combined with convex geometry and recent insights from rigidity theory. In addition, we also give an explicit proof of Tsirelson's bound (see Corollary 6.11) and we show examples where the bound is tight.

More specifically, we construct such quantum correlations from bipartite correlation matrices. For this we use the classical results of Tsirelson [Tsi87, Tsi93], which characterize bipartite correlation matrices in terms of operator representations and, using Clifford algebras, we relate the rank of extremal bipartite correlations to the local dimension of their operator representations. In this way we reduce the problem to finding bipartite correlation matrices that are extreme points of the set of bipartite correlations and have large rank.

Organization. The proof of our main result in Theorem 6.1 boils down to several key ingredients which we treat in the subsequent sections. In Section 6.2 we group old and new results about the set of bipartite correlation matrices. We give a geometric characterization of the extreme points, we revisit some conditions due to Tsirelson and links to rigidity theory, and we construct a class of extreme bipartite correlations with optimal parameters. In Section 6.3 we recall some characterizations, due to Tsirelson, of bipartite correlations in terms of operator representations. We also recall connections to Clifford algebras, and for bipartite correlations that are extreme points we relate their rank to the dimension of their operator representations. In Section 6.4 we show how to construct quantum correlations from bipartite correlation matrices, and we prove the main theorem. Finally, in Section 6.5 we briefly mention some related work.

6.2 The set of bipartite correlations

In this section we define the set $\mathrm{Cor}(m,n)$ of bipartite correlation matrices. The set of bipartite correlation matrices should not be confused with the bipartite (quantum) correlations that we have seen in Section 3.2, they are different objects. However, there is of course a connection: to every $m \times n$ bipartite correlation matrix we can associate a bipartite quantum correlation $P \in \mathbb{R}^{\{0,1\}^2 \times [m] \times [n]}$ and vice versa (see Lemma 6.18 and its proof). This connection dates back to the work of Tsirelson [Tsi87], and it plays a crucial role in this chapter.

After defining the set $\mathrm{Cor}(m,n)$ of bipartite correlation matrices, we discuss properties of the extreme points of $\mathrm{Cor}(m,n)$ which will play a crucial role in the construction of CS_+-matrices with large complex completely positive semidefinite rank. In particular we give a characterization of the extreme points of $\mathrm{Cor}(m,n)$ in terms of extreme points of the related set $\mathcal{E}_{m+n}$ of correlation matrices. We use it to give a simple construction of a class of extreme points of $\mathrm{Cor}(m,n)$ with rank r, when $m = n = \binom{r+1}{2}$. We also revisit conditions for extreme points introduced by Tsirelson [Tsi87] and point out links with universal rigidity. Based on these we can construct extreme points of $\mathrm{Cor}(m,n)$ with rank r when $m = r$ and $n = \binom{r}{2} + 1$, which are used to prove our main result (Theorem 6.1).

Notation. Throughout we set $S = [m]$ and $T = [n]$, and with $S \sqcup T$ we denote the disjoint union of S and T, that is, a set of size $m + n$ whose elements belong either to S or to T.

6.2.1 Bipartite correlations and correlation matrices

A matrix $C \in \mathbb{R}^{m \times n}$ is called a *bipartite correlation matrix* if there exist real unit vectors $x_1, \ldots, x_m$, $y_1, \ldots, y_n \in \mathbb{R}^d$ (for some $d \geq 1$) such that $C_{s,t} = \langle x_s, y_t \rangle$ for all $s \in [m] = S$ and $t \in [n] = T$. Following Tsirelson [Tsi87], any such system of real unit vectors is called a *C-system.* We let $\mathrm{Cor}(m,n)$ denote the set of all $m \times n$ bipartite correlation matrices.

The *elliptope* $\mathcal{E}_n$ is defined as

$$\mathcal{E}_n = \Big\{ E \in \mathrm{S}^n_+ : E_{ii} = 1 \text{ for } i = 1, \ldots, n \Big\},$$

its elements are the *correlation matrices*, which can alternatively be defined as all matrices of the form $(\langle z_i, z_j \rangle)_{i,j=1}^n$ for some real unit vectors $z_1, \ldots, z_n \in \mathbb{R}^d$ $(d \geq 1)$. We have the surjective projection

$$\pi \colon \mathcal{E}_{m+n} \to \mathrm{Cor}(m,n), \begin{pmatrix} Q & C \\ C^T & R \end{pmatrix} \mapsto C. \tag{6.1}$$

Hence, $\mathrm{Cor}(m,n)$ is a projection of the elliptope $\mathcal{E}_{m+n}$ (which is a convex set) and therefore $\mathrm{Cor}(m,n)$ is a convex set. Given $C \in \mathrm{Cor}(m,n)$, any matrix $E \in \mathcal{E}_{m+n}$ such that $\pi(E) = C$ is called an *extension* of C to the elliptope and we let $\mathrm{fib}(C)$ denote the fiber (the set of extensions) of C. If $\mathrm{fib}(C) \neq \emptyset$ we say that C has an extension to the elliptope. If $|\,\mathrm{fib}(C)| = 1$, then we say that C has a unique extension to the elliptope.

Theorem 6.4 below characterizes extreme points of $\mathrm{Cor}(m,n)$ in terms of extreme points of $\mathcal{E}_{m+n}$. It is based on two intermediary results. The first result (whose proof is easy) relates extreme points $C \in \mathrm{Cor}(m,n)$ to properties of their set of extensions $\mathrm{fib}(C)$. It is shown in [ENLV14] in a more general setting.

Lemma 6.2 ([ENLV14, Lemma 2.4]). *Let $C \in \mathrm{Cor}(m,n)$. Then C is an extreme point of $\mathrm{Cor}(m,n)$ if and only if the set $\mathrm{fib}(C)$ is a face of $\mathcal{E}_{m+n}$. Moreover, if C is an extreme point of $\mathrm{Cor}(m,n)$, then every extreme point of $\mathrm{fib}(C)$ is an extreme point of $\mathcal{E}_{m+n}$.*

The second result (from Tsirelson [Tsi87]) shows that every extreme point C of $\mathrm{Cor}(m,n)$ has a unique extension E in $\mathcal{E}_{m+n}$. We give a proof for completeness.

Lemma 6.3 ([Tsi87]). *Assume C is an extreme point of $\mathrm{Cor}(m,n)$.*

(i) If $x_1, \ldots, x_m, y_1, \ldots, y_n$ is a C-system, i.e., $C = (\langle x_s, y_t \rangle)$, then

$$\mathrm{Span}\{x_1, \ldots, x_m\} = \mathrm{Span}\{y_1, \ldots, y_n\}.$$

(ii) The matrix C has a unique extension to a matrix $E \in \mathcal{E}_{m+n}$, and there exists a C-system $x_1, \ldots, x_m, y_1, \ldots, y_n \in \mathbb{R}^r$, with $r = \mathrm{rank}(C)$, such that

$$E = \mathrm{Gram}(x_1, \ldots, x_m, y_1, \ldots, y_n).$$

Proof. We will use the following observation: Each matrix $C = (\langle a_s, b_t \rangle)_{s \in [m], t \in [n]}$, where a_s, b_t are vectors with $\|a_s\|, \|b_t\| \leq 1$, belongs to $\mathrm{Cor}(m,n)$ since it satisfies

$$C_{s,t} = \left\langle \begin{pmatrix} a_s \\ \sqrt{1 - \|a_s\|^2} \\ 0 \end{pmatrix}, \begin{pmatrix} b_t \\ 0 \\ \sqrt{1 - \|b_t\|^2} \end{pmatrix} \right\rangle \quad \text{for all } (s,t) \in S \times T.$$

(i) Set $V = \mathrm{Span}\{x_1, \dots, x_m\}$ and assume $y_k \notin V$ for some $k \in [n]$. Let w denote the orthogonal projection of y_k onto V. Then $\|w\| < 1$ and one can choose a nonzero vector $u \in V$ such that $\|w \pm u\| \leq 1$. Define the matrices $C^{\pm} \in \mathbb{R}^{m\times n}$ by

$$C^{\pm}_{s,t} = \begin{cases} \langle x_s, w \pm u\rangle & \text{if } t = k, \\ \langle x_s, y_t\rangle & \text{if } t \neq k. \end{cases}$$

Then, $C^{\pm} \in \mathrm{Cor}(m, n)$ (by the above observation) and $C = (C^+ + C^-)/2$. As C is an extreme point of $\mathrm{Cor}(m, n)$ one must have $C = C^+ = C^-$. Hence u is orthogonal to each x_s and thus $u = 0$, a contradiction. This shows the inclusion $\mathrm{Span}\{y_1, \dots, y_m\} \subseteq \mathrm{Span}\{x_1, \dots, x_m\}$ and the reverse one follows in the same way.

(ii) Assume that $\{x'_s, y'_t\}$ and $\{x''_s, y''_t\}$ are two C-systems. We now show that $\langle x'_r, x'_s\rangle = \langle x''_r, x''_s\rangle$ for all $r, s \in S$ and $\langle y'_t, y'_u\rangle = \langle y''_t, y''_u\rangle$ for all $t, u \in T$. For this define the vectors

$$x_s = \frac{x'_s \oplus x''_s}{\sqrt{2}} \quad \text{and} \quad y_t = \frac{y'_t \oplus y''_t}{\sqrt{2}},$$

which again form a C-system. Using (i), for any $s \in S$, there exist scalars λ^s_t such that $x_s = \sum_{t\in T} \lambda^s_t y_t$ and thus $x'_s = \sum_{t\in T} \lambda^s_t y'_t$ and $x''_s = \sum_{t\in T} \lambda^s_t y''_t$. This shows

$$\langle x'_r, x'_s\rangle = \sum_{t\in T} \lambda^r_t \langle y'_t, x'_s\rangle = \sum_{t\in T} \lambda^r_t C_{s,t} = \sum_{t\in T} \lambda^r_t \langle y''_t, x''_s\rangle = \langle x''_r, x''_s\rangle$$

for all $r, s \in S$. The analogous argument shows $\langle y'_t, y'_u\rangle = \langle y''_t, y''_u\rangle$ for all $t, u \in T$. This shows C has a unique extension to a matrix $E \in \mathcal{E}_{m+n}$.

Finally, we show that $\mathrm{rank}(E) = \mathrm{rank}(C)$. Say E is the Gram matrix of $x_1, \dots, x_m, y_1, \dots, y_n$. In view of (i), $\mathrm{rank}(E) = \mathrm{rank}\{x_1, \dots, x_m\}$ and thus it suffices to show that $\mathrm{rank}\{x_1, \dots, x_m\} \leq \mathrm{rank}(C)$. For this note that if $\{x_s : s \in I\}$ (for some $I \subseteq S$) is linearly independent then the corresponding rows of C are linearly independent, since $\sum_{s\in I} \lambda_s \langle x_s, y_t\rangle = 0$ (for all $t \in T$) implies $\sum_{s\in I} \lambda_s x_s = 0$ (using (i)) and thus $\lambda_s = 0$ for all s. □

Theorem 6.4. *A matrix C is an extreme point of* $\mathrm{Cor}(m, n)$ *if and only if C has a unique extension to a matrix $E \in \mathcal{E}_{m+n}$ and E is an extreme point of $\mathcal{E}_{m+n}$.*

Proof. Direct application of Lemma 6.2 and Lemma 6.3 (ii). □

We can use the following lemma to construct explicit examples of extreme points of $\mathrm{Cor}(m, n)$ for the case $m = n$.

Lemma 6.5. *Each extreme point of $\mathcal{E}_n$ is an extreme point of* $\mathrm{Cor}(n, n)$.

Proof. Let C be an extreme point of $\mathcal{E}_n$. Define the matrix

$$E = \begin{pmatrix} C & C \\ C & C \end{pmatrix}.$$

Then $E \in \mathcal{E}_{2n}$ is an extension of C. In view of Theorem 6.4 it suffices to show that E is the unique extension of C and that E is an extreme point of $\mathcal{E}_{2n}$. With $e_1, \dots, e_n$

denoting the standard unit vectors in $\mathbb{R}^n$, observe that the vectors $e_i \oplus -e_i$ ($i \in [n]$) lie in the kernel of any matrix $E' \in \mathrm{fib}(C)$. Indeed, since E' and C have an all-ones diagonal we have

$$(e_i \oplus -e_i)^T E'(e_i \oplus -e_i) = 0,$$

and since E' is positive semidefinite this implies that $e_i \oplus -e_i \in \ker(E')$. This implies that $\mathrm{fib}(C) = \{E\}$. We now show that E is an extreme point of $\mathcal{E}_{2n}$. For this let $E_1, E_2 \in \mathcal{E}_{2n}$ and $0 < \lambda < 1$ such that $E = \lambda E_1 + (1-\lambda)E_2$. As E_1, E_2 are positive semidefinite, the kernel of E is the intersection of the kernels of E_1 and E_2. Hence the vectors $e_i \oplus -e_i$ belong to the kernels of E_1 and E_2 and thus

$$E_1 = \begin{pmatrix} C_1 & C_1 \\ C_1 & C_1 \end{pmatrix} \quad \text{and} \quad E_2 = \begin{pmatrix} C_2 & C_2 \\ C_2 & C_2 \end{pmatrix}$$

for some $C_1, C_2 \in \mathcal{E}_n$. Hence, $C = \lambda C_1 + (1-\lambda)C_2$, which implies $C = C_1 = C_2$, since C is an extreme point of $\mathcal{E}_n$. Thus $E = E_1 = E_2$, which completes the proof. □

The above lemma shows how to construct extreme points of $\mathrm{Cor}(n,n)$ from extreme points of the elliptope $\mathcal{E}_n$. Li and Tam [LT95] give the following characterization of the extreme points of $\mathcal{E}_n$.

Theorem 6.6 ([LT95]). *Consider a matrix $E \in \mathcal{E}_n$ with rank r and unit vectors $z_1, \ldots, z_n \in \mathbb{R}^r$ such that $E = \mathrm{Gram}(z_1, \ldots, z_n)$. Then E is an extreme point of $\mathcal{E}_n$ if and only if*

$$\binom{r+1}{2} = \dim(\mathrm{Span}\{z_1 z_1^T, \ldots, z_n z_n^T\}). \tag{6.2}$$

In particular, if E is an extreme point of $\mathcal{E}_n$, then $\binom{r+1}{2} \leq n$.

Example 6.7 ([LT95]). For each integer $r \geq 1$ there exists an extreme point of $\mathcal{E}_n$ of rank r, where $n = \binom{r+1}{2}$. For example, let $e_1, \ldots, e_r$ be the standard basis vectors of $\mathbb{R}^r$ and define

$$E = \mathrm{Gram}\Big(e_1, \ldots, e_r, \frac{e_1 + e_2}{\sqrt{2}}, \frac{e_1 + e_3}{\sqrt{2}}, \ldots, \frac{e_{r-1} + e_r}{\sqrt{2}}\Big).$$

Then E is an extreme point of $\mathcal{E}_n$ of rank r. △

Note that the above example is optimal in the sense that a rank-r extreme point of $\mathcal{E}_n$ can exist only if $n \geq \binom{r+1}{2}$ (by Theorem 6.6). By combining this with Lemma 6.5, this gives a class of extreme points of $\mathrm{Cor}(m,n)$ with rank r and $m = n = \binom{r+1}{2}$.

6.2.2 Tsirelson's bound

If C is an extreme point of $\mathrm{Cor}(m,n)$ with rank r, then by Theorems 6.4 and 6.6 we have $\binom{r+1}{2} \leq m+n$. Tsirelson [Tsi93] claimed the stronger bound $\binom{r+1}{2} \leq m+n-1$ (see Corollary 6.11 below). In the rest of this section we show how to derive this stronger bound of Tsirelson (which is given in [Tsi93] without proof). In the next section, we construct two classes of extreme bipartite correlation matrices, of which one meets Tsirelson's bound. To show Tsirelson's bound we need to investigate in more detail the unique extension property for extreme points of $\mathrm{Cor}(m, n)$.

Let $C \in \mathrm{Cor}(m, n)$ with rank r, let $\{x_s\}$, $\{y_t\}$ be a C-system in $\mathbb{R}^r$, and let

$$E = \mathrm{Gram}(x_1, \ldots, x_m, y_1, \ldots, y_n) \in \mathcal{E}_{m+n}.$$

In view of Theorem 6.4, if C is an extreme point of $\mathrm{Cor}(m, n)$, then E is the unique extension of C in $\mathcal{E}_{m+n}$. This uniqueness property can be rephrased as the requirement that an associated semidefinite program has a unique solution. Namely, consider the following dual pair of semidefinite programs:

$$\max\Big\{0 : X \in \mathrm{S}_+^{S\sqcup T},\ X_{k,k} = 1 \text{ for } k \in S \sqcup T,\ X_{s,t} = C_{s,t} \text{ for } s \in S, t \in T\Big\}, \quad (6.3)$$

$$\min\Big\{\sum_{s\in S} \lambda_s + \sum_{t\in T} \mu_t + 2 \sum_{s\in S, t\in T} W_{s,t} C_{s,t} : \Omega = \begin{pmatrix} \mathrm{Diag}(\lambda) & W \\ W^T & \mathrm{Diag}(\mu) \end{pmatrix} \in \mathrm{S}_+^{S\sqcup T}\Big\}. \quad (6.4)$$

The feasible region of problem (6.3) consists of all possible extensions of C in $\mathcal{E}_{m+n}$, and the feasible region of (6.4) consists of the positive semidefinite matrices Ω whose support (consisting of all off-diagonal pairs (i, j) with $\Omega_{i,j} \neq 0$) is contained in the complete bipartite graph with bipartition $S \sqcup T$. Moreover, strong duality holds: the optimal values of both problems are equal to 0. Finally, for any primal feasible (optimal) X and dual optimal Ω, equality $\Omega X = 0$ holds, which implies that $\mathrm{rank}(X) + \mathrm{rank}(\Omega) \leq m + n$.

Theorem 6.8 below (shown in [LV14] in the more general context of universal rigidity) shows that if the equality $\mathrm{rank}(X) + \mathrm{rank}(\Omega) = m + n$ holds (also known as *strict complementarity*), then X is in fact the *unique* feasible solution of program (6.3), and thus C has a *unique* extension in $\mathcal{E}_{m+n}$.

Theorem 6.8. *Let $C \in \mathrm{Cor}(m, n)$ and let $\{x_s\}$, $\{y_t\}$ be a C-system spanning $\mathbb{R}^r$. Assume $E = \mathrm{Gram}(x_1, \ldots, x_m, y_1, \ldots, y_n)$ is an extreme point of $\mathcal{E}_{m+n}$. If there exists an optimal solution Ω of program* (6.4) *with* $\mathrm{rank}(\Omega) = m + n - r$*, then E is the only extension of C in $\mathcal{E}_{m+n}$.*

Proof. Apply [LV14, Thm. 3.2] to the bar framework $G(\mathbf{p})$, where G is the complete bipartite graph $K_{m,n}$ with bipartition $S \sqcup T$ and where $\mathbf{p} = \{x_s(s \in S), y_t(t \in T)\}$. The conditions (v), (vi) in [LV14, Thm. 3.2] follow from $\Omega E = 0$ and the fact that $\{x_s\}, \{y_t\} \subset \mathbb{R}^r$ are C-systems spanning $\mathbb{R}^r$. □

In addition one can relate uniqueness of an extension of C in the elliptope to the existence of a quadric separating the two point sets $\{x_s\}$ and $\{y_t\}$ (Theorem 6.10 below). Roughly speaking, such a quadric allows us to construct a suitable optimal dual solution Ω and to apply Theorem 6.8. This property was stated by Tsirelson [Tsi93], however without proof. Interestingly, an analogous result was shown recently by Connelly and Gortler [CG17] in the setting of universal rigidity. We will give a sketch of a proof for Theorem 6.10. For this we use Theorem 6.8, arguments in [CG17], and the following basic property of semidefinite programs (which can be seen as an analog of Farkas' lemma for linear programs, see Chapter 1).

Lemma 6.9. *Let $A_1, \ldots, A_m \in \mathrm{S}^n$ and $b \in \mathbb{R}^m$. Assume that there exists a matrix $X_0 \in \mathrm{S}^n$ such that $\langle A_j, X_0 \rangle = b_j$ for all $j \in [m]$. Then exactly one of the following two alternatives holds:*

(i) There exists a matrix $X \succ 0$ such that $\langle A_j, X \rangle = b_j$ for all $j \in [m]$.

(ii) There exists $y \in \mathbb{R}^m$ such that $\Omega = \sum_{j=1}^m y_j A_j \succeq 0$, $\Omega \neq 0$, and $b^T y \leq 0$.

Theorem 6.10 ([Tsi93, Theorems 2.21-2.22]). *Let $C \in \mathrm{Cor}(m,n)$, let $\{x_s\}$, $\{y_t\}$ be a C-system spanning $\mathbb{R}^r$, and let $E = \mathrm{Gram}(x_1, \ldots, x_m, y_1, \ldots, y_n) \in \mathcal{E}_{m+n}$.*

(i) If C is an extreme point of $\mathrm{Cor}(m,n)$, then there exist nonnegative scalars $\lambda_1, \ldots, \lambda_m$, $\mu_1, \ldots, \mu_n$, not all equal to zero, such that

$$\sum_{s=1}^m \lambda_s x_s x_s^T = \sum_{t=1}^n \mu_t y_t y_t^T. \tag{6.5}$$

(ii) If E is an extreme point of $\mathcal{E}_{m+n}$ and there exist strictly positive scalars $\lambda_1, \ldots, \lambda_m, \mu_1, \ldots, \mu_n$ for which relation (6.5) holds, then C is an extreme point of $\mathrm{Cor}(m,n)$.

Proof. (i) By assumption, C is an extreme point of $\mathrm{Cor}(m,n)$, so by Theorem 6.4 E is the only feasible solution of the program (6.3) and E is an extreme point of the elliptope $\mathcal{E}_{m+n}$. As E is an extreme point of $\mathcal{E}_{m+n}$, Theorem 6.6 shows that $\mathrm{rank}(E) = r \leq \binom{r+1}{2} \leq m+n$, and therefore $r < m+n$. It follows that the program (6.3) does not have a positive definite feasible solution. Applying Lemma 6.9 it follows that there exists a nonzero matrix Ω that is feasible for the dual program (6.4) and satisfies

$$\mathrm{Tr}(\Omega E) = \sum_{s \in S} \lambda_s + \sum_{t \in T} \mu_t + 2 \sum_{s \in S, t \in T} W_{s,t} C_{s,t} \leq 0.$$

Since both Ω and E are positive semidefinite, this implies $\Omega E = 0$. This gives:

$$\lambda_s x_s + \sum_{t \in T} W_{s,t} y_t = 0 \ (s \in S), \quad \mu_t y_t + \sum_{s \in S} W_{s,t} x_s = 0 \ (t \in T).$$

Since $\Omega \succeq 0$, the scalars λ_s, μ_t are nonnegative. We claim that they satisfy (6.5). We multiply the left relation by x_s^T and the right one by y_t^T to obtain

$$\lambda_s x_s x_s^T + \sum_{t \in T} W_{s,t} y_t x_s^T = 0 \ (s \in S), \quad \mu_t y_t y_t^T + \sum_{s \in S} W_{s,t} x_s y_t^T = 0 \ (t \in T).$$

Summing the left relation over $s \in S$, and summing the right relation over $t \in T$ and taking the transpose, we get:

$$\sum_{s \in S} \lambda_s x_s x_s^T = -\sum_{s \in S} \sum_{t \in T} W_{s,t} y_t x_s^T = \sum_{t \in T} \mu_t y_t y_t^T,$$

and thus (6.5) holds.

(ii) Assume that E is an extreme point of $\mathcal{E}_{m+n}$ and that there exist strictly positive scalars $\lambda_1, \ldots, \lambda_m, \mu_1, \ldots, \mu_n$ for which (6.5) holds. The key idea is to construct a matrix Ω that is optimal for the program (6.4) and has rank $m+n-r$, since then we can apply Theorem 6.8 and conclude that E is the only extension of C in $\mathcal{E}_{m+n}$. The construction of such a matrix Ω is analogous to the construction given in [CG17] for frameworks (see their Theorem 4.3 and its proof), so we omit the details. □

Corollary 6.11 ([Tsi93]). *If C is an extreme point of* $\mathrm{Cor}(m,n)$*, then*

$$\binom{\mathrm{rank}(C)+1}{2} \leq n+m-1. \tag{6.6}$$

Proof. Let $x_1, \ldots, x_m, y_1, \ldots, y_n \in \mathbb{R}^r$, with $r = \mathrm{rank}(C)$, be a C-system spanning $\mathbb{R}^r$ and let E be their Gram matrix. As E is an extreme point of $\mathcal{E}_{m+n}$, it follows from relation (6.2) that S^r is spanned by the $m+n$ matrices $x_i x_i^T, y_j y_j^T$ where $i \in S$, $j \in T$. The identity (6.5) provides one linear dependence between these $m+n$ matrices and therefore we have that S^r is spanned by a set of $m+n-1$ matrices and thus its dimension $\binom{r+1}{2}$ is at most $m+n-1$. □

Our first construction in the next section provides bipartite correlation matrices for which the bound (6.6) is tight.

6.2.3 Constructing extreme bipartite correlation matrices

We construct two families of extreme points of $\mathrm{Cor}(m,n)$, which we will use in Section 6.4 to construct completely positive semidefinite matrices with exponentially large completely positive semidefinite rank. The first construction meets Tsirelson's bound and is used to prove Theorem 6.1. The second construction will be used to recover one of the results of [Slo11].

We begin with constructing a family of extreme points C_1 of $\mathrm{Cor}(m,n)$ with $\mathrm{rank}(C_1) = r$, $m = r$, and $n = \binom{r}{2}+1$, which thus shows that inequality (6.6) is tight. Such a family of bipartite correlation matrices can also be inferred from [VP09], where the correlation matrices are obtained through analytical methods as optimal solutions of linear optimization problems over $\mathrm{Cor}(m,n)$. Instead, we use the sufficient conditions for extremality of bipartite correlations given above.

For this we will construct matrices $E_1, \Omega_1 \in \mathrm{S}^{r+n}$ that satisfy the conditions of Theorem 6.8; that is, E_1 is an extreme point of $\mathcal{E}_{r+n}$, Ω_1 is positive semidefinite with support contained in the complete bipartite graph $K_{r,n}$, $\mathrm{rank}(E_1) = r$, $\mathrm{rank}(\Omega_1) = n$, and $\Omega_1 E_1 = 0$. Our construction of Ω_1 is inspired by [GP90], which studies the maximum possible rank of extremal positive semidefinite matrices with a complete bipartite support.

Consider the matrix $\widehat{B} \in \mathbb{R}^{r \times \binom{r}{2}}$, whose columns are indexed by the pairs (i,j) with $1 \leq i < j \leq r$, with entries $\widehat{B}_{i,(i,j)} = 1$, $\widehat{B}_{j,(i,j)} = -1$ for $1 \leq i < j \leq r$, and all other entries 0. We also consider the matrix $B \in \mathbb{R}^{r \times n}$ obtained by adjoining to $\widehat{B}$ a last column equal to the all-ones vector e. Note that $BB^T = rI_r$ and $\widehat{B}\widehat{B}^T = rI_r - J_r$. Then define the following matrices:

$$\Omega' = \begin{pmatrix} nI_r & \sqrt{n}B \\ \sqrt{n}B^T & rI_n \end{pmatrix} \in \mathrm{S}^{r+n}, \quad E' = \begin{pmatrix} I_r & -\frac{\sqrt{n}}{r}B \\ -\frac{\sqrt{n}}{r}B^T & \frac{n}{r^2}B^TB \end{pmatrix} \in \mathrm{S}^{r+n}.$$

Since

$$\Omega' = \begin{pmatrix} \sqrt{\frac{n}{r}}B \\ \sqrt{r}I_n \end{pmatrix}\begin{pmatrix} \sqrt{\frac{n}{r}}B \\ \sqrt{r}I_n \end{pmatrix}^T \quad \text{and} \quad E' = \begin{pmatrix} I_r \\ -\frac{\sqrt{n}}{r}B^T \end{pmatrix}\begin{pmatrix} I_r \\ -\frac{\sqrt{n}}{r}B^T \end{pmatrix}^T,$$

it follows that Ω' and E' are positive semidefinite, $\Omega'E' = 0$, $\operatorname{rank}(\Omega') = n$, and $\operatorname{rank}(E') = r$. It suffices now to modify the matrix E' in order to get a matrix E_1 with an all-ones diagonal. For this, consider the diagonal matrix

$$D = I_r \oplus \frac{r}{\sqrt{2n}}I_{n-1} \oplus \sqrt{\frac{r}{n}}I_1$$

and set $E_1 = DE'D$ and $\Omega_1 = D^{-1}\Omega'D^{-1}$. Then E_1 has an all-ones diagonal, it is in fact the Gram matrix of the vectors $e_1, \ldots, e_r$, $(e_i - e_j)/\sqrt{2}$ (for $1 \le i < j \le r$), and $(e_1 + \ldots + e_r)/\sqrt{r}$, and thus E_1 is an extreme point of $\mathcal{E}_{r+n}$. Moreover, $\Omega_1E_1 = 0$, $\operatorname{rank} E_1 = r$, and $\operatorname{rank}\Omega_1 = n$. Therefore the conditions of Theorem 6.8 are fulfilled and we can conclude that the matrix $C_1 = \pi(E_1)$ is an extreme point of $\operatorname{Cor}(r,n)$. So we have shown part (i) in Theorem 6.12 below.

Our second construction is inspired by the XOR-game considered by Slofstra in [Slo11, Sec. 7.2]. We construct a family of extreme points C_2 of $\operatorname{Cor}(m,n)$ with $\operatorname{rank}(C_2) = r - 1$, $m = r$ and $n = \binom{r}{2}$. Define the $(r+n) \times (r+n)$ matrices

$$\Omega_2 = \begin{pmatrix} \sqrt{n}I_r & \widehat{B} \\ \widehat{B}^T & \frac{r}{\sqrt{n}}I_n \end{pmatrix}, \quad E_2 = \begin{pmatrix} \frac{1}{r-1}\widehat{B}\widehat{B}^T & -\frac{r}{2\sqrt{n}}\widehat{B} \\ -\frac{r}{2\sqrt{n}}\widehat{B}^T & \frac{1}{2}\widehat{B}^T\widehat{B} \end{pmatrix}. \tag{6.7}$$

Note that

$$\Omega_2 = \sqrt{n}\begin{pmatrix} \frac{1}{\sqrt{r}}\widehat{B} & \frac{1}{\sqrt{r}}e \\ \sqrt{\frac{r}{n}}I_n & 0 \end{pmatrix}\begin{pmatrix} \frac{1}{\sqrt{r}}\widehat{B} & \frac{1}{\sqrt{r}}e \\ \sqrt{\frac{r}{n}}I_n & 0 \end{pmatrix}^T, \quad E_2 = \begin{pmatrix} \frac{-1}{\sqrt{2n}}\widehat{B}\widehat{B}^T \\ \frac{1}{\sqrt{2}}\widehat{B}^T \end{pmatrix}\begin{pmatrix} \frac{-1}{\sqrt{2n}}\widehat{B}\widehat{B}^T \\ \frac{1}{\sqrt{2}}\widehat{B}^T \end{pmatrix}^T,$$

where we use that $\widehat{B}\widehat{B}^T\widehat{B} = (rI_r - J_r)\widehat{B} = r\widehat{B}$. It follows that Ω_2 and E_2 are positive semidefinite, $\operatorname{rank}(\Omega_2) = n + 1$ and $\operatorname{rank}(E_2) = r - 1$. Moreover, one can check that $\Omega_2E_2 = 0$. In order to be able to apply Theorem 6.8 it remains to verify that E_2 is an extreme point of $\mathcal{E}_{r+n}$.

The above factorization of E_2 shows that it is the Gram matrix of the system of vectors in $\mathbb{R}^r$:

$$\left\{u_k = \frac{1}{\sqrt{2n}}(e - re_k) : k \in [r]\right\} \cup \left\{v_{ij} = \frac{1}{\sqrt{2}}(e_i - e_j) : 1 \le i < j \le r\right\}.$$

As the vectors u_k, v_{ij} lie in $\mathbb{R}^r$ while E_2 has rank $r-1$ we need to consider another Gram representation of E_2 by vectors in $\mathbb{R}^{r-1}$. For this, let Q be an $r \times r$ orthogonal matrix with columns $p_1, \ldots, p_r$ and $p_r = \frac{1}{\sqrt{r}}e$. Then the vectors $\{Q^T u_k\} \cup \{Q^T v_{ij}\}$ form again a Gram representation of E_2. Furthermore, as all u_k, v_{ij} are orthogonal to the vector p_r it follows that the vectors $Q^T u_k$ and $Q^T v_{ij}$ are all orthogonal to $Q^T p_r = e_r$. Hence $Q^T u_k = (x_k, 0)$ and $Q^T v_{ij} = (y_{ij}, 0)$ for some vectors $x_k, y_{ij} \in \mathbb{R}^{r-1}$ which now provide a Gram representation of E_2 in $\mathbb{R}^{r-1}$.

In order to conclude that E_2 is an extreme point of $\mathcal{E}_{r+n}$ it suffices, by Theorem 6.6, to verify that the set $\{x_k x_k^T\} \cup \{y_{ij} y_{ij}^T\}$ spans the whole space S^{r-1}. Equivalently, it suffices to show that the set $\{Q^T u_k u_k^T Q\} \cup \{Q^T v_{ij} v_{ij}^T Q\}$ spans the subspace $\{R \oplus 0 : R \in \mathrm{S}^{r-1}\}$ of S^r, or, in other words, that the set $\{u_k u_k^T\} \cup \{v_{ij} v_{ij}^T\}$ spans the subspace

$$\mathcal{M} := \{Q(R \oplus 0)Q^T : R \in \mathrm{S}^{r-1}\} \subseteq \mathrm{S}^r.$$

Observe that $\dim(\mathcal{M}) = \binom{r}{2}$. We also have that $\text{span}\{v_{ij} v_{ij}^T : 1 \le i < j \le r\}$ is contained in

$$\text{span}(\{u_k u_k^T : k \in [r]\} \cup \{v_{ij} v_{ij}^T : 1 \le i < j \le r\}) \subseteq \mathcal{M},$$

and that

$$\text{span}\{v_{ij} v_{ij}^T : 1 \le i < j \le r\} = \text{span}\{(e_i - e_j)(e_i - e_j)^T : 1 \le i < j \le r\}$$

has dimension $\binom{r}{2}$. Therefore, equality holds throughout:

$$\text{span}(\{u_k u_k^T : k \in [r]\} \cup \{v_{ij} v_{ij}^T : 1 \le i < j \le r\}) = \mathcal{M},$$

and thus E_2 is an extreme point of $\mathcal{E}_{r+n}$.

This shows that the conditions of Theorem 6.8 are satisfied and we can conclude that the matrix $C_2 = \pi(E_2)$ is an extreme point of $\text{Cor}(r, n)$. So we have shown part (ii) in Theorem 6.12 below.

Theorem 6.12. *Consider an integer $r \ge 1$ and let $e_1, \ldots, e_r$ denote the standard unit vectors in $\mathbb{R}^r$.*

(i) There exists a matrix C_1 which is an extreme point of $C(r, \binom{r}{2} + 1)$ and has rank r. We can take C_1 to be the matrix with columns $(e_i - e_j)/\sqrt{2}$ (for $1 \le i < j \le r$) and $(e_1 + \ldots + e_r)/\sqrt{r}$.

(ii) There exists a matrix C_2 which is an extreme point of $\text{Cor}(r, \binom{r}{2})$ and has rank $r - 1$. We can take C_2 to be the matrix whose columns are $-\sqrt{\frac{r}{2(r-1)}}(e_i - e_j)$ for $1 \le i < j \le r$.

We now connect the above results to the study of XOR-games, a particular type of nonlocal game. We refer to Section 3.3 for an introduction to the relevant aspects of XOR-games. We explain how our second construction permits to recover a lower bound of Slofstra [Slo11] for the amount of entanglement needed by any optimal quantum strategy for the XOR-game he considers in [Slo11, Sec. 7.2]. Recall that the goal of an XOR-game is to find a quantum strategy with maximal winning probability, or, equivalently, a strategy that maximizes the bias of the game. An XOR-game is given by a game matrix, see Equation (3.13), and the game presented in [Slo11, Sec. 7.2] has game matrix $\widehat{B}$ as defined above. To rephrase the presentation of Section 3.3 in the language of correlation matrices, an optimal quantum strategy corresponds to an optimal solution of the following optimization problem:

$$\max\{\langle \widehat{B}, C\rangle : C \in \mathrm{Cor}(m,n)\}. \tag{6.8}$$

Slofstra [Slo11] showed (using the notion of 'solution algebra' of the game) that any tensor operator representation of any optimal solution C of (6.8) has local dimension at least $2^{\lfloor (r-1)/2 \rfloor}$ (see Section 6.3 for the definition of a tensor operator representation). As we now point out this can also be derived from Tsirelson's results using our treatment.

For this note first that problem (6.8) is equivalent to

$$\min\{\langle \widehat{B}, C\rangle : C \in \mathrm{Cor}(m,n)\} \tag{6.9}$$

(since $C \in \mathrm{Cor}(m,n)$ if and only if $-C \in \mathrm{Cor}(m,n)$). Problem (6.9) is in turn equivalent to the following optimization problem over the elliptope:

$$\min\{\langle \Omega_2, E\rangle : E \in \mathcal{E}_{m+n}\}, \tag{6.10}$$

with Ω_2 being defined as above in Equation (6.7) (since $E \in \mathcal{E}_{m+n}$ is optimal for (6.10) if and only if $C = \pi(E) \in \mathrm{Cor}(m,n)$ is optimal for (6.9)). As Ω_2 is positive semidefinite and $\langle \Omega_2, E_2\rangle = 0$, it follows that E_2 is optimal for (6.10) and thus $C_2 = \pi(E_2)$ is optimal for (6.9). Moreover, as $\mathrm{rank}(E_2) = m+n-\mathrm{rank}(\Omega_2)$ is the largest possible rank of an optimal solution of (6.10), it follows from a geometric property of semidefinite programming that E_2 must lie in the relative interior of the set of optimal solutions of (6.10). This, combined with the fact that E_2 is an extreme point of $\mathcal{E}_{m+n}$, implies that E_2 is the unique optimal solution of (6.10) and thus C_2 is the unique optimal solution of (6.9). Finally, as C_2 is an extreme point of $\mathrm{Cor}(m,n)$ with rank $r-1$, we can conclude using Corollary 6.17 below that any tensor operator representation of C_2 uses local dimension at least $2^{\lfloor (r-1)/2 \rfloor}$, and the same holds for the unique optimal solution $-C_2$ of (6.8).

6.3 Lower bounding the size of operator representations

We start with recalling, in Theorem 6.13, some equivalent characterizations for bipartite correlations in terms of operator representations, due to Tsirelson. These equivalent characterizations will eventually allow us to associate a bipartite quantum correlation to each bipartite correlation matrix (Lemma 6.18), therefore the terminology used here reflects the terminology used in Section 3.2 where we defined bipartite quantum correlations.

Consider a matrix $C \in \mathbb{R}^{m\times n}$. We say that C admits a *tensor operator representation* if there exist an integer d (the *local dimension*), a unit vector $\psi \in \mathbb{C}^d \otimes \mathbb{C}^d$, and Hermitian $d \times d$ matrices $\{X_s\}_{s=1}^m$ and $\{Y_t\}_{t=1}^n$ with spectra contained in $[-1,1]$, such that $C_{s,t} = \psi^*(X_s \otimes Y_t)\psi$ for all s and t.

Moreover we say that C admits a (finite-dimensional) *commuting operator representation* if there exist an integer d, a Hermitian positive semidefinite $d \times d$ matrix W with $\text{trace}(W) = 1$, and Hermitian $d \times d$ matrices $\{X_s\}$ and $\{Y_t\}$ with spectra contained in $[-1,1]$, such that $X_sY_t = Y_tX_s$ and $C_{s,t} = \text{Tr}(X_sY_tW)$ for all s and t. A commuting operator representation is said to be *pure* if $\text{rank}(W) = 1$.

The above definitions of tensor and commuting operator representations are equivalent to those used in Section 3.2 for bipartite quantum correlations: we change variables from POVMs $\{X_s^0, X_s^1\}$ to observables $X_s = X_s^0 - X_s^1$ in a similar fashion as we have seen in Equation (3.10).

Existence of these various operator representations relies on using Clifford algebras. For an integer $r \geq 1$ the *Clifford algebra* $\mathcal{C}(r)$ of order r can be defined as the universal C^*-algebra with Hermitian generators $a_1, \ldots, a_r$ and relations

$$a_i^2 = 1 \quad \text{and} \quad a_ia_j + a_ja_i = 0 \quad \text{for} \quad i \neq j. \tag{6.11}$$

We call these relations the *Clifford relations.* To represent the elements of $\mathcal{C}(r)$ by matrices we can use the following map, which is a $*$-isomorphism onto its image:

$$\varphi_r \colon \mathcal{C}(r) \to \mathbb{C}^{2^{\lceil r/2 \rceil} \times 2^{\lceil r/2 \rceil}}, \; \varphi_r(a_i) = \begin{cases} Z^{\otimes \frac{i-1}{2}} \otimes X \otimes I^{\otimes \lceil \frac{r}{2} \rceil - \frac{i+1}{2}} & \text{for } i \text{ odd,} \\ Z^{\otimes \frac{i-2}{2}} \otimes Y \otimes I^{\otimes \lceil \frac{r}{2} \rceil - \frac{i}{2}} & \text{for } i \text{ even.} \end{cases} \tag{6.12}$$

Here we use the *Pauli matrices*

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -\mathbf{i} \\ \mathbf{i} & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

For even r the representation φ_r is irreducible and thus $\mathcal{C}(r)$ is isomorphic to the full matrix algebra with matrix size $2^{r/2}$. For odd r the representation φ_r decomposes as a direct sum of two irreducible representations, each of dimension $2^{\lfloor r/2 \rfloor}$. Therefore, if $X_1, \ldots, X_r$ is a set of Hermitian matrices satisfying the relations $X_i^2 = I$ and $X_iX_j + X_jX_i = 0$ for $i \neq j$, then they must have size at least $2^{\lfloor r/2 \rfloor}$. We refer to [Pro07, Section 5.4] for details about (representations of) Clifford algebras.

We are now ready to state the fundamental result of Tsirelson that connects bipartite correlation matrices to bipartite quantum correlations. As we have seen in Section 3.3.1, this result allows us to find the quantum value of a 2-player XOR game using semidefinite programming.

Theorem 6.13 ([Tsi87]). *Let $C \in \mathbb{R}^{m \times n}$. The following statements are equivalent:*

1. *C is a bipartite correlation.*
2. *C admits a tensor operator representation.*
3. *C admits a pure commuting operator representation.*
4. *C admits a commuting operator representation.*

Proof. We first prove the core of the theorem, the implication (1) ⇒ (2). Let $C \in \mathrm{Cor}(m,n)$. That means there exist unit vectors $\{x_s\}$ and $\{y_t\}$ in $\mathbb{R}^r$, where $r = \mathrm{rank}(C)$, such that $C_{s,t} = \langle x_s, y_t\rangle$ for all s and t. Set $d = 2^{\lfloor r/2 \rfloor}$. Let π be an irreducible representation of $\mathcal{C}(r)$ by matrices of size d (note that for r even we could use the explicit representation φ_r). Then we define the $d \times d$ matrices

$$X_s = \sum_{i=1}^r (x_s)_i \pi(a_i), \quad Y_t = \sum_{i=1}^r (y_t)_i \pi(a_i)^T,$$

and set $\psi = \frac{1}{\sqrt{d}} \sum_{k=1}^d e_k \otimes e_k$. The Clifford relations imply that $\pi(a_i)^2 = I_d$ and $\mathrm{Tr}(\pi(a_i)\pi(a_j)) = 0$ for all $i \neq j$. From this it follows that

$$\mathrm{Tr}(X_s Y_t^T) = \sum_{i,j \in [r]} (x_s)_i (y_t)_j \,\mathrm{Tr}(\pi(a_i)\pi(a_j)) = d\,\langle x_s, y_t\rangle.$$

Therefore, we have

$$C_{s,t} = \langle x_s, y_t\rangle = \mathrm{Tr}(X_s Y_t^T)/d = \psi^*(X_s \otimes Y_t)\psi \quad \text{for all} \quad s \in S, t \in T,$$

where the last inequality follows from Equation (3.21) (using that $\psi = \mathrm{vec}(I_d/d)$).

The eigenvalues of the matrices $\pi(a_1), \ldots, \pi(a_r)$ lie in $\{-1, 1\}$. Using the Clifford relations (6.11) one can show that $X_s^2 = I_d = Y_t^2$ for all $s \in S, t \in T$. Thus, $(\{X_s\}, \{Y_t\}, \psi)$ is a tensor operator representation of C.

(2) ⇒ (3) If $(\{X_s\}, \{Y_t\}, \psi)$ is a tensor operator representation of C, then the operators $X_s \otimes I$ and $I \otimes Y_t$ commute, and by using the identity

$$\psi^*(X_s \otimes Y_t)\psi = \mathrm{Tr}((X_s \otimes I)(I \otimes Y_t)\psi\psi^*)$$

we see that $(\{X_s \otimes I\}, \{I \otimes Y_y\}, \psi\psi^*)$ is a pure commuting operator representation.

(3) ⇒ (4) This is immediate.

(4) ⇒ (1) Suppose $(\{X_s\}, \{Y_t\}, W)$ is a commuting operator representation of C. Since W is positive semidefinite and has trace 1, there exist nonnegative scalars λ_i and orthonormal unit vectors $\psi_i \in \mathbb{C}^d \otimes \mathbb{C}^d$ such that $W = \sum_i \lambda_i \psi_i \psi_i^*$ and $\sum_i \lambda_i = 1$. Then,

$$C_{s,t} = \mathrm{Tr}(X_s Y_t W) = \sum_i \lambda_i \,\mathrm{Tr}(X_s Y_t \psi_i \psi_i^*) = \sum_i \lambda_i \psi_i^* X_s Y_t \psi_i.$$

So, with

$$x_s = \bigoplus_i \sqrt{\lambda_i} \begin{pmatrix} \mathrm{Re}(X_s\psi_i) \\ \mathrm{Im}(X_s\psi_i) \end{pmatrix} \quad \text{and} \quad y_t = \bigoplus_i \sqrt{\lambda_i} \begin{pmatrix} \mathrm{Re}(Y_t\psi_i) \\ \mathrm{Im}(Y_t\psi_i) \end{pmatrix}$$

we have $C_{s,t} = \langle x_s, y_t\rangle$ and $\|x_s\|, \|y_s\| \leq 1$, and by using the observation in the proof of Lemma 6.3 we can extend the vectors x_s and y_t to unit vectors. □

The proof of the above theorem also shows the following result.

Corollary 6.14. *If C is a bipartite correlation matrix of rank r, then it admits a tensor operator representation in local dimension $2^{\lfloor r/2\rfloor}$. If C is a bipartite correlation matrix that admits a tensor operator representation in local dimension d, then it has a commuting operator representation by matrices of size d^2.*

The remainder of this section is devoted to showing that there are bipartite correlation matrices for which every operator representation requires a large dimension.

For this we need two more definitions. A commuting operator representation $(\{X_s\}, \{Y_t\}, W)$ is *nondegenerate* if there does not exist a projection matrix $P \neq I$ such that $PWP = W$, $X_sP = PX_s$, and $Y_tP = PY_t$ for all s and t. It is said to be *Clifford* if there exist matrices $Q \in \mathbb{R}^{m\times m}$ and $R \in \mathbb{R}^{n\times n}$ with all-ones diagonals, such that

$$\begin{aligned} X_sX_{s'} + X_{s'}X_s &= 2Q_{s,s'}I \quad \text{for all} \quad s, s' \in S, \\ Y_tY_{t'} + Y_{t'}Y_t &= 2R_{t,t'}I \quad \text{for all} \quad t, t' \in T. \end{aligned}$$

Note that the matrices Q and R are both symmetric.

We will use the following theorem from Tsirelson as crucial ingredient.

Theorem 6.15 ([Tsi87, Theorem 3.1])**.** *If C is an extreme point of $\mathrm{Cor}(m, n)$, then any nondegenerate commuting operator representation of C is Clifford.*

We can now state and prove the main result of this section.

Theorem 6.16. *Let C be an extreme point of $\mathrm{Cor}(m, n)$ and let $r = \mathrm{rank}(C)$. Every commuting operator representation of C uses matrices of size at least $(2^{\lfloor r/2\rfloor})^2$.*

Proof. Let $(\{X_s\}, \{Y_t\}, W)$ be a commuting operator representation of C where X_s, Y_t and W are matrices of size d. We will show that $d \geq (2^{\lfloor r/2\rfloor})^2$. If this representation is degenerate, then there exists a projection matrix $P \neq I$ such that $PWP = W$, $X_sP = PX_s$, and $Y_tP = PY_t$ for all s and t. Let $P = \sum_{i=1}^k v_iv_i^*$ be its spectral decomposition, where the vectors $v_1, \dots, v_k$ are orthonormal, and set $U = (v_1, \dots, v_k)$. Then one can verify that $(\{U^*X_sU\}, \{U^*Y_sU\}, U^*WU)$ is a commuting operator representation of C of smaller dimension. So, since we are proving a lower bound on the dimension, we may assume $(\{X_s\}, \{Y_t\}, W)$ to be a nondegenerate commuting operator representation.

By extremality of C we may assume the operator representation is pure. Hence, there is a unit vector ψ such that $W = \psi\psi^*$. This gives

$$C_{s,t} = \mathrm{Tr}(X_sY_tW) = \psi^*X_sY_t\psi = \langle x_s, y_t\rangle,$$

where

$$x_s = \begin{pmatrix}\mathrm{Re}(X_s\psi)\\ \mathrm{Im}(X_s\psi)\end{pmatrix} \quad \text{and} \quad y_t = \begin{pmatrix}\mathrm{Re}(Y_t\psi)\\ \mathrm{Im}(Y_t\psi)\end{pmatrix}.$$

These vectors x_s and y_t are unit vectors because C is extreme (see the proof of Lemma 6.3), and therefore, they form a C-system.

By Theorem 6.15 the commuting operator representation $(\{X_s\}, \{Y_t\}, W)$ is Clifford. So, there exist matrices $Q \in \mathbb{R}^{m\times m}$ and $R \in \mathbb{R}^{n\times n}$ with all-one diagonals such that

$$\begin{aligned} X_s X_{s'} + X_{s'} X_s &= 2Q_{s,s'} I \quad \text{for all} \quad s, s' \in S, \\ Y_t Y_{t'} + Y_{t'} Y_t &= 2R_{t,t'} I \quad \text{for all} \quad t, t' \in T. \end{aligned}$$

We show that

$$E = \begin{pmatrix} Q & C \\ C^T & R \end{pmatrix}$$

is an extension of C to the elliptope of C. For this, we have to show $Q_{s,s'} = \langle x_s, x_{s'}\rangle$ and $R_{t,t'} = \langle y_t, y_{t'}\rangle$. Indeed,

$$\begin{aligned} \langle x_s, x_{s'}\rangle + \langle x_{s'}, x_s\rangle &= \operatorname{Re}\big(\psi^* X_s X_{s'} \psi + \psi^* X_{s'} X_s \psi\big) \\ &= \operatorname{Re}\big(\psi^*(X_s X_{s'} + X_{s'} X_s)\psi\big) \\ &= \operatorname{Re}\big(\psi^*(2Q_{s,s'} I)\psi\big) = 2Q_{s,s'}, \end{aligned}$$

and in the same way $\langle y_t, y_{t'}\rangle + \langle y_{t'}, y_t\rangle = 2R_{t,t'}$.

By Theorem 6.4 the matrix E is the unique extension of C to the elliptope. Furthermore, Lemma 6.3 tells us that $\mathrm{rank}(Q) = \mathrm{rank}(R) = \mathrm{rank}(C) = r$.

Now consider the spectral decomposition $Q = \sum_{i=1}^r \alpha_i v_i v_i^*$, where the vectors $v_1, \ldots, v_r$ are orthonormal, and the algebra $\mathbb{C}\langle A_1, \ldots, A_r\rangle$, where

$$A_i = \frac{1}{\sqrt{\alpha_i}} \sum_{s=1}^m (v_i)_s X_s \quad \text{for} \quad i \in [r].$$

We have

$$\begin{aligned} A_i A_j + A_j A_i &= \frac{1}{\sqrt{\alpha_i \alpha_j}} \sum_{s,s'=1}^m \big((v_i)_s (v_j)_{s'} X_s X_{s'} + (v_j)_s (v_i)_{s'} X_s X_{s'}\big) \\ &= \frac{1}{\sqrt{\alpha_i \alpha_j}} \sum_{s,s'=1}^m (v_i)_s (v_j)_{s'} (X_s X_{s'} + X_{s'} X_s) \\ &= \frac{1}{\sqrt{\alpha_i \alpha_j}} \sum_{s,s'=1}^m (v_i)_s (v_j)_{s'} 2Q_{s,s'} I = \frac{2}{\sqrt{\alpha_i \alpha_j}} v_i^* Q v_j I = 2\delta_{i,j} I, \end{aligned}$$

which means that we have the representation $\pi_A \colon \mathcal{C}(r) \to \mathbb{C}\langle A_1, \ldots, A_r\rangle$ defined by $\pi_A(a_i) = A_i$, where the a_i are the generators of $\mathcal{C}(r)$. In the same way we can define matrices $B_1, \ldots, B_r$ by taking linear combinations of the matrices Y_t so that we obtain the representation $\pi_B \colon \mathcal{C}(r) \to \mathbb{C}\langle B_1, \ldots, B_r\rangle$ defined by $\pi_B(a_i) = B_i$.

By assumption, the algebras $\mathbb{C}\langle X_1, \ldots, X_m\rangle$ and $\mathbb{C}\langle Y_1, \ldots, Y_n\rangle$ commute. This implies that the algebras $\mathbb{C}\langle A_1, \ldots, A_r\rangle$ and $\mathbb{C}\langle B_1, \ldots, B_r\rangle$ also commute and that $\mathbb{C}\langle A_1, \ldots, A_r\rangle\mathbb{C}\langle B_1, \ldots, B_r\rangle$ is an algebra. Moreover, we have

$$[\pi_A(a), \pi_B(b)] = \pi_A(a)\pi_B(b) - \pi_A(a)\pi_B(b) = 0 \quad \text{for all} \quad a, b \in \mathcal{C}(r).$$

By the universal property of the tensor product of algebras (see, e.g., [Kas95, Proposition II.4.1]), there exists a (unique) algebra homomorphism

$$\pi : \mathcal{C}(r) \otimes \mathcal{C}(r) \to \mathbb{C}\langle A_1, \dots, A_r \rangle \mathbb{C}\langle B_1, \dots, B_r \rangle$$

such that $\pi(a \otimes 1) = \pi_A(a)$ and $\pi(1 \otimes a) = \pi_B(a)$ for all $a \in \mathcal{C}(r)$. Moreover, each finite-dimensional, irreducible representation of a tensor product of algebras is the tensor product of two irreducible representations of those algebras (see, e.g., [EGH$^+$11, Rem. 2.27]). This means that each irreducible representation of $\mathcal{C}(r) \otimes \mathcal{C}(r)$ is the tensor product of two irreducible representations of $\mathcal{C}(r)$. Since irreducible representations of $\mathcal{C}(r)$ have size at least $2^{\lfloor r/2 \rfloor}$, it follows that irreducible representations of the tensor product $\mathcal{C}(r) \otimes \mathcal{C}(r)$ must have size at least $(2^{\lfloor r/2 \rfloor})^2$. Since π is a representation of $\mathcal{C}(r) \otimes \mathcal{C}(r)$, this means that the matrices A_i and B_j must have size at least $(2^{\lfloor r/2 \rfloor})^2$, which shows $d \geq (2^{\lfloor r/2 \rfloor})^2$. □

Corollary 6.17. *Let C be an extreme point of* $\mathrm{Cor}(m, n)$ *and let* $r = \mathrm{rank}(C)$. *The minimum local dimension of a tensor operator representation of C is* $2^{\lfloor r/2 \rfloor}$.

Proof. The proof follows directly from Corollary 6.14 and Theorem 6.16. □

6.4 Matrices with high completely positive semidefinite rank

In this section we prove our main result and construct completely positive semidefinite matrices with exponentially large cpsd-rank. In order to do so we first explain the connection between bipartite correlation matrices and bipartite quantum correlations in $C_q(\{0,1\}^2 \times [m] \times [n])$. We then obtain our main result by using this connection and the link between bipartite quantum correlations and completely positive semidefinite matrices that we have explained in Section 3.4: for $\Gamma = A \times B \times S \times T$, the set $C_q(\Gamma)$ of bipartite quantum correlations can be obtained as a projection of $\mathrm{CS}_+^{(A\times S)\sqcup(B\times T)} \cap \mathcal{L}$ onto the coordinates indexed by $A \times S$ and $B \times T$, where $\mathcal{L}$ is an appropriate affine subspace (3.18). To be more concrete, let us repeat the construction of the completely positive semidefinite matrix associated to a bipartite quantum correlation $P = (P(a, b|s, t)) \in \mathbb{R}^\Gamma$ that we have seen in the proof of Proposition 3.7.

As we have shown there, we may assume that P has a tensor operator representation of the form $\{\psi = \sum_{i=1}^d \sqrt{\lambda_i}\, v_i \otimes v_i, \{E_s^a\}, \{F_t^b\}\}$. We then define the matrices

$$K = \sum_{i=1}^d \sqrt{\lambda_i}\, v_i v_i^*, \quad X_s^a = K^{1/2} E_s^a K^{1/2}, \quad Y_t^b = K^{1/2} (F_t^b)^T K^{1/2}.$$

The completely positive semidefinite matrix associated to P is then the Gram matrix of the Hermitian positive semidefinite matrices X_s^a and Y_t^b. Indeed, notice that

$\mathrm{vec}(K) = \psi$. Moreover, we have the identity

$$\begin{aligned} P(a,b|s,t) = \mathrm{vec}(K)^*(E_s^a \otimes F_t^b)\mathrm{vec}(K) &= \mathrm{Tr}(KE_s^a K(F_t^b)^T) \\ &= \mathrm{Tr}(K^{1/2}E_s^a K^{1/2}K^{1/2}(F_t^b)^T K^{1/2}) \\ &= \mathrm{Tr}(X_s^a Y_t^b), \end{aligned}$$

where we use the identity (3.21) in the first equality. In a similar way one can verify that M satisfies the linear constraints (3.18).

We now show how to construct from a bipartite correlation $C \in \mathrm{Cor}(m,n)$ a quantum correlation $P = (P(a,b|s,t)) \in \mathbb{R}^\Gamma$, where $\Gamma = \{0,1\} \times \{0,1\} \times [m] \times [n]$. This quantum correlation P has the property that the smallest local dimension in which P can be realized is lower bounded by the smallest local dimension of a tensor representation of C.

Lemma 6.18. *Let $C \in \mathrm{Cor}(m,n)$ and assume C admits a tensor operator representation in local dimension d, but does not admit a tensor operator representation in smaller dimension. Then there exists a quantum correlation P defined on $\{0,1\} \times \{0,1\} \times [m] \times [n]$, satisfying the relations*

$$C(s,t) = P(0,0|s,t) + P(1,1|s,t) - P(0,1|s,t) - P(1,0|s,t) \text{ for } s \in [m], t \in [n], \tag{6.13}$$

that can be realized in local dimension d, but cannot be realized in smaller dimension.

Proof. We first show the existence of a quantum correlation that satisfies (6.13). Let $C \in \mathrm{Cor}(m,n)$. By assumption there exists a unit vector $\psi \in \mathbb{C}^d \otimes \mathbb{C}^d$, and Hermitian $d \times d$ matrices $X_1, \ldots, X_m, Y_1, \ldots, Y_n$, whose spectra are contained in $[-1,1]$, such that $C_{s,t} = \psi^*(X_s \otimes Y_t)\psi$ for all s and t. We define the Hermitian positive semidefinite matrices

$$X_s^a = \frac{I + (-1)^a X_s}{2}, \ Y_t^b = \frac{I + (-1)^b Y_t}{2} \quad \text{for } a,b \in \{0,1\}. \tag{6.14}$$

Using the fact that $X_s^0 + X_s^1 = Y_t^0 + Y_t^1 = I$, $X_s = X_s^0 - X_s^1$, and $Y_t = Y_t^0 - Y_t^1$, it follows that the function $P(a,b|s,t) = \psi^*(X_s^a \otimes Y_t^b)\psi$ is a quantum correlation that can be realized in local dimension d and satisfies (6.13).

Assume that P can be realized in dimension k. We show that $k \geq d$. As P is realizable in dimension k, there exist a unit vector $\widetilde{\psi} \in \mathbb{C}^k \otimes \mathbb{C}^k$ and Hermitian positive semidefinite $k \times k$ matrices $\{\widetilde{X}_s^a\}$ and $\{\widetilde{Y}_t^b\}$ such that

$$\sum_{a \in \{0,1\}} \widetilde{X}_s^a = \sum_{b \in \{0,1\}} \widetilde{Y}_t^b = I \quad \text{for all} \quad s \in S, t \in T,$$

for which we have $P(a,b|s,t) = \widetilde{\psi}^*(\widetilde{X}_s^a \otimes \widetilde{Y}_t^b)\widetilde{\psi}$. Observe that the spectrum of the operators $\widetilde{X}_s^a$ and $\widetilde{Y}_t^b$ is contained in $[0,1]$. We define $\widetilde{X}_s = \widetilde{X}_s^0 - \widetilde{X}_s^1, \widetilde{Y}_t = \widetilde{Y}_t^0 - \widetilde{Y}_t^1$. Then, using (6.13), we can conclude

$$C_{s,t} = \widetilde{\psi}^*(\widetilde{X}_s \otimes \widetilde{Y}_t)\widetilde{\psi}.$$

This means that C has a tensor operator representation in local dimension k and thus, by the assumption of the lemma, $k \geq d$. □

We can now prove our main theorem:

Theorem 6.19. *For each positive integer k, there exists a completely positive semidefinite matrix M of size $4k^2 + 2k + 2$ with* $\text{cpsd-rank}_{\mathbb{C}}(M) = 2^k$.

Proof. Let k be a positive integer, let $r = 2k$, and set $n = \binom{r}{2} + 1$. By Theorem 6.12(i) there exists an extreme point C of $\mathrm{Cor}(r, n)$ with $\mathrm{rank}(C) = r$. Corollary 6.17 tells us there exists a tensor operator representation of C using local dimension $d = 2^{\lfloor r/2 \rfloor} = 2^k$, and there does not exist a smaller tensor operator representation. Then, by Lemma 6.18, there exists a quantum correlation $P\colon \{0,1\} \times \{0,1\} \times [r] \times [n] \to [0,1]$ that can be realized in local dimension d and not in smaller dimension. Let M be a completely positive semidefinite matrix constructed from P as indicated in Theorem 3.6 (see also the construction at the beginning of this section), so that $\text{cpsd-rank}_{\mathbb{C}}(M) = d$ and the size of M is $2r + 2n = r^2 + r + 2 = 4k^2 + 2k + 2$. □

We note that by using Theorem 6.12(ii) we would get a matrix with the same completely positive semidefinite rank 2^k, but with larger size $4k^2+6k+2$. Likewise, the result of [Ji13] combined with Theorem 3.6 also leads to a matrix with the same completely positive semidefinite rank, but with larger size $(148k^2 - 58k)$. It is an open problem to find an explicit family of completely positive semidefinite matrices where the ratio of the completely positive semidefinite rank to the matrix size is larger than in the above theorem. It is not possible to obtain such an improved family by the above method. Indeed, if M is a completely positive semidefinite matrix with $\text{cpsd-rank}_{\mathbb{C}}(M) = 2^k$, constructed from an extreme bipartite correlation matrix $C \in \mathrm{Cor}(m, n)$ as in the above theorem, then the size $2m + 2n$ of M is at least $4k^2 + 2k + 2$. To see this, note that, by Corollary 6.17 and the results in this section, C has to have rank $2k$. Then, by Tsirelson's bound, $m + n - 1 \geq \binom{2k+1}{2}$ and therefore $2m + 2n \geq 4k^2 + 2k + 2$.

6.5 Related work

Upon completion of the work in this chapter we learned of the simultaneous independent work [PSVW18], where a class of matrices with exponential cpsd-rank is also constructed. The key idea of using extremal bipartite correlation matrices having large rank is the same. Our construction uses bipartite correlation matrices with optimized parameters meeting Tsirelson's upper bound (6.6) (see Corollary 6.11 and Theorem 6.12). As a consequence, our completely positive semidefinite matrices have the best ratio between cpsd-rank and size that can be obtained using this technique.

Finally, the subsequent work [PV18] gives a more direct proof of the main result of [PSVW18] and [GdLL17], avoiding the language of quantum correlations.

Chapter 7

Average entanglement dimension

This chapter is based on the paper "Bounds on entanglement dimensions and quantum graph parameters via noncommutative polynomial optimization", by S. Gribling, D. de Laat, and M. Laurent [GdLL18].

In this chapter we continue to study bipartite quantum correlations, but now not with the goal of constructing correlations which need a large amount of entanglement as in the previous chapter, but with the goal of finding a good way to quantify the minimal amount of entanglement necessary to realize a given correlation. The study of this topic was initiated in [BPA$^+$08] and continued, e.g., in [PV08, WCD08, SVW16].

We propose and study a new measure for the amount of entanglement needed to realize a bipartite quantum correlation. Let us briefly recall the definition of a bipartite quantum correlation and the measure for the amount of entanglement that we have seen in Chapters 3 and 6. Let $\Gamma = A \times B \times S \times T$ for some finite sets A, B, S, and T. Recall that $P \in \mathbb{R}^\Gamma$ is called a *bipartite quantum correlation realizable in the tensor model in dimension* d if there exist POVMs $\{E_s^a\}_{a\in A} \subseteq \mathrm{H}_+^d$ and $\{F_t^b\}_{b\in B} \subseteq \mathrm{H}_+^d$, and a unit vector $\psi \in \mathbb{C}^d \otimes \mathbb{C}^d$ such that (3.2) holds, that is,

$$P(a, b|s, t) = \mathrm{Tr}((E_s^a \otimes F_t^b)\psi\psi^*) = \psi^*(E_s^a \otimes F_t^b)\psi \qquad \text{for all } (a, b, s, t) \in \Gamma.$$

The set of quantum correlations realizable in dimension d is denoted by $C_q^d(\Gamma)$ and the *entanglement dimension* $D_q(P)$ is defined by (3.3):

$$D_q(P) = \min\bigl\{d^2 : d \in \mathbb{N},\, P \in C_q^d(\Gamma)\bigr\}.$$

It might seem artificial that we consider here d^2 instead of d. We do so to remain consistent with the second model of quantum correlations, the commuting operator model. We say that $P \in \mathbb{R}^\Gamma$ is a *bipartite quantum correlation realizable in the commuting operator model in dimension* d if there exist POVMs $\{X_s^a\}_{a\in A} \subseteq \mathrm{H}_+^d$

and $\{Y_t^b\}_{b\in B} \subseteq \mathrm{H}_+^d$ and a unit vector $\psi \in \mathbb{C}^d$ such that

$$P(a,b|s,t) = \mathrm{Tr}((X_s^a Y_t^b)\psi\psi^*) = \psi^*(X_s^a Y_t^b)\psi \qquad \text{for all } (a,b,s,t) \in \Gamma,$$

and, most importantly, the POVMs $\{X_s^a\}_{a\in A} \subseteq \mathrm{H}_+^d$ and $\{Y_t^b\}_{b\in B} \subseteq \mathrm{H}_+^d$ commute (i.e., $X_s^a Y_t^b = Y_t^b X_s^a$ for all a,b,s,t). The set of bipartite quantum correlations that are realizable in the commuting operator model in dimension d is denoted $C_{qc}^d(\Gamma)$ and we analogously define

$$D_{qc}(P) = \min\big\{d : d \in \mathbb{N},\ P \in C_{qc}^d(\Gamma)\big\}.$$

It now becomes clear why we defined $D_q(P)$ using the Hilbert space dimension d^2 instead of the local dimension d: by setting $X_s^a = E_s^a \otimes I_d$ and $Y_t^b = I_d \otimes F_t^b$, we see that $D_{qc}(P) \leq D_q(P)$ for all $P \in C_q(\Gamma)$. As we will show, d^2 is the 'right' choice: for extreme points of the set $C_q(\Gamma)$ we have $D_{qc}(P) = D_q(P)$.

So far, we have recalled the entanglement dimension $D_q(P)$ that was used extensively in the previous Chapter 6 to measure the amount of entanglement needed to realize a quantum correlation. Let us now propose a new measure that corresponds to the setting where shared randomness is considered free.

When we allow the two parties free access to shared randomness, it becomes more natural to measure the amount of entanglement used not just by the total dimension d, but by, for instance, the maximum dimension or the average dimension over the realizations of the shared random variables. From a geometric perspective the maximum would correspond to finding the smallest d such that $P \in \mathrm{conv}(C_q^d(\Gamma))$. This seems to be a parameter that is not easy to compute and that is why we propose to study the *average entanglement dimension* instead:

$$A_q(P) = \inf\Big\{\sum_{i=1}^I \lambda_i D_q(P_i) : I \in \mathbb{N}, \lambda \in \mathbb{R}_+^I,\ \sum_{i=1}^I \lambda_i = 1,\ P = \sum_{i=1}^I \lambda_i P_i,\ P_i \in C_q(\Gamma)\Big\}.$$

Here $I \in \mathbb{N}, \lambda \in \mathbb{R}_+^I$ and the P_i's are the variables. Our main results are as follows: We show that the average entanglement dimension equals 1 if and only if the bipartite correlation is classical, we show that the parameter does not change if we choose the commuting operator model instead of the tensor model, and, most importantly, we show that there is a hierarchy of semidefinite programming lower bounds.

7.1 Our results

Let us now give a more formal overview of the results in this chapter.

We are interested in the minimal entanglement dimension needed to realize a given correlation $P \in C_q(\Gamma)$. If P is deterministic or only uses local randomness, then $D_q(P) = D_{qc}(P) = 1$. But other classical correlations (which use shared randomness) have $D_q(P) \geq D_{qc}(P) > 1$, which means the shared quantum state is used as a shared randomness resource. In [BPA$^+$08] the concept of dimension witness is introduced. A *d-dimensional witness* is defined as a halfspace containing

$\mathrm{conv}(C_q^d(\Gamma))$, but not the full set $C_q(\Gamma)$. As a measure of entanglement this suggests the parameter

$$\inf\Big\{\max_{i\in[I]} D_q(P_i) : I \in \mathbb{N},\, \lambda \in \mathbb{R}_+^I,\, \sum_{i=1}^I \lambda_i = 1,\, P = \sum_{i=1}^I \lambda_i P_i,\, P_i \in C_q(\Gamma)\Big\}. \tag{7.1}$$

Here $I \in \mathbb{N}, \lambda \in \mathbb{R}_+^I$ and the P_i's are the variables. Observe that, for a bipartite correlation P, this parameter is equal to 1 if and only if P is classical. Hence, it more closely measures the minimal entanglement dimension when the parties have free access to shared randomness.

Let us now give an operational interpretation of (7.1). Before, in Section 3.3 we have seen that bipartite (quantum) correlations can be used as strategies in a nonlocal game. There we said that the objective of a nonlocal game is to maximize some linear function over the space of bipartite (quantum) correlations. In the previous Chapter 6 we encountered the situation where there was a unique optimal quantum strategy. It thus makes sense to think of realizing a given quantum correlation P as a nonlocal game where two players receive questions (s,t) and have to produce answers (a,b) with the 'correct' probability $P(a,b|s,t)$. The objective of the two players is to use as little entanglement as possible. When we measure the amount of entanglement using the local dimension, the optimal value of the game would equal $D_q(P)$.

In this language, Equation (7.1) corresponds to an entanglement measure where shared randomness is free: Before the game starts the parties may select a finite number of pure states ψ_i $(i \in I)$ (instead of a single one), in possibly different dimensions d_i, and POVMs $\{E_s^a(i)\}_a$, $\{F_t^b(i)\}_b$ for each $i \in I$ and $(s,t) \in S \times T$. As before, we assume that the parties cannot communicate after receiving their questions (s,t), but now they do have access to shared randomness, which they use to decide on which state ψ_i to use. The parties proceed to measure state ψ_i using POVMs $\{E_s^a(i)\}_a$, $\{F_t^b(i)\}_b$, so that the probability of answers (a,b) is given by the quantum correlation P_i. Equation (7.1) then asks for the largest dimension needed in order to generate P when access to shared randomness is free.

It is not clear how to compute (7.1). Here we propose a variation of (7.1), and we provide a hierarchy of semidefinite programs that converges to this variation under flatness. Instead of considering the largest dimension needed to generate P, we consider the *average* dimension. That is, we minimize $\sum_{i\in I} \lambda_i D_q(P_i)$ over all convex combinations $P = \sum_{i\in I} \lambda_i P_i$. Hence, the *minimal average entanglement dimension* is defined by

$$A_q(P) = \inf\Big\{\sum_{i=1}^I \lambda_i D_q(P_i) : I \in \mathbb{N},\, \lambda \in \mathbb{R}_+^I,\, \sum_{i=1}^I \lambda_i = 1,\, P = \sum_{i=1}^I \lambda_i P_i,\, P_i \in C_q(\Gamma)\Big\} \tag{7.2}$$

in the tensor model. Here $I \in \mathbb{N}, \lambda \in \mathbb{R}_+^I$ and the P_i's are the variables. In the commuting model, the parameter $A_{qc}(P)$ is defined by the same expression with $D_q(P_i)$ being replaced by $D_{qc}(P_i)$. Observe that we need not replace $C_q(\Gamma)$ by

$C_{qc}(\Gamma)$ since $D_{qc}(P) = \infty$ for any $P \in C_{qc}(\Gamma) \setminus C_q(\Gamma)$. Moreover, since $D_{qc}(P) \leq D_q(P)$ by (3.6), we have the inequality

$$A_{qc}(P) \leq A_q(P) \quad \text{for all} \quad P \in C_q(\Gamma). \tag{7.3}$$

It follows by convexity that for the above definitions it does not matter whether we use pure or mixed states. We will show in this chapter that for the average minimal entanglement dimension it also does not matter whether we use the tensor or commuting model.

Proposition 7.2. *For any $P \in C_q(\Gamma)$ we have $A_q(P) = A_{qc}(P)$.*

We have $A_q(P) \leq D_q(P)$ and $A_{qc}(P) \leq D_{qc}(P)$ for $P \in C_q(\Gamma)$, and it is easy to see that $A_q(P) = D_q(P)$ and $A_{qc}(P) = D_{qc}(P)$ holds if P is an extreme point of $C_q(\Gamma)$. The above proposition thus shows that $D_q(P) = D_{qc}(P)$ if P is an extreme point of $C_q(\Gamma)$.

Next, we show that the parameter $A_q(P)$ can be used to distinguish between classical and nonclassical correlations.

Proposition 7.3. *For $P \in C_q(\Gamma)$ we have $A_q(P) = 1$ if and only if $P \in C_{loc}(\Gamma)$.*

As mentioned before, there exist sets Γ for which $C_q(\Gamma)$ is not closed [Slo19, DPP19], which implies the existence of a sequence $\{P_i\}_{i \in \mathbb{N}} \subseteq C_q(\Gamma)$ such that $D_q(P) \to \infty$ as $i \to \infty$. We show this also implies the existence of such a sequence with $A_q(P_i) \to \infty$.

Proposition 7.4. *If $C_q(\Gamma)$ is not closed, then there exists a sequence $\{P_i\} \subseteq C_q(\Gamma)$ with $A_q(P_i) \to \infty$.*

Using tracial polynomial optimization we construct a hierarchy $\{\xi_r^{\mathrm{q}}(P)\}$ of lower bounds on $A_{qc}(P)$. For each $r \in \mathbb{N}$ this is a semidefinite program, and for $r = \infty$ it is an infinite-dimensional semidefinite program. We further define a variation $\xi_*^{\mathrm{q}}(P)$ of $\xi_\infty^{\mathrm{q}}(P)$ by adding a constraint that the matrix variable has to have finite rank, so that

$$\xi_1^{\mathrm{q}}(P) \leq \xi_2^{\mathrm{q}}(P) \leq \ldots \leq \xi_\infty^{\mathrm{q}}(P) \leq \xi_*^{\mathrm{q}}(P) \leq A_{qc}(P).$$

We do not know whether $\xi_\infty^{\mathrm{q}}(P) = \xi_*^{\mathrm{q}}(P)$ always holds. First we show that we imposed enough constraints in the bounds $\xi_r^{\mathrm{q}}(P)$ so that $\xi_*^{\mathrm{q}}(P) = A_{qc}(P)$.

Proposition 7.5. *For any $P \in C_q(\Gamma)$ we have $\xi_*^{\mathrm{q}}(P) = A_{qc}(P)$.*

Then we show that the infinite-dimensional semidefinite program $\xi_\infty^{\mathrm{q}}(P)$ is the limit of the finite-dimensional semidefinite programs.

Proposition 7.6. *For any $P \in C_q(\Gamma)$ we have $\xi_r^{\mathrm{q}}(P) \to \xi_\infty^{\mathrm{q}}(P)$ as $r \to \infty$.*

Finally we give a flatness criterion under which finite convergence $\xi_r^{\mathrm{q}}(P) = \xi_*^{\mathrm{q}}(P)$ holds, this criterion is easy to check given a solution to $\xi_r^{\mathrm{q}}(P)$.

Proposition 7.7. *If $\xi_r^{\mathrm{q}}(P)$ admits a $(\lceil r/3 \rceil + 1)$-flat optimal solution, then we have $\xi_r^{\mathrm{q}}(P) = \xi_*^{\mathrm{q}}(P)$.*

Before proving the above propositions, let us revisit a small example to illustrate our results.

Example 7.1. As we have seen in Chapter 3, a Bell inequality is an inequality that is valid for the set of classical correlations, but that can be violated by a quantum correlation. Recall that the set of classical bipartite correlations $C_{loc}(\Gamma)$ equals $\mathrm{conv}(C_q^1(\Gamma))$. Therefore, in the language of Brunner et al. [BPA+08], a Bell inequality forms a 1-*dimensional witness*. If a quantum correlation violates a Bell inequality we know that its entanglement dimension is at least 2. What about its average entanglement dimension? By Proposition 7.3 such a correlation would also have average entanglement dimension strictly larger than 1.

One can say a bit more when the violation is large. To be concrete, let us revisit the Clauser-Horne-Shimony-Holt game (CHSH) that we have described in Section 3.3.2. There we have seen a Bell inequality consisting of a nonlocal game for which the maximum winning probability using classical strategies equals 3/4, while the maximum winning probability using quantum strategies equals $\frac{1}{2}+\frac{1}{2\sqrt{2}}$. We have seen a strategy that achieves the maximum quantum winning probability and has entanglement dimension equal to 2. Can violations of the CHSH inequality be used to quantify the average entanglement dimension? As we argue below, this is indeed the case. If a quantum strategy P has a winning probability of the form

$$(1-\lambda)\cdot\frac{3}{4}+\lambda\cdot(\frac{1}{2}+\frac{1}{2\sqrt{2}}), \tag{7.4}$$

for some $0 \leq \lambda \leq 1$, then its average entanglement dimension is at least $1+\lambda$. Thus, violations of the CHSH inequality form *average entanglement dimension witnesses*. In particular, an optimal quantum strategy for the CHSH game ($\lambda = 1$) has average entanglement dimension equal to 2.

Let us now show that violations of the CHSH inequality indeed form average entanglement dimension witnesses. Let P be a quantum correlation whose winning probability is given by Equation (7.4) for some $0 \leq \lambda \leq 1$. Consider an arbitrary convex decomposition $P = \sum_{i\in I}\lambda_i P_i$ with $P_i \in C_q(\Gamma)$, as in the definition of $A_q(P)$. Let I_q be the subset of I corresponding to the non-classical correlations P_i, and set $\lambda_0 := \sum_{i\in I_q}\lambda_i$. Using the linearity of the CHSH inequality, and the maximum classical and quantum winning probabilities, we see that the winning probability of P is at most $(1-\lambda_0)\cdot\frac{3}{4}+\lambda_0\cdot(\frac{1}{2}+\frac{1}{2\sqrt{2}})$. It thus follows that if the winning probability of P equals Equation (7.4), then in any convex decomposition of P we must have that $\lambda_0 \geq \lambda$. Since any quantum correlation has entanglement dimension at least 2 this shows that the average entanglement dimension of P is at least $(1-\lambda)\cdot 1+\lambda\cdot 2 = 1+\lambda$. Therefore, a winning probability of at least $(1-\lambda)\cdot\frac{3}{4}+\lambda\cdot(\frac{1}{2}+\frac{1}{2\sqrt{2}})$ provides an *average entanglement witness*: $A_q(P) \geq 1+\lambda$. △

7.2 Some properties of the average entanglement dimension

Here we investigate some properties of the average entanglement dimension $A_q(\cdot)$. We start by showing that it does not matter whether we use the tensor model or the commuting model.

Proposition 7.2. *For any $P \in C_q(\Gamma)$ we have $A_q(P) = A_{qc}(P)$.*

Proof. The inequality $A_{qc}(P) \leq A_q(P)$ was observed in (7.3). For the reverse inequality assume we have a convex decomposition $P = \sum_{i=1}^I \lambda_i P_i$, which is feasible for $A_{qc}(P)$. This means that we have POVMs $\{X_s^a(i)\}_a$ and $\{Y_t^b(i)\}_b$ in $\mathbb{C}^{d_i \times d_i}$ with and unit vectors $\psi_i \in \mathbb{C}^{d_i}$ such that for all $(a,b,s,t) \in \Gamma$ and $i \in [I]$ we have $[X_s^a(i), Y_t^b(i)] = 0$ and $P_i(a,b|s,t) = \psi_i^* X_s^a(i) Y_t^b(i) \psi_i$. We will construct another decomposition of P which will provide a feasible solution to $A_q(P)$ with value at most $\sum_i \lambda_i d_i$.

Fix some index $i \in [I]$. Applying Theorem 4.2 to $\mathbb{C}\langle\{X_s^a(i)\}_{a,s}\rangle$, the matrix $*$-algebra $\mathbb{C}\langle\{X_s^a(i)\}_{a,s}\rangle$ generated by the matrices $X_s^a(i)$ for $(a,s) \in A \times S$, shows that there exist a unitary matrix U_i and integers[1] K_i, m_k, n_k such that

$$U_i \mathbb{C}\langle\{X_s^a(i)\}_{a,s}\rangle U_i^* = \bigoplus_{k=1}^{K_i} (\mathbb{C}^{n_k \times n_k} \otimes I_{m_k}) \quad \text{and} \quad d_i = \sum_{k=1}^{K_i} m_k n_k.$$

By assumption each matrix $Y_t^b(i)$ commutes with all the matrices in the algebra $\mathbb{C}\langle\{X_s^a(i)\}_{a,s}\rangle$, and thus $U_i Y_t^b(i) U_i^*$ lies in the algebra $\bigoplus_k (I_{n_k} \otimes \mathbb{C}^{m_k \times m_k})$. Hence, we may assume

$$X_s^a(i) = \bigoplus_{k=1}^{K_i} E_s^a(i,k) \otimes I_{m_k}, \quad Y_t^b(i) = \bigoplus_{k=1}^{K_i} I_{n_k} \otimes F_t^b(i,k), \quad \psi_i = \bigoplus_{k=1}^{K_i} \psi_{i,k},$$

with $E_s^a(i,k) \in \mathbb{C}^{n_k \times n_k}$, $F_t^b(i,k) \in \mathbb{C}^{m_k \times m_k}$, and $\psi_{i,k} \in \mathbb{C}^{n_k} \otimes \mathbb{C}^{m_k}$. Then we have

$$P_i(a,b|s,t) = \mathrm{Tr}(X_s^a(i) Y_t^b(i) \psi_i \psi_i^*) = \sum_{k=1}^{K_i} \|\psi_{i,k}\|^2 \underbrace{\mathrm{Tr}\bigg(E_s^a(i,k) \otimes F_t^b(i,k) \frac{\psi_{i,k}\psi_{i,k}^*}{\|\psi_{i,k}\|^2} \bigg)}_{Q_{i,k}(a,b|s,t)},$$

where $Q_{i,k} \in C_q(\Gamma)$. As $\sum_k \|\psi_{i,k}\|^2 = \|\psi_i\|^2 = 1$, we have that $P_i = \sum_k \|\psi_{i,k}\|^2 Q_{i,k}$ is a convex combination of the $Q_{i,k}$'s.

We now show that $Q_{i,k} \in C_q^{\min\{m_k,n_k\}}(\Gamma)$. Consider the Schmidt decomposition

$$\frac{\psi_{i,k}}{\|\psi_{i,k}\|} = \sum_{\ell=1}^{\min\{m_k,n_k\}} \lambda_{i,k,\ell}\, v_{i,k,\ell} \otimes w_{i,k,\ell},$$

where $\lambda_{i,k,\ell} \geq 0$, and $\{v_{i,k,\ell}\}_{\ell=1}^{n_k} \subseteq \mathbb{C}^{n_k}$ and $\{w_{i,k,\ell}\}_{\ell=1}^{m_k} \subseteq \mathbb{C}^{m_k}$ are orthonormal bases.[2] Define unitary matrices $V_k \in \mathbb{C}^{n_k \times n_k}$ and $W_k \in \mathbb{C}^{m_k \times m_k}$ such that $V_k v_{i,k,\ell}$ is the ℓth unit vector in $\mathbb{R}^{n_k}$ for $1 \leq \ell \leq n_k$ and $W_k w_{i,k,\ell}$ is the ℓth unit vector

[1] We omit the explicit dependence on i in the integers m_k, n_k to simplify the notation.

[2] For convenience we recall here Footnote 9 of Chapter 3. The Schmidt decomposition $\psi = \sum_{i=1}^d \sqrt{\lambda_i}\, u_i \otimes v_i$ of $\psi \in \mathbb{C}^d \otimes \mathbb{C}^d$ can be viewed as the singular value decomposition $\sum_{i=1}^d \sqrt{\lambda_i} u_i v_i^*$ of the matrix $A \in \mathbb{C}^{d \times d}$ for which $\psi = \sum_{i,j=1}^d A_{ij} e_i \otimes e_j$.

in $\mathbb{R}^{m_k}$ for $1 \leq \ell \leq m_k$. Let $E_s^a(i,k)'$ (resp., $F_t^b(i,k)'$) be the leading principal submatrices of $V_k E_s^a(i,k) V_k^*$ (resp., $W_k F_t^b(i,k) W_k^*$) of size $\min\{m_k, n_k\}$. Moreover, set $\phi_{i,k} = \sum_{\ell=1}^{\min\{m_k,n_k\}} \lambda_{i,k,\ell}\, e_\ell \otimes e_\ell$, where e_ℓ is the ℓth unit vector in $\mathbb{R}^{\min\{m_k,n_k\}}$. Then we hve

$$\begin{aligned} Q_{i,k}(a,b|s,t) &= \mathrm{Tr}\bigg(E_s^a(i,k) \otimes F_t^b(i,k) \frac{\psi_{i,k}\psi_{i,k}^*}{\|\psi_{i,k}\|^2} \bigg) \\ &= \sum_{\ell,\ell'=1}^{\min\{m_k,n_k\}} \lambda_{i,k,\ell}\lambda_{i,k,\ell'} (v_{i,k,\ell}^* E_s^a(i,k) v_{i,k,\ell'})(w_{i,k,\ell}^* F_t^b(i,k) w_{i,k,\ell'}) \\ &= \sum_{\ell,\ell'=1}^{\min\{m_k,n_k\}} \lambda_{i,k,\ell}\lambda_{i,k,\ell'} (e_\ell^* E_s^a(i,k)' e_{\ell'})(e_\ell^* F_t^b(i,k)' e_{\ell'}) \\ &= \mathrm{Tr}((E_s^a(i,k)' \otimes F_t^b(i,k)')\phi_{i,k}\phi_{i,k}^*), \end{aligned}$$

which shows $Q_{i,k} \in C_q^{\min\{m_k,n_k\}}(\Gamma)$.

Combining the convex decompositions $P = \sum_i \lambda_i P_i$ and $P_i = \sum_k \|\psi_{i,k}\|^2 Q_{i,k}$, we get the following convex decomposition $P = \sum_{i,k} \lambda_i \|\psi_{i,k}\|^2 Q_{i,k}$, from which we obtain that $A_q(P)$ is at most

$$\sum_{i,k} \lambda_i \|\psi_{i,k}\|^2 \min\{m_k,n_k\}^2 \leq \sum_{i,k} \lambda_i \min\{m_k,n_k\}^2 \leq \sum_{i,k} \lambda_i m_k n_k = \sum_i \lambda_i d_i. \quad \square$$

We now show that the parameter $A_q(\cdot)$ permits to characterize classical correlations.

Proposition 7.3. *For $P \in C_q(\Gamma)$ we have $A_q(P) = 1$ if and only if $P \in C_{loc}(\Gamma)$.*

Proof. If $P \in C_{loc}(\Gamma)$, then P can be written as a convex combination of deterministic correlations (which belong to $C_q^1(\Gamma)$), and thus $A_q(P) = 1$.

For the reverse implication, assume $A_q(P) = 1$. Then there exists a sequence of convex decompositions $P = \sum_{i \in I^\ell} \lambda_i^\ell P_i^\ell$ indexed by $\ell \in \mathbb{N}$, with $\{P_i^\ell\} \subseteq C_q(\Gamma)$ and $\lim_{\ell\to\infty} \sum_{i\in I^\ell} \lambda_\ell D_q(P_i^\ell) = 1$. Note that for finite $\ell \in \mathbb{N}$ we may have that $\sum_{i\in I^\ell} \lambda_i^\ell D_q(P_i^\ell) > 1$. Decompose the set I^ℓ as the disjoint union $I_-^\ell \cup I_+^\ell$, where $D_q(P_i^\ell) = 1$ for $i \in I_-^\ell$ and $D_q(P_i^\ell) > 1$ for $i \in I_+^\ell$. Let $\varepsilon > 0$. Then, for all sufficiently large ℓ we have

$$\begin{aligned} 1 + \sum_{i\in I_+^\ell} \lambda_i &= \big(1 - \sum_{i\in I_+^\ell} \lambda_i^\ell\big) + 2\sum_{i\in I_+^\ell} \lambda_i^\ell \\ &\leq \sum_{i\in I_-^\ell} \lambda_i^\ell + \sum_{i\in I_+^\ell} \lambda_i^\ell D_q(P_i^\ell) \\ &= \sum_{i\in I^\ell} \lambda_\ell D_q(P_i^\ell) \\ &\leq 1 + \varepsilon, \end{aligned}$$

implying $\sum_{i\in I_+^\ell}\lambda_i^\ell \le \varepsilon$. This shows that the sequence $\mu^\ell := \sum_{i\in I_-^\ell}\lambda_i$ tends to 1 as $\ell \to \infty$. The correlation $P^\ell := \sum_{i\in I_-^\ell}\lambda_i^\ell P_i^\ell/\mu^\ell$ is a convex combination of deterministic correlations and thus it belongs to $C_{loc}(\Gamma)$. Moreover, since $C_{loc}(\Gamma)$ is a polytope (a closed set) and $P^\ell \to P$ as $\ell \to \infty$, we have that $P \in C_{loc}(\Gamma)$. □

As we have observed earlier, when the set $C_q(\Gamma)$ is not closed, the inclusion $C_q^d(\Gamma) \subseteq C_q(\Gamma)$ is strict for all d (because with a compactness argument one can show that $C_q^d(\Gamma)$ is closed), and thus there exists a sequence $\{P_i\} \subseteq C_q(\Gamma)$ with $D_q(P_i) \to \infty$ as $i \to \infty$. We show the analogous unboundedness property for the average entanglement dimension $A_q(\cdot)$. For the proof we will use the fact that also the sets $C_{qc}^d(\Gamma)$ are closed for all $d \in \mathbb{N}$.

Proposition 7.4. *If $C_q(\Gamma)$ is not closed, then there exists a sequence $\{P_i\} \subseteq C_q(\Gamma)$ with $A_q(P_i) \to \infty$.*

Proof. Assume for contradiction there exists an integer K such that $A_q(P) \le K$ for all $P \in C_q(\Gamma)$. We will show this results in a uniform upper bound K' on $D_{qc}(P)$, which, in view of (3.7), implies that $C_q(\Gamma)$ is equal to the closed set $C_{qc}^{K'}(\Gamma)$, contradicting the assumption that $C_q(\Gamma)$ is not closed. We claim that any $P \in C_q(\Gamma)$ belongs to $\mathrm{conv}(C_{qc}^K(\Gamma))$. Then, we can conclude the proof as follows. The extreme points of the compact convex set $\mathrm{conv}(C_{qc}^K(\Gamma))$ lie in $C_{qc}^K(\Gamma)$, so, by the Carathéodory theorem, any $P \in \mathrm{conv}(C_{qc}^K(\Gamma))$ is a convex combination of c elements from $C_{qc}^K(\Gamma)$, where $c = |\Gamma|+1-|S||T|$. By using a direct sum construction similar to Lemma 3.5 one can obtain $D_{qc}(P) \le cK$, which shows $K' := cK$ is a uniform upper bound on $D_{qc}(P)$ for all $P \in C_q(\Gamma)$.

It remains to prove the claim that any $P \in C_q(\Gamma)$ belongs to $\mathrm{conv}(C_{qc}^K(\Gamma))$. Towards that end, suppose that $P \in C_q(\Gamma) \setminus \mathrm{conv}(C_{qc}^K(\Gamma))$. We first observe that P can be decomposed as

$$P = \mu_1 R_1 + (1-\mu_1)Q_1, \tag{7.5}$$

where $R_1 \in C_q(\Gamma)$, $Q_1 \in \mathrm{conv}(C_{qc}^K(\Gamma))$, and $0 < \mu_1 \le K/(K+1)$. Indeed, by the assumption that $A_q(P) \le K$ and Proposition 7.2, we have $A_{qc}(P) = A_q(P) \le K$, so P can be written as a convex combination $P = \sum_{i\in I}\lambda_i P_i$ with $\{P_i\} \subseteq C_q(\Gamma)$ and $\sum_{i\in I}\lambda_i D_{qc}(P_i) \le K$. As $P \notin \mathrm{conv}(C_{qc}^K(\Gamma))$, the set J of indices $i \in I$ with $D_{qc}(P_i) \ge K+1$ is non empty. Then $(K+1)\sum_{i\in J}\lambda_i \le \sum_{i\in J}\lambda_i D_{qc}(P_i) \le K$, and thus $0 < \mu_1 := \sum_{i\in J}\lambda_i \le K/(K+1)$. Hence (7.5) holds after setting $R_1 = (\sum_{i\in J}\lambda_i P_i)/\mu_1$ and $Q_1 = (\sum_{i\in I\setminus J}\lambda_i P_i)/(1-\mu_1)$.

As $R_1 \in C_q(\Gamma)$, we have either $R_1 \in \mathrm{conv}(C_{qc}^K(\Gamma))$ or $R_1 \in C_q(\Gamma)\setminus\mathrm{conv}(C_{qc}^K(\Gamma))$. In the first case we have shown that $P \in \mathrm{conv}(C_{qc}^K(\Gamma))$. In the second case we may repeat the same argument for R_1. By iterating we obtain for each integer $k \in \mathbb{N}$ a decomposition

$$P = \mu_1\mu_2\cdots\mu_k R_k + \underbrace{(1-\mu_1)Q_1 + \mu_1(1-\mu_2)Q_2 + \ldots + \mu_1\mu_2\cdots\mu_{k-1}(1-\mu_k)Q_k}_{=(1-\mu_1\mu_2\cdots\mu_k)\widehat{Q}_k},$$

where $R_k \in C_q(\Gamma)$, $\widehat{Q}_k \in \mathrm{conv}(C_{qc}^K(\Gamma))$ and $\mu_1\mu_2\cdots\mu_k \le (K/(K+1))^k$. Then the sequence $\mu_1\mu_2\cdots\mu_k$ tends to 0 as $k \to \infty$. As the entries of R_k lie in $[0,1]$ we can conclude that $\mu_1\mu_2\cdots\mu_k R_k$ tends to 0 as $k \to \infty$. Hence the sequence $(\widehat{Q}_k)_k$ has a limit $\widehat{Q}$ and $P = \widehat{Q}$ holds. As all $\widehat{Q}_k$ lie in the compact set $\mathrm{conv}(C_{qc}^K(\Gamma))$, we also have $P \in \mathrm{conv}(C_{qc}^K(\Gamma))$. So we reach a contradiction with the assumption that $P \in C_q(\Gamma) \setminus \mathrm{conv}(C_{qc}^K(\Gamma))$, which shows that $C_q(\Gamma) \subseteq \mathrm{conv}(C_{qc}^K(\Gamma))$. □

7.3 A hierarchy of SDP lower bounds

We will now construct a hierarchy of lower bounds on the minimal average entanglement dimension, using its formulation via $A_{qc}(\cdot)$. Our approach is based on noncommutative polynomial optimization, similar to the approach we used in Chapter 5 for bounding matrix factorization ranks.

We first need some notation. Define the following sets of noncommutative variables

$$\mathbf{x} = \{x_s^a : (a,s) \in A \times S\} \quad \text{and} \quad \mathbf{y} = \{y_t^b : (b,t) \in B \times T\},$$

and let $\langle \mathbf{x}, \mathbf{y}, z\rangle$ be the set of all words in the $n = |S||A| + |T||B| + 1$ symbols x_s^a, y_t^b, and z.

The hierarchy of bounds on $A_{qc}(P)$ is based on the following idea: For any feasible solution to $A_{qc}(P)$, its objective value can be modeled as $L(1)$ for a certain tracial linear form L on the space of noncommutative polynomials (truncated to degree $2r$). Indeed, assume $\{(P_i, \lambda_i)_i\}$ is a feasible solution to the program defining $A_{qc}(P)$. That is, $P = \sum_i \lambda_i P_i$ with $\lambda_i \geq 0$, $\sum_i \lambda_i = 1$ and $P_i \in C_q(\Gamma)$. Assume $P_i(a,b|s,t) = \mathrm{Tr}\big(X_s^a(i) Y_t^b(i) \psi_i \psi_i^*\big)$, where $\psi_i \in \mathbb{C}^{d_i}$ and the POVM's $\{X_s^a(i)\}, \{Y_t^b(i)\} \subset \mathbb{C}^{d_i \times d_i}$ are such that for all $(a,b,s,t) \in \Gamma$ the matrices $X_s^a(i)$ and $Y_t^b(i)$ commute: $[X_s^a(i), Y_t^b(i)] = X_s^a(i)Y_t^b(i) - Y_t^b(i)X_s^a(i) = 0$. For $r \in \mathbb{N} \cup \{\infty\}$, consider the linear functional $L \in \mathbb{R}\langle \mathbf{x}, \mathbf{y}, z\rangle_{2r}^*$ defined by

$$L(p) = \sum_i \lambda_i \,\mathrm{Re}(\mathrm{Tr}(p(\mathbf{X}(i), \mathbf{Y}(i), \psi_i\psi_i^*))) \quad \text{for} \quad p \in \mathbb{R}\langle \mathbf{x}, \mathbf{y}, z\rangle_{2r}.$$

Here, for each index i, we set

$$\mathbf{X}(i) = (X_s^a(i) : (a,s) \in A \times S), \quad \mathbf{Y}(i) = (Y_t^b(i) : (b,t) \in B \times T),$$

and we replace the variables x_s^a, y_t^b, z by $X_s^a(i)$, $Y_t^b(i)$, and $\psi_i\psi_i^*$, respectively. First note that we have $L(1) = \sum_i \lambda_i d_i$. That is, $L(1)$ is equal to the objective value of the feasible solution $\{(P_i, \lambda_i)_i\}$ to $A_{qc}(P)$. Secondly, for all $(s,t,a,b) \in \Gamma$ we have $L(x_s^a y_t^b z) = P(a,b|s,t)$.

We will now identify several computationally tractable properties that this linear functional L satisfies. The hierarchy of lower bounds on $A_{qc}(P)$ then consists of optimization problems where we minimize $L(1)$ over the set of linear functionals that satisfy these properties.

First note that L is *symmetric*, that is, $L(w) = L(w^*)$ for all $w \in \langle \mathbf{x}, \mathbf{y}, z\rangle_{2r}$, and *tracial*, that is, $L(ww') = L(w'w)$ for all $w, w' \in \langle \mathbf{x}, \mathbf{y}, z\rangle$ with $\deg(ww') \leq 2r$.

Next, for all $p \in \mathbb{R}\langle \mathbf{x}, \mathbf{y}, z\rangle_{r-1}$ we have

$$L(p^* x_s^a p) = \sum_i \lambda_i \,\mathrm{Re}(\mathrm{Tr}(C(i)^* X_s^a(i) C(i)) \geq 0, \text{ where } C(i) = p(\mathbf{X}(i), \mathbf{Y}(i), \psi_i\psi_i^*),$$

as $C(i)^* X_s^a(i) C(i)$ is positive semidefinite since $X_s^a(i)$ is positive semidefinite. In the same way we have $L(p^* y_t^b p) \geq 0$ and $L(p^* z p) \geq 0$. That is, if we set

$$\mathcal{G} = \big\{x_s^a : s \in S,\, a \in A\big\} \cup \big\{y_t^b : t \in T,\, b \in B\big\} \cup \{z\},$$

then L is nonnegative (denoted as $L \geq 0$) on the truncated quadratic module $\mathcal{M}_{2r}(\mathcal{G})$. Similarly, setting

$$\mathcal{H} = \big\{z - z^2\big\} \cup \big\{1 - \sum_{a \in A} x_s^a : s \in S\big\} \cup \big\{1 - \sum_{b \in B} y_t^b : t \in T\big\} \cup \big\{[x_s^a, y_t^b] : (s,t,a,b) \in \Gamma\big\},$$

we have that $L = 0$ on the truncated ideal $\mathcal{I}_{2r}(\mathcal{H})$. Moreover, we have $L(z) = \sum_i \lambda_i \mathrm{Re}(\mathrm{Tr}(\psi_i \psi_i^*)) = 1$. In addition, for any matrices $U, V \in \mathbb{C}^{d_i \times d_i}$ we have

$$\psi_i \psi_i^* U \psi_i \psi_i^* V \psi_i \psi_i^* = \psi_i \psi_i^* V \psi_i \psi_i^* U \psi_i \psi_i^*,$$

and therefore, in particular,

$$L(wzuzvz) = L(wzvzuz) \text{ for all } u, v, w \in \langle \mathbf{x}, \mathbf{y}, z\rangle \text{ with } \deg(wzuzvz) \leq 2r.$$

That is, we have $L = 0$ on $\mathcal{I}_{2r}(\mathcal{R}_r)$, where

$$\mathcal{R}_r = \big\{zuzvz - zvzuz : u, v \in \langle \mathbf{x}, \mathbf{y}, z\rangle \text{ with } \deg(zuzvz) \leq 2r\big\},$$

where $r \in \mathbb{N} \cup \{\infty\}$. We get the idea of adding these last constraints from [NPA12], where this is used to study the mutually unbiased bases problem.

For $r \in \mathbb{N} \cup \{\infty\}$ we can now define the parameter:

$$\begin{aligned} \xi_r^{\mathrm{q}}(P) = \min\Big\{ L(1) : {} & L \in \mathbb{R}\langle \mathbf{x}, \mathbf{y}, z\rangle_{2r}^* \text{ tracial and symmetric}, \\ & L(z) = 1,\ L(x_s^a y_t^b z) = P(a,b|s,t) \text{ for all } (a,b,s,t) \in \Gamma, \\ & L \geq 0 \text{ on } \mathcal{M}_{2r}(\mathcal{G}),\ L = 0 \text{ on } \mathcal{I}_{2r}(\mathcal{H} \cup \mathcal{R}_r)\Big\}. \end{aligned}$$

Note that for order $r = 1$ we get the trivial bound $\xi_1^{\mathrm{q}}(P) = 1$.

Additionally, we define the parameter $\xi_*^{\mathrm{q}}(P)$ by adding to the definition of $\xi_\infty^{\mathrm{q}}(P)$ the constraint $\mathrm{rank}(M(L)) < \infty$. By construction this gives a hierarchy of lower bounds for $A_{qc}(P)$:

$$\xi_1^{\mathrm{q}}(P) \leq \ldots \leq \xi_r^{\mathrm{q}}(P) \leq \xi_\infty^{\mathrm{q}}(P) \leq \xi_*^{\mathrm{q}}(P) \leq A_{qc}(P).$$

Indeed, if $L \in \mathbb{R}\langle \mathbf{x}, \mathbf{y}, z\rangle_{2r}^*$ is feasible for $\xi_r^q(P)$ then its restriction to $\mathbb{R}\langle \mathbf{x}, \mathbf{y}, z\rangle_{2r-2}^*$ is feasible for $\xi_{r-1}^q(P)$, which implies $\xi_{r-1}^q(P) \leq L(1)$ and thus $\xi_{r-1}^q(P) \leq \xi_r^q(P)$.

7.3.1 Convergence results

We first show that the parameter $\xi_*^{\mathrm{q}}(P)$ coincides with the average entanglement dimension $A_q(P)$ and then we consider convergence properties of the bounds $\xi_r^{\mathrm{q}}(P)$ to the parameters $\xi_\infty^{\mathrm{q}}(P)$ and $\xi_*^{\mathrm{q}}(P)$.

Proposition 7.5. *For any $P \in C_q(\Gamma)$ we have $\xi_*^{\mathrm{q}}(P) = A_{qc}(P)$.*

Proof. We already know $\xi_*^{\mathrm{q}}(P) \leq A_{qc}(P)$. To show $\xi_*^{\mathrm{q}}(P) \geq A_{qc}(P)$ we let L be feasible for $\xi_*^{\mathrm{q}}(P)$, so that $L \geq 0$ on $\mathcal{M}(\mathcal{G})$, $L = 0$ on $\mathcal{I}(\mathcal{H} \cup \mathcal{R}_\infty)$ and $\mathrm{rank}(M(L)) < \infty$. We apply Theorem 4.6 to the scaled linear form $L/L(1)$ (note that $L(1) > 0$ since $L(z) = 1$): there exist finitely many scalars $\lambda_i \geq 0$ with $\sum_i \lambda_i = L(1)$, Hermitian matrix tuples $\mathbf{X}(i) = (X_s^a(i))_{a,s}$ and $\mathbf{Y}(i) = (Y_t^b(i))_{b,t}$, and Hermitian matrices Z_i, so that

$$g(\mathbf{X}(i), \mathbf{Y}(i), Z_i) \succeq 0 \ \text{ for all } g \in \mathcal{G}, \quad h(\mathbf{X}(i), \mathbf{Y}(i), Z_i) = 0 \ \text{ for all } h \in \mathcal{H} \cup \mathcal{R}_\infty, \tag{7.6}$$

and

$$L(p) = \sum_i \lambda_i \,\mathrm{Tr}(p(\mathbf{X}(i), \mathbf{Y}(i), Z_i)) \quad \text{for all} \quad p \in \mathbb{R}\langle \mathbf{x}, \mathbf{y}, z\rangle. \tag{7.7}$$

By Artin-Wedderburn theory (Theorem 4.2) we know that for each i there is a unitary matrix V_i such that $V_i \mathbb{C}\langle \mathbf{X}(i), \mathbf{Y}(i), Z_i\rangle V_i^* = \bigoplus_k \mathbb{C}^{d_k \times d_k} \otimes I_{m_k}$. Hence, after applying this further block diagonalization we may assume that in the decomposition (7.7), for each i, $\mathbb{C}\langle \mathbf{X}(i), \mathbf{Y}(i), Z_i\rangle$ is a full matrix algebra $\mathbb{C}^{d_i \times d_i}$.

Since $h(\mathbf{X}(i), \mathbf{Y}(i), Z_i) = 0$ for all $h \in R_\infty \cup \{z - z^2\}$, Z_i is a projector and the commutator $\big[Z_i u Z_i, Z_i v Z_i\big]$ vanishes for all $u, v \in \langle \mathbf{X}(i), \mathbf{Y}(i), Z_i\rangle$ and hence for all $u, v \in \mathbb{C}\langle \mathbf{X}(i), \mathbf{Y}(i), Z_i\rangle$. This means that $[Z_i T_1 Z_i, Z_i T_2 Z_i] = 0$ for all $T_1, T_2 \in \mathbb{C}^{d_i \times d_i}$. As Z_i is a projector, there exists a unitary matrix U_i such that $U_i Z_i U_i^* = \mathrm{Diag}(1, \ldots, 1, 0, \ldots, 0)$. The above then implies that for all T_1 and T_2, the leading principal submatrices of size $\mathrm{rank}(Z_i)$ of $U_i T_1 U_i^*$ and $U_i T_2 U_i^*$ commute. This implies $\mathrm{rank}(Z_i) \leq 1$ and thus $\mathrm{Tr}(Z_i) \in \{0, 1\}$. Let I be the set of indices with $\mathrm{Tr}(Z_i) = 1$. Then we have $\sum_{i \in I} \lambda_i = \sum_i \lambda_i \,\mathrm{Tr}(Z_i) = L(z) = 1$.

For each $i \in I$ define $P_i = (\mathrm{Tr}(X_s^a(i) Y_t^b(i) Z_i))$, which is a quantum correlation in $C_{qc}^{d_i}(\Gamma)$ because $\mathrm{Tr}(Z_i) = 1$, and $X_s^a, Y_t^b \succeq 0$ with $\sum_a X_s^a(i) = \sum_b Y_t^b(i) = I$ and $[X_s^a(i), Y_t^b(i)] = 0$ in view of (7.6). Using Equation (7.7) we obtain that $P = \sum_{i \in I} \lambda_i P_i$. Hence, $(P_i, \lambda_i)_{i \in I}$ forms a feasible solution to $A_{qc}(P)$ with objective value $\sum_{i \in I} \lambda_i D_{qc}(P_i) \leq \sum_{i \in I} \lambda_i d_i \leq \sum_i \lambda_i d_i = L(1)$. □

The problem $\xi_r^{\mathrm{q}}(P)$ differs in two ways from a standard tracial optimization problem. First it does not have the normalization $L(1) = 1$ (and instead it minimizes $L(1)$), and second it has ideal constraints $L = 0$ on $\mathcal{I}_{2r}(\mathcal{R}_r)$ where $\mathcal{R}_r$ depends on the relaxation order r. Nevertheless we can show that asymptotic convergence still holds.

Proposition 7.6. *For any $P \in C_q(\Gamma)$ we have $\xi_r^{\mathrm{q}}(P) \to \xi_\infty^{\mathrm{q}}(P)$ as $r \to \infty$.*

Proof. First observe that $1 - z^2$, $1 - (x_s^a)^2$, $1 - (y_t^b)^2 \in \mathcal{M}_4(\mathcal{G} \cup \mathcal{H}_0)$, where $\mathcal{H}_0$ contains the symmetric polynomials in $\mathcal{H}$; i.e., omitting the commutators $[x_s^a, y_t^b]$. Indeed, we have $1 - z^2 = (1 - z)^2 + 2(z - z^2)$ and

$$1 - (x_s^a)^2 = (1 - x_s^a)^2 + 2(1 - x_s^a) x_s^a (1 - x_s^a) + 2 x_s^a \big((1 - \sum_{a'} x_s^{a'}) + \sum_{a' \neq a} x_s^{a'}\big) x_s^a,$$

and the same for y_t^b. Hence $R - z^2 - \sum_{a,s} (x_s^a)^2 - \sum_{b,t} (y_t^b)^2 \in \mathcal{M}_4(\mathcal{G} \cup \mathcal{H}_0)$ for some $R > 0$. Fix $\varepsilon > 0$ and for each $r \in \mathbb{N}$ let L_r be feasible for $\xi_r^{\mathrm{q}}(P)$ with value

$L_r(1) \leq \xi_r^{\mathrm{q}}(P)+\varepsilon$. As L_r is tracial and zero on $\mathcal{I}_{2r}(\mathcal{H}_0)$, it follows (using the identity $p^*gp = pp^*g + [p^*g, p]$) that $L = 0$ on $\mathcal{M}_{2r}(\mathcal{H}_0)$. Hence, $L_r \geq 0$ on $\mathcal{M}_{2r}(\mathcal{G} \cup \mathcal{H}_0)$. Since $\sup_r L_r(1) \leq A_q(P) + \varepsilon$, we can apply Lemma 4.15 and conclude that $\{L_r\}_r$ has a converging subsequence; denote its limit by $L_\varepsilon \in \mathbb{R}\langle \mathbf{x}\rangle^*$. One can verify that L_ε is feasible for $\xi_\infty^{\mathrm{q}}(P)$, and $\xi_\infty^{\mathrm{q}}(P) \leq L_\varepsilon(1) \leq \lim_{r\to\infty} \xi_r^{\mathrm{q}}(P) + \varepsilon \leq \xi_\infty^{\mathrm{q}}(P) + \varepsilon$. Letting $\varepsilon \to 0$ we obtain that $\xi_\infty^{\mathrm{q}}(P) = \lim_{r\to\infty} \xi_r^{\mathrm{q}}(P)$. □

Next we show that finite convergence holds under a certain flatness condition: if $\xi_r^{\mathrm{q}}(P)$ admits a δ-flat optimal solution with $\delta = \lceil r/3\rceil + 1$, then $\xi_r^{\mathrm{q}}(P) = \xi_*^{\mathrm{q}}(P)$. This result is a variation of the flat extension result from Theorem 4.7, where δ now depends on the order r because the ideal constraints in $\xi_r^{\mathrm{q}}(P)$ depend on r.

Proposition 7.7. *If $\xi_r^{\mathrm{q}}(P)$ admits a $(\lceil r/3\rceil+1)$-flat optimal solution, then we have $\xi_r^{\mathrm{q}}(P) = \xi_*^{\mathrm{q}}(P)$.*

Proof. Let $\delta = \lceil r/3\rceil + 1$ and let L be a δ-flat optimal solution to $\xi_r^{\mathrm{q}}(P)$, i.e., such that $\mathrm{rank}(M_r(L)) = \mathrm{rank}(M_{r-\delta}(L))$. We have to show $\xi_r^{\mathrm{q}}(P) \geq \xi_*^{\mathrm{q}}(P)$, which we do by constructing a feasible solution $\hat{L}$ to $\xi_*^{\mathrm{q}}(P)$ with the same objective value $\hat{L}(1) = L(1)$. In the proof of Theorem 4.7, the linear form L is extended to a tracial symmetric linear form $\hat{L}$ on $\mathbb{R}\langle \mathbf{x}, \mathbf{y}, z\rangle$ that is nonnegative on $\mathcal{M}(\mathcal{G})$, zero on $\mathcal{I}(\mathcal{H})$, with $\mathrm{rank}(M(\hat{L})) < \infty$. To do this a subset W of $\langle \mathbf{x}, \mathbf{y}, z\rangle_{r-\delta}$ is found such that we have the vector space direct sum $\mathbb{R}\langle \mathbf{x}, \mathbf{y}, z\rangle = \mathrm{span}(W) \oplus \mathcal{I}(N_r(L))$, where $N_r(L)$ is the vector space

$$N_r(L) = \big\{p \in \mathbb{R}\langle \mathbf{x}, \mathbf{y}, z\rangle_r : L(qp) = 0 \text{ for all } q \in \mathbb{R}\langle \mathbf{x}, \mathbf{y}, z\rangle_r\big\}.$$

It is moreover shown that $\mathcal{I}(N_r(L)) \subseteq N(\hat{L})$. For $p \in \mathbb{R}\langle \mathbf{x}, \mathbf{y}, z\rangle$ we denote by r_p the unique element in $\mathrm{span}(W)$ such that $p - r_p \in \mathcal{I}(N_r(L))$.

We only need to show that $\hat{L}$ is zero on $\mathcal{I}(\mathcal{R}_\infty)$. Fix $u, v, w \in \mathbb{R}\langle \mathbf{x}, \mathbf{y}, z\rangle$. Then we have

$$\hat{L}(w(zuzvz - zvzuz)) = \hat{L}(wzuzvz) - \hat{L}(wzvzuz).$$

Since $\hat{L}$ is tracial and $u - r_u, v - r_v, w - r_w \in \mathcal{I}(N_r(L)) \subseteq N(\hat{L})$, we have

$$\hat{L}(wzuzvz) = \hat{L}(r_wzr_uzr_vz) \quad \text{and} \quad \hat{L}(wzvzuz) = \hat{L}(r_wzr_vzr_uz).$$

Since $\deg(r_uzr_vzr_wz) = \deg(r_vzr_uzr_wz) \leq 3 + 3(r - \delta) \leq 2r$ we have

$$\hat{L}(r_wzr_uzr_vz) = L(r_wzr_uzr_vz) \quad \text{and} \quad \hat{L}(r_wzr_vzr_uz) = L(r_wzr_vzr_uz).$$

So $L = 0$ on $\mathcal{I}_{2r}(\mathcal{R}_r)$ implies $\hat{L} = 0$ on $\mathcal{I}(\mathcal{R}_\infty)$.

Since $\hat{L}$ extends L we have $\hat{L}(z) = L(z) = 1$ and $\hat{L}(x_s^a y_t^b z) = L(x_s^a y_t^b z) = P(a, b|s, t)$ for all a, b, s, t. So, $\hat{L}$ is feasible for $\xi_*^{\mathrm{q}}(P)$ and has the same objective value $\hat{L}(1) = L(1)$. □

Chapter 8

Quantum graph parameters

This chapter is based on the paper "Bounds on entanglement dimensions and quantum graph parameters via noncommutative polynomial optimization", by S. Gribling, D. de Laat, and M. Laurent [GdLL18].

In this chapter we continue to study entanglement in bipartite quantum correlations. In particular, we study the advantage entanglement can provide in the setting of two specific nonlocal games: the quantum coloring game and the quantum stability game. We introduce semidefinite programming hierarchies and unify existing bounds on quantum chromatic and quantum stability numbers by placing them in the framework of tracial polynomial optimization.

We have introduced nonlocal games in Section 3.3 as abstract models to quantify the power of entanglement, and in Chapter 6 we have used them to construct matrices with a high completely positive semidefinite rank.

8.1 Nonlocal games for graph parameters

Let $G = (V, E)$ be a simple, undirected graph with vertex set V and edge set $E \subseteq V \times V$. Assume $|V| = n \in \mathbb{N}$. The *stability number* of G, denoted $\alpha(G)$, is defined as the size of the largest stable set in the graph G:

$$\alpha(G) = \max\{|S| : S \subseteq V,\, (S \times S) \cap E = \emptyset\}.$$

The *chromatic number*, also called the coloring number, of G is defined as the smallest number of colors needed to color the vertices of G in such a way that no two adjacent vertices receive the same color:

$$\chi(G) = \min\{k \in \mathbb{N} : \exists\, c : V \to [k] \text{ s.t. } (i,j) \in E \Rightarrow c(i) \neq c(j)\}.$$

Alternatively, one can define each of these parameters in terms of a nonlocal game in which two parties try to convince a referee that the graph has a certain stability number/chromatic number. For instance, in the graph coloring game the two players Alice and Bob try to convince the referee that they know a coloring c

using only k colors. The referee attempts to verify this by selecting a pair of vertices $(i,j) \in V \times V$ uniformly at random, and asking Alice how she colors i and Bob how he colors j. Alice and Bob are allowed to decide on a strategy before the game starts, but during the game they are not allowed to communicate (in particular they don't know each other's questions). Alice and Bob 'win' if their answers are consistent with the same k-coloring. The referee becomes convinced that they know a valid k-coloring if they win with probability 1, i.e., if they have a perfect strategy (see Equation (3.9)). Below we describe these nonlocal games formally and we show how they can be used to define the quantum analogues of the classical parameters. These nonlocal games use the set $[k]$ (whose elements are denoted as a, b) and the set V of vertices of a graph G (whose elements are denoted as i, j) as question and answer sets.

The quantum coloring number. In the *quantum coloring game*, introduced in [AHKS06, CMN$^+$07], we have a graph $G = (V, E)$ and an integer k. Here we have question sets $S = T = V$ and answer sets $A = B = [k]$, and the distribution π is strictly positive on $V \times V$. The predicate f is such that the players' answers have to be consistent with having a k-coloring of G; that is, $f(a, b, i, j) = 0$ precisely when ($i = j$ and $a \neq b$) or ($\{i, j\} \in E$ and $a = b$), and $f(a, b, i, j) = 1$ otherwise. This expresses the fact that if Alice and Bob receive the same vertex, they should return the same color and if they receive adjacent vertices, they should return distinct colors. A perfect classical strategy exists if and only if a perfect deterministic strategy exists, and a perfect deterministic strategy means that the players agree on a fixed k-coloring of G. Hence the smallest number k of colors for which there exists a perfect classical strategy is equal to the classical chromatic number $\chi(G)$. It is therefore natural to define the quantum chromatic number as the smallest k for which there exists a perfect quantum strategy. Recall that a strategy is perfect if the probability of giving a wrong answer is zero (see Equation (3.9)). In this case, a strategy P is perfect if $P(a, b|i, j) = 0$ whenever ($i = j$ and $a \neq b$) or ($\{i, j\} \in E$ and $a = b$). Here the first condition says precisely that a perfect strategy P needs to be synchronous (see Equation (3.19)); when Alice and Bob receive the same question, they should provide the same answer.[1] We therefore have the following definition of the quantum chromatic number:

Definition 8.1. *The quantum chromatic number $\chi_q(G)$ is the smallest $k \in \mathbb{N}$ for which there exists a synchronous correlation $P = (P(a, b|i, j))$ in $C_{q,s}([k]^2 \times V^2)$ such that*

$$P(a, a|i, j) = 0 \quad \textit{for all} \quad a \in [k], \{i, j\} \in E.$$

The commuting quantum chromatic number $\chi_{qc}(G)$ is defined analogously by taking $P \in C_{qc,s}([k]^2 \times V^2)$.

The quantum stability number. In the *quantum stability number game*, introduced in [MR16b, Rob13], we again have a graph $G = (V, E)$ and $k \in \mathbb{N}$, but now

[1] Recall that the set of synchronous quantum correlations is denoted by $C_{q,s}(\Gamma)$ in the tensor model and $C_{qc,s}(\Gamma)$ in the commuting operator model.

we use the question set $[k] \times [k]$ and the answer set $V \times V$. The distribution π is again strictly positive on the question set and now the predicate f of the game is such that the players' answers have to be consistent with having a stable set of size k, that is, $f(i,j,a,b) = 0$ precisely when ($a = b$ and $i \neq j$) or ($a \neq b$ and ($i = j$ or $\{i,j\} \in E$)). This expresses the fact that when Alice and Bob receive the same index $a = b \in [k]$, they should answer with the same vertex $i = j$ of G, and if they receive distinct indices $a \neq b$ from $[k]$, they should answer with distinct nonadjacent vertices i and j of G. There is a perfect classical strategy precisely when there exists a stable set of size k, so that the largest integer k for which there exists a perfect classical strategy is equal to the stability number $\alpha(G)$. Again, such a strategy is necessarily synchronous, so we get the following definition.

Definition 8.2. *The quantum stability number $\alpha_q(G)$ is the largest integer $k \in \mathbb{N}$ for which there exists a synchronous correlation $P = (P(i,j|a,b))$ in $C_{q,s}(V^2 \times [k]^2)$ such that*

$$P(i,j|a,b) = 0 \quad \textit{whenever} \quad (i = j \textit{ or } \{i,j\} \in E) \textit{ and } a \neq b \in [k].$$

The commuting quantum stability number $\alpha_{qc}(G)$ is defined analogously by taking $P \in C_{qc,s}(V^2 \times [k]^2)$.

The classical parameters $\chi(G)$ and $\alpha(G)$ are NP-hard. The same holds for the quantum coloring number $\chi_q(G)$ [Ji13], and also for the quantum stability number $\alpha_q(G)$ in view of the following reduction to coloring shown in [MR16b]:

$$\chi_q(G) = \min\{k \in \mathbb{N} : \alpha_q(G \square K_k) = |V|\}. \tag{8.1}$$

Here $G \square K_k$ is the Cartesian product of the graph $G = (V,E)$ and the complete graph K_k. By construction we have

$$\chi_{qc}(G) \leq \chi_q(G) \leq \chi(G) \quad \text{and} \quad \alpha(G) \leq \alpha_q(G) \leq \alpha_{qc}(G).$$

A natural question is whether or not the above inequalities can be strict. We revisit this topic in Section 8.4. In short, the quantum parameters can indeed be strictly separated from their classical analogues, but we do not know how to separate the quantum parameter and its commuting operator model analogue. Such a separation would require infinite-dimensional entanglement to be useful for either of the nonlocal games. Finding such a separation is a motivation for the work in this chapter: new bounds on these parameters could potentially lead to a separation there as well.

8.2 Our results

We now give an overview of the results of Section 8.3 and refer to that section for formal definitions. In Section 8.3.1 we first reformulate the quantum graph parameters in terms of C^*-algebras, which allows us to use techniques from tracial polynomial optimization to formulate bounds on the quantum graph parameters. We define a hierarchy $\{\gamma_r^{\mathrm{col}}(G)\}$ of lower bounds on the commuting quantum chromatic number and a hierarchy $\{\gamma_r^{\mathrm{stab}}(G)\}$ of upper bounds on the commuting quantum stability number. We show the following convergence results for these hierarchies.

Proposition 8.5. *There is an integer $r_0 \in \mathbb{N}$ such that $\gamma_r^{\mathrm{col}}(G) = \chi_{qc}(G)$ and $\gamma_r^{\mathrm{stab}}(G) = \alpha_{qc}(G)$ for all $r \geq r_0$. Moreover, if $\gamma_r^{\mathrm{col}}(G)$ admits a flat optimal solution, then $\gamma_r^{\mathrm{col}}(G) = \chi_q(G)$, and if $\gamma_r^{\mathrm{stab}}(G)$ admits a flat optimal solution, then $\gamma_r^{\mathrm{stab}}(G) = \alpha_q(G)$.*

Then in Section 8.3.2 we define tracial analogues $\{\xi_r^{\mathrm{stab}}(G)\}$ and $\{\xi_r^{\mathrm{col}}(G)\}$ of Lasserre-type bounds on $\alpha(G)$ and $\chi(G)$ that provide hierarchies of bounds for their quantum analogues. These bounds are more economical than the bounds $\gamma_r^{\mathrm{col}}(G)$ and $\gamma_r^{\mathrm{stab}}(G)$ (since they use less variables) and they also permit to recover some known bounds for the quantum parameters. We show that $\xi_*^{\mathrm{stab}}(G)$, which is the parameter $\xi_\infty^{\mathrm{stab}}(G)$ with an additional rank constraint on the matrix variable, coincides with the projective packing number $\alpha_p(G)$ from [Rob13] and that $\xi_\infty^{\mathrm{stab}}(G)$ upper bounds $\alpha_{qc}(G)$.

Proposition 8.7. *For every graph G we have*

$$\xi_*^{\mathrm{stab}}(G) = \alpha_p(G) \geq \alpha_q(G) \text{ and } \xi_\infty^{\mathrm{stab}}(G) \geq \alpha_{qc}(G).$$

Next, we consider the chromatic number. The tracial hierarchy $\{\xi_r^{\mathrm{col}}(G)\}$ unifies two known bounds: the projective rank $\xi_f(G)$, a lower bound on the quantum chromatic number from [MR16b], and the tracial rank $\xi_{tr}(G)$, a lower bound on the commuting quantum chromatic number from [PSS$^+$16].

Proposition 8.9. *For every graph G we have*

$$\xi_*^{\mathrm{col}}(G) = \xi_f(G) \leq \chi_q(G) \text{ and } \xi_\infty^{\mathrm{col}}(G) = \xi_{tr}(G) \leq \chi_{qc}(G).$$

Let us put the result in perspective. For each graph G we have the inequality $\xi_{tr}(G) \leq \xi_f(G)$. In [DP16, Cor. 3.10] it is shown that the projective rank and the tracial rank coincide if Connes' embedding conjecture is true. That is, if Connes' embedding conjecture is true, then $\xi_*^{\mathrm{col}}(G) = \xi_\infty^{\mathrm{col}}(G)$ for every graph G.

Next, we establish some relations between the four hierarchies $\xi_r^{\mathrm{col}}(G)$, $\gamma_r^{\mathrm{col}}(G)$, $\xi_r^{\mathrm{stab}}(G)$, and $\gamma_r^{\mathrm{stab}}(G)$. For the coloring parameters, we show the analogue of reduction (8.1).

Proposition 8.14. *For every graph G and $r \in \mathbb{N} \cup \{\infty\}$ we have*

$$\gamma_r^{\mathrm{col}}(G) = \min\{k : \xi_r^{\mathrm{stab}}(G \square K_k) = |V|\}.$$

We show an analogous statement for the stability parameters, when using the homomorphic graph product of K_k with the complement of G, denoted here as $K_k \star G$, and the following reduction shown in [MR16b]:

$$\alpha_q(G) = \max\{k \in \mathbb{N} : \alpha_q(K_k \star G) = k\}.$$

Proposition 8.15. *For every graph G and $r \in \mathbb{N} \cup \{\infty\}$ we have*

$$\gamma_r^{\mathrm{stab}}(G) = \max\{k : \xi_r^{\mathrm{stab}}(K_k \star G) = k\}.$$

Finally, we show that the hierarchies $\{\gamma_r^{\mathrm{col}}(G)\}$ and $\{\gamma_r^{\mathrm{stab}}(G)\}$ refine the hierarchies $\{\xi_r^{\mathrm{col}}(G)\}$ and $\{\xi_r^{\mathrm{stab}}(G)\}$.

Proposition 8.16. *For every graph G and $r \in \mathbb{N} \cup \{\infty, *\}$ we have*

$$\xi_r^{\mathrm{col}}(G) \leq \gamma_r^{\mathrm{col}}(G) \text{ and } \xi_r^{\mathrm{stab}}(G) \geq \gamma_r^{\mathrm{stab}}(G).$$

8.3 Bounding quantum graph parameters

We investigate the quantum graph parameters $\alpha_q(G)$, $\gamma_q(G)$, $\alpha_{qc}(G)$, and $\chi_{qc}(G)$. They were introduced earlier in Section 8.1 in terms of nonlocal games and synchronous quantum correlations (in the tensor and commuting models). As we will see below, they can be reformulated in terms of the existence of positive semidefinite matrices with arbitrary size (or operators) satisfying a system of equations corresponding to the natural integer linear programming formulation of $\alpha(G)$ and $\chi(G)$. This opens the way to using techniques from noncommutative polynomial optimization for designing hierarchies of bounds for the quantum graph parameters. We present these approaches and compare them with known hierarchies for the classical graph parameters.

8.3.1 Hierarchies $\gamma_r^{\mathrm{col}}(G)$ and $\gamma_r^{\mathrm{stab}}(G)$ based on synchronous correlations

In Section 8.1 we introduced quantum chromatic numbers (Definition 8.1) and quantum stability numbers (Definition 8.2) in terms of synchronous quantum correlations satisfying certain linear constraints. We first give (known) reformulations in terms of C^*-algebras, and then we reformulate those in terms of tracial optimization, which leads to the hierarchies $\gamma_r^{\mathrm{col}}(G)$ and $\gamma_r^{\mathrm{stab}}(G)$.

The following result from [PSS$^+$16] allows us to write a synchronous quantum correlation in terms of C^*-algebras admitting a tracial state.

Theorem 8.3 ([PSS$^+$16]). *Let $\Gamma = A^2 \times S^2$ and $P \in \mathbb{R}^\Gamma$. We have $P \in C_{qc,s}(\Gamma)$ (resp., $P \in C_{q,s}(\Gamma)$) if and only if there exists a unital (resp., finite-dimensional) C^*-algebra $\mathcal{A}$ with a faithful tracial state τ and a set of projectors $\{X_s^a : s \in S, a \in A\} \subseteq \mathcal{A}$ satisfying $\sum_{a\in A} X_s^a = 1$ for all $s \in S$ and $P(a,b|s,t) = \tau(X_s^a X_t^b)$ for all $s,t \in S, a,b \in A$.*

Here we add the condition that τ is faithful, that is, $\tau(X^*X) = 0$ implies $X = 0$, since it follows from the GNS construction in the proof of [PSS$^+$16]. This means that

$$0 = P(a,b|s,t) = \tau(X_s^a X_t^b) = \tau((X_s^a)^2 (X_t^b)^2) = \tau((X_s^a X_t^b)^* X_s^a X_t^b)$$

implies $X_s^a X_t^b = 0$. It follows from Definition 8.1 and the above that $\chi_{qc}(G)$ is equal to the smallest $k \in \mathbb{N}$ for which there exists a C^*-algebra $\mathcal{A}$, a tracial state τ on $\mathcal{A}$, and a family of projectors $\{X_i^c : i \in V, c \in [k]\} \subseteq \mathcal{A}$ satisfying

$$\sum_{c\in[k]} X_i^c - 1 = 0 \quad \text{for all} \quad i \in V, \tag{8.2}$$

$$X_i^c X_j^{c'} = 0 \quad \text{if} \quad (c \neq c' \text{ and } i = j) \quad \text{or} \quad (c = c' \text{ and } \{i,j\} \in E). \tag{8.3}$$

The quantum chromatic number $\chi_q(G)$ is equal to the smallest $k \in \mathbb{N}$ for which there exists a *finite-dimensional* C^*-algebra $\mathcal{A}$ with the above properties.

Analogously, $\alpha_{qc}(G)$ is equal to the largest $k \in \mathbb{N}$ for which there is a C^*-algebra $\mathcal{A}$, a tracial state τ on $\mathcal{A}$, and a set of projectors $\{X_c^i : c \in [k], i \in V\} \subseteq \mathcal{A}$ satisfying

$$\sum_{i \in V} X_c^i - 1 = 0 \quad \text{for all} \quad c \in [k], \tag{8.4}$$

$$X_c^i X_{c'}^j = 0 \quad \text{if } (i \neq j \text{ and } c = c') \quad \text{or} \quad ((i = j \text{ or } \{i,j\} \in E) \text{ and } c \neq c'), \tag{8.5}$$

and $\alpha_q(G)$ is equal to the largest $k \in \mathbb{N}$ for which $\mathcal{A}$ can be taken finite-dimensional.

These reformulations of the graph parameters $\chi_q(G), \chi_{qc}(G), \alpha_q(G)$ and $\alpha_{qc}(G)$ also follow from [OP16, Thm. 4.7], where general quantum graph homomorphisms are considered; the reformulations of $\chi_q(G)$ and $\chi_{qc}(G)$ are also made explicit in [OP16, Thm. 4.12].

Remark 8.4. *The above definition for the parameters $\alpha_q(G)$ and $\chi_q(G)$ (tensor model) can be simplified. Indeed, instead of asking for projectors $\{X_i^c\}$ in a finite-dimensional C^*-algebra equipped with a tracial state and satisfying the constraints* (8.2)-(8.3) *or* (8.4)-*8.5, one may ask for such projectors that are* matrices *of unspecified (but finite) size (as in [CMN*$^+$*07, MR16b, SV17]). This can be seen in the following two ways.*

A first possibility is to apply Artin-Wedderburn theory, which tells us that any finite-dimensional C^*-algebra is isomorphic to a matrix algebra.

An alternative, more elementary way is to use the link presented in Section 3.4 between synchronous quantum correlations and completely positive semidefinite matrices. Indeed, as we have seen there, having a synchronous quantum correlation $P = (P(c,c'|i,j)) \in \mathbb{R}^{V^2 \times [k]^2}$ certifying $\chi_q(G) \leq k$ is equivalent to having a set of positive semidefinite matrices $\{X_i^c\}$ satisfying the constraints (8.2)-(8.3)*. Indeed, the proof of Proposition 3.7 shows that there exist Hermitian positive semidefinite matrices $\{Y_i^c\}$ and K that satisfy $P(c,c'|i,j) = \mathrm{Tr}(Y_i^c Y_j^{c'})$ for all $i,j \in V$, $c,c' \in [k]$ and $\sum_{c \in [k]} Y_i^c = K$ for all $i \in V$. We can see that the matrices $\{Y_i^c\}$ satisfy* (8.3) *by using the fact that, since $Y_i^c, Y_j^{c'} \succeq 0$, we have $P(c,c'|i,j) = \mathrm{Tr}(Y_i^c Y_j^{c'}) = 0$ if and only if $Y_i^c Y_j^{c'} = 0$. Next, we obtain matrices $\{X_i^c\}$ that satisfy both the constraints* (8.2) *and* (8.3) *by letting each X_i^c be the projection onto the image of Y_i^c. Finally, observe that the constraints* (8.2)-(8.3) *imply that the matrices X_i^c are projectors. Indeed, for every i, c', by multiplying* (8.2) *by $X_i^{c'}$ and using* (8.3) *we obtain $(X_i^{c'})^2 = X_i^{c'}$. The analogous result holds of course for the quantum stability number $\alpha_q(G)$.*

Note that restricting to scalar solutions (1×1 matrices) in these feasibility problems recovers the classical graph parameters $\chi(G)$ and $\alpha(G)$.

We now reinterpret the above formulations in terms of tracial optimization. Given a graph $G = (V,E)$, let $i \sim j$ denote $\{i,j\} \in E$ and let $i \simeq j$ denote $\{i,j\} \in E$ or $i = j$. For $k \in \mathbb{N}$, let $\mathcal{H}_{G,k}^{\mathrm{col}}$ and $\mathcal{H}_{G,k}^{\mathrm{stab}}$ denote the sets of polynomials corresponding to equations (8.2)–(8.3) and (8.4)–(8.5):

$$\mathcal{H}_{G,k}^{\mathrm{col}} = \big\{1 - \sum_{c\in[k]} x_i^c : i \in V\big\} \cup \big\{x_i^c x_j^{c'} : (c \neq c' \text{ and } i = j) \text{ or } (c = c' \text{ and } i \sim j)\big\},$$
$$\mathcal{H}_{G,k}^{\mathrm{stab}} = \big\{1 - \sum_{i\in V} x_c^i : c \in [k]\big\} \cup \big\{x_c^i x_{c'}^j : (i \neq j \text{ and } c = c') \text{ or } (i \simeq j \text{ and } c \neq c')\big\}.$$

We have

$$1 - (x_i^c)^2 \in \mathcal{M}_2(\emptyset) + \mathcal{I}_2(\mathcal{H}_{G,k}^{\mathrm{col}}),$$

since $1 - (x_i^c)^2 = (1 - x_i^c)^2 + 2(x_i^c - (x_i^c)^2)$, and

$$x_i^c - (x_i^c)^2 = x_i^c\big(1 - \sum_{c'} x_i^{c'}\big) + \sum_{c':c'\neq c} x_i^c x_i^{c'} \in \mathcal{I}_2(\mathcal{H}_{G,k}^{\mathrm{col}}), \tag{8.6}$$

and the analogous statements hold for $\mathcal{H}_{G,k}^{\mathrm{stab}}$. Hence, both $\mathcal{M}(\emptyset) + \mathcal{I}(\mathcal{H}_k^{\mathrm{col}})$ and $\mathcal{M}(\emptyset) + \mathcal{I}(\mathcal{H}_k^{\mathrm{stab}})$ are Archimedean and we can apply Theorems 4.5 and 4.6 to express the quantum graph parameters in terms of positive tracial linear functionals. Namely,

$$\chi_{qc}(G) = \min\big\{k \in \mathbb{N} : \exists\, L \in \mathbb{R}\langle\{x_i^c : i \in V, c \in [k]\}\rangle^* \text{ symmetric, tracial, positive,}$$
$$L(1) = 1,\ L = 0 \text{ on } \mathcal{I}(\mathcal{H}_{G,k}^{\mathrm{col}})\big\},$$

and $\chi_q(G)$ is obtained by adding the constraint $\mathrm{rank}(M(L)) < \infty$. Likewise,

$$\alpha_{qc}(G) = \max\big\{k \in \mathbb{N} : \exists\, L \in \mathbb{R}\langle\{x_c^i : c \in [k], i \in V\}\rangle^* \text{ symmetric, tracial, positive,}$$
$$L(1) = 1,\ L = 0 \text{ on } \mathcal{I}(\mathcal{H}_{G,k}^{\mathrm{stab}})\big\},$$

and $\alpha_q(G)$ is given by this program with the additional constraint $\mathrm{rank}(M(L)) < \infty$.

Starting from these formulations it is natural to define a hierarchy $\{\gamma_r^{\mathrm{col}}(G)\}$ of lower bounds on $\chi_{qc}(G)$ and a hierarchy $\{\gamma_r^{\mathrm{stab}}(G)\}$ of upper bounds on $\alpha_{qc}(G)$, where the bounds of order $r \in \mathbb{N}$ are obtained by truncating L to polynomials of degree at most $2r$ and truncating the ideal to degree $2r$:

$$\gamma_r^{\mathrm{col}}(G) = \min\big\{k \in \mathbb{N} : \exists\, L \in \mathbb{R}\langle\{x_i^c : i \in V, c \in [k]\}\rangle_{2r}^* \text{ symmetric, tracial,}$$
$$\text{positive, } L(1) = 1,\ L = 0 \text{ on } \mathcal{I}_{2r}(\mathcal{H}_{G,k}^{\mathrm{col}})\big\},$$
$$\gamma_r^{\mathrm{stab}}(G) = \max\big\{k \in \mathbb{N} : \exists\, L \in \mathbb{R}\langle\{x_c^i : c \in [k], i \in V\}\rangle_{2r}^* \text{ symmetric, tracial,}$$
$$\text{positive, } L(1) = 1,\ L = 0 \text{ on } \mathcal{I}_{2r}(\mathcal{H}_{G,k}^{\mathrm{stab}})\big\}.$$

Then, by defining $\gamma_*^{\mathrm{col}}(G)$ and $\gamma_*^{\mathrm{stab}}(G)$ by adding the constraint $\mathrm{rank}(M(L)) < \infty$ to $\gamma_\infty^{\mathrm{col}}(G)$ and $\gamma_\infty^{\mathrm{stab}}(G)$, we have

$$\gamma_\infty^{\mathrm{col}}(G) = \chi_{qc}(G), \quad \gamma_\infty^{\mathrm{stab}}(G) = \alpha_{qc}(G), \quad \gamma_*^{\mathrm{col}}(G) = \chi_q(G), \quad \text{and} \quad \gamma_*^{\mathrm{stab}}(G) = \alpha_q(G).$$

The optimization problems $\gamma_r^{\mathrm{col}}(G)$, for $r \in \mathbb{N}$, can be computed by semidefinite programming and binary search on k. If there is an optimal solution (k, L) to $\gamma_r^{\mathrm{col}}(G)$ with L flat, then, by Theorem 4.7, we have equality $\gamma_r^{\mathrm{col}}(G) = \chi_q(G)$. Since

$\{\gamma_r^{\mathrm{col}}(G)\}_{r\in\mathbb{N}}$ is a monotone nondecreasing sequence of lower bounds on $\chi_q(G)$, there exists an r_0 such that for all $r \geq r_0$ we have $\gamma_r^{\mathrm{col}}(G) = \gamma_{r_0}^{\mathrm{col}}(G)$, which is equal to $\gamma_\infty^{\mathrm{col}}(G) = \chi_{qc}(G)$ by Lemma 4.15. The analogous statements hold for the parameters $\gamma_r^{\mathrm{stab}}(G)$. Hence, we have shown the following result.

Proposition 8.5. *There is an integer $r_0 \in \mathbb{N}$ such that $\gamma_r^{\mathrm{col}}(G) = \chi_{qc}(G)$ and $\gamma_r^{\mathrm{stab}}(G) = \alpha_{qc}(G)$ for all $r \geq r_0$. Moreover, if $\gamma_r^{\mathrm{col}}(G)$ admits a flat optimal solution, then $\gamma_r^{\mathrm{col}}(G) = \chi_q(G)$, and if $\gamma_r^{\mathrm{stab}}(G)$ admits a flat optimal solution, then $\gamma_r^{\mathrm{stab}}(G) = \alpha_q(G)$.*

Remark 8.6. *A hierarchy $\{\mathcal{Q}_r(\Gamma)\}$ of semidefinite outer approximations for the set $C_{qc}(\Gamma)$ of commuting quantum correlations was constructed in [PSS+16] (revisiting the approach in [NPA08, PNA10]). This hierarchy converges, that is,*

$$C_{qc}(\Gamma) = \mathcal{Q}_\infty(\Gamma) = \bigcap_{r\in\mathbb{N}} \mathcal{Q}_r(\Gamma).$$

These approximations $\mathcal{Q}_r(\Gamma)$ are based on the eigenvalue optimization approach, applied to the formulation (3.4) of commuting quantum correlations. So they use linear functionals on polynomials involving the two sets of variables x_s^a and y_t^b for $(a,b,s,t) \in \Gamma$. Paulsen et al. [PSS+16] use these outer approximations to define a hierarchy of lower bounds converging to $\chi_{qc}(G)$, where the bounds are defined in terms of feasibility problems over the sets $\mathcal{Q}_r(\Gamma)$.

For synchronous correlations we can use the result of Theorem 8.3 and the tracial optimization approach used here to directly define a converging hierarchy $\{\mathcal{Q}_{r,s}(\Gamma)\}$ of outer semidefinite approximations for the set $C_{qc,s}(\Gamma)$ of synchronous commuting quantum correlations. These approximations now use linear functionals on polynomials involving only one set of variables x_s^a for $(a,s) \in A \times S$. Namely, for $r \in \mathbb{N} \cup \{\infty\}$ define $\mathcal{Q}_{r,s}(\Gamma)$ as the set of $P \in \mathbb{R}^\Gamma$ for which there exists a symmetric, tracial, positive linear functional $L \in \mathbb{R}\langle\{x_s^a : (a,s) \in A \times S\}\rangle_{2r}^*$ such that $L(1) = 1$ and $L = 0$ on the ideal generated by the polynomials $x_s^a - (x_s^a)^2$ $((a,s) \in A \times S)$ and $1 - \sum_{a\in A} x_s^a$ $(s \in S)$, truncated at degree $2r$. Then we have

$$C_{qc,s}(\Gamma) = \mathcal{Q}_{\infty,s}(\Gamma) = \bigcap_{r\in\mathbb{N}} \mathcal{Q}_{r,s}(\Gamma).$$

The synchronous value *of a nonlocal game is defined in [DP16] as the maximum value of the objective function (3.8) over the set $C_{qc,s}(\Gamma)$. By maximizing the objective (3.8) over the relaxations $\mathcal{Q}_{r,s}(\Gamma)$ we get a hierarchy of semidefinite programming upper bounds that converges to the synchronous value of the game. Finally note that one can also view the parameters $\gamma_r^{\mathrm{col}}(G)$ as solving feasibility problems over the sets $\mathcal{Q}_{r,s}(\Gamma)$.*

8.3.2 Hierarchies $\xi_r^{\mathrm{col}}(G)$ and $\xi_r^{\mathrm{stab}}(G)$ based on Lasserre-type bounds

Here we revisit some known Lasserre-type hierarchies for the classical stability number $\alpha(G)$ and chromatic number $\chi(G)$ and we show that their tracial noncommutative analogues can be used to recover known parameters such as the projective packing number $\alpha_p(G)$, the projective rank $\xi_f(G)$, and the tracial rank $\xi_{\mathrm{tr}}(G)$. Both the commutative hierarchies and the tracial noncommutative hierarchies can be viewed as strengthenings of the Lovász theta number towards either the (quantum) stability number or the (quantum) chromatic number: the first level corresponds to the theta number. Compared to the hierarchies defined in the previous section, these Lasserre-type hierarchies use less variables (they only use variables indexed by the vertices of the graph G), but they also do not converge to the (commuting) quantum chromatic or stability number.

Given a graph $G = (V, E)$, define the set of polynomials

$$\mathcal{H}_G = \big\{x_i - x_i^2 : i \in V\big\} \cup \big\{x_i x_j : \{i,j\} \in E\big\}$$

in the variables $\mathbf{x} = (x_i : i \in V)$ (which are commutative or noncommutative depending on the context). Note that $1 - x_i^2 \in \mathcal{M}_2(\emptyset) + \mathcal{I}_2(\mathcal{H}_G)$ for all $i \in V$, so that $\mathcal{M}(\emptyset) + \mathcal{I}(\mathcal{H}_G)$ is Archimedean.

Semidefinite programming bounds on the projective packing number

We first recall the Lasserre hierarchy of bounds for the classical stability number $\alpha(G)$. Starting from the formulation of $\alpha(G)$ via the optimization problem

$$\alpha(G) = \sup\Big\{\sum_{i\in V} x_i : x \in \mathbb{R}^n,\ h(x) = 0 \text{ for all } h \in \mathcal{H}_G\Big\}, \tag{8.7}$$

the r-th level of the Lasserre hierarchy for $\alpha(G)$ (introduced in [Las01, Lau03]) is defined by

$$\mathrm{las}_r^{\mathrm{stab}}(G) = \sup\Big\{L\big(\sum_{i\in V} x_i\big) : L \in \mathbb{R}[\mathbf{x}]_{2r}^* \text{ positive},\ L(1) = 1,\ L = 0 \text{ on } \mathcal{I}_{2r}(\mathcal{H}_G)\Big\}.$$

Then we have $\mathrm{las}_{r+1}^{\mathrm{stab}}(G) \leq \mathrm{las}_r^{\mathrm{stab}}(G)$ and the first bound is Lovász' theta number: $\mathrm{las}_1^{\mathrm{stab}}(G) = \vartheta(G)$. Finite convergence to $\alpha(G)$ is shown in [Lau03]:

$$\mathrm{las}_{\alpha(G)}^{\mathrm{stab}}(G) = \alpha(G).$$

Roberson [Rob13] introduces the *projective packing number*

$$\begin{aligned}\alpha_p(G) &= \sup\Big\{\frac{1}{d}\sum_{i\in V} \operatorname{rank} X_i : d \in \mathbb{N},\ \mathbf{X} \in (\mathrm{S}^d)^n \text{ projectors}, \\ &\qquad\qquad X_i X_j = 0 \text{ for } \{i,j\} \in E\Big\} \qquad (8.8)\\ &= \sup\Big\{\frac{1}{d}\mathrm{Tr}\Big(\sum_{i\in V} X_i\Big) : d \in \mathbb{N},\ \mathbf{X} \in (\mathrm{S}^d)^n,\ h(\mathbf{X}) = 0 \text{ for } h \in \mathcal{H}_G\Big\} \qquad (8.9)\end{aligned}$$

as an upper bound for the quantum stability number $\alpha_q(G)$. Note that the inequality $\alpha_q(G) \leq \alpha_p(G)$ also follows from Proposition 8.7 below. Comparing (8.7) and (8.9) we see that the parameter $\alpha_p(G)$ can be viewed as a noncommutative analogue of $\alpha(G)$.

For $r \in \mathbb{N} \cup \{\infty\}$ we define the noncommutative analogue of $\text{las}_r^{\text{stab}}(G)$ by

$$\xi_r^{\text{stab}}(G) = \sup\Big\{L\Big(\sum_{i\in V} x_i\Big) : L \in \mathbb{R}\langle \mathbf{x}\rangle_{2r}^* \text{ tracial, symmetric, and positive,} \\ L(1) = 1,\ L = 0 \text{ on } \mathcal{I}_{2r}(\mathcal{H}_G)\Big\},$$

and $\xi_*^{\text{stab}}(G)$ by adding the constraint $\text{rank}(M(L)) < \infty$ to the definition of $\xi_\infty^{\text{stab}}(G)$.

In view of Theorems 4.5 and 4.6, both $\xi_\infty^{\text{stab}}(G)$ and $\xi_*^{\text{stab}}(G)$ can be reformulated in terms of C^*-algebras: $\xi_\infty^{\text{stab}}(G)$ (resp., $\xi_*^{\text{stab}}(G)$) is the largest value of $\tau(\sum_{i\in V} X_i)$, where $\mathcal{A}$ is a (resp., finite-dimensional) C^*-algebra with tracial state τ and $X_i \in \mathcal{A}$ ($i \in [n]$) are projectors satisfying $X_iX_j = 0$ for all $\{i,j\} \in E$. Moreover, as we now see, the parameter $\xi_*^{\text{stab}}(G)$ coincides with the projective packing number and the parameters $\xi_*^{\text{stab}}(G)$ and $\xi_\infty^{\text{stab}}(G)$ upper bound the quantum stability numbers.

Proposition 8.7. *For every graph G we have*

$$\xi_*^{\text{stab}}(G) = \alpha_p(G) \geq \alpha_q(G) \text{ and } \xi_\infty^{\text{stab}}(G) \geq \alpha_{qc}(G).$$

Proof. By (8.9), $\alpha_p(G)$ is the largest value of $L(\sum_{i\in V} x_i)$ taken over all linear functionals L that are normalized trace evaluations at projectors $\mathbf{X} \in (\mathcal{S}^d)^n$ (for some $d \in \mathbb{N}$) with $X_iX_j = 0$ for $\{i,j\} \in E$. By convexity the optimum remains unchanged when considering a convex combination of such trace evaluations. In view of Theorem 4.6 (the equivalence between (1) and (3)), we can conclude that this optimum value is precisely the parameter $\xi_*^{\text{stab}}(G)$. This shows equality $\alpha_p(G) = \xi_*^{\text{stab}}(G)$.

Consider a C^*-algebra $\mathcal{A}$ with tracial state τ and a set of projectors $X_c^i \in \mathcal{A}$ (for $i \in V$, $c \in [k]$) satisfying (8.4)-(8.5). Then, setting $X_i = \sum_{c\in[k]} X_c^i$ for $i \in V$, we obtain projectors $X_i \in \mathcal{A}$ that satisfy $X_iX_j = 0$ if $\{i,j\} \in E$. Moreover, the following holds: $\tau(\sum_{i\in V} X_i) = \sum_{c\in[k]} \tau(\sum_{i\in V} X_c^i) = k$. This shows that $\xi_\infty^{\text{stab}}(G) \geq \alpha_{qc}(G)$ and, when restricting $\mathcal{A}$ to be finite-dimensional, $\xi_*^{\text{stab}}(G) \geq \alpha_q(G)$. □

Using Lemma 4.15 one can verify that $\xi_r^{\text{stab}}(G)$ converges to $\xi_\infty^{\text{stab}}(G)$ as $r \to \infty$, and for $r \in \mathbb{N} \cup \{\infty\}$ the infimum in $\xi_r^{\text{stab}}(G)$ is attained. Moreover, by Theorem 4.7, if $\xi_r^{\text{stab}}(G)$ admits a flat optimal solution, then equality $\xi_r^{\text{stab}}(G) = \xi_*^{\text{stab}}(G)$ holds. The first bound $\xi_1^{\text{stab}}(G)$ coincides with the theta number, since $\xi_1^{\text{stab}}(G) = \text{las}_1^{\text{stab}}(G) = \vartheta(G)$. Summarizing, we have $\alpha_{qc}(G) \leq \xi_\infty^{\text{stab}}(G)$ and the following chain of inequalities

$$\alpha_q(G) \leq \alpha_p(G) = \xi_*^{\text{stab}}(G) \leq \xi_\infty^{\text{stab}}(G) \leq \xi_r^{\text{stab}}(G) \leq \xi_1^{\text{stab}}(G) = \vartheta(G),$$

where the bounds $\xi_r^{\text{stab}}(G)$ ($r \in \mathbb{N}$) are semidefinite programs, and $\alpha_q(G)$ is NP-hard to compute.

Semidefinite programming bounds on the projective rank and tracial rank

We now turn to the (quantum) chromatic numbers. First recall the definition of the fractional chromatic number:

$$\chi_f(G) := \min\Big\{ \sum_{S\in\mathcal{S}_G} \lambda_S : \lambda \in \mathbb{R}_+^{\mathcal{S}_G},\ \sum_{S\in\mathcal{S}_G : i\in S} \lambda_S = 1 \text{ for all } i \in V \Big\},$$

where $\mathcal{S}_G$ is the set of stable sets of G. Clearly, $\chi_f(G) \leq \chi(G)$. The following Lasserre type lower bounds for the classical chromatic number $\chi(G)$ are defined in [GL08b]:

$$\text{las}_r^{\text{col}}(G) = \inf\big\{L(1) : L \in \mathbb{R}[\mathbf{x}]_{2r}^* \text{ positive},\ L(x_i) = 1\ (i \in V),\ L = 0 \text{ on } \mathcal{I}_{2r}(\mathcal{H}_G)\big\}.$$

Note that we may view $\chi_f(G)$ as minimizing $L(1)$ over all linear functionals $L \in \mathbb{R}[\mathbf{x}]^*$ that are conic combinations of evaluations at characteristic vectors of stable sets. From this we see that $\text{las}_r^{\text{col}}(G) \leq \chi_f(G)$ for all $r \geq 1$. In [GL08b] it is shown that finite convergence to $\chi_f(G)$ holds:

$$\text{las}_{\alpha(G)}^{\text{col}}(G) = \chi_f(G).$$

The bound of order $r = 1$ coincides with the theta number: $\text{las}_1^{\text{col}}(G) = \vartheta(\overline{G})$.

The following parameter $\xi_f(G)$, called the *projective rank* of G, was introduced in [MR16b] as a lower bound on the quantum chromatic number $\chi_q(G)$:

$$\xi_f(G) := \inf\Big\{\frac{d}{r} : d, r \in \mathbb{N},\ X_1, \ldots, X_n \in \mathrm{S}^d,\ \mathrm{Tr}(X_i) = r\ (i \in V),$$
$$X_i^2 = X_i\ (i \in V),\ X_iX_j = 0\ (\{i,j\} \in E)\Big\}.$$

Proposition 8.8 ([MR16b])**.** *For every graph G we have $\xi_f(G) \leq \chi_q(G)$.*

Proof. Set $k = \chi_q(G)$. It is shown in [CMN$^+$07] that in the definition of $\chi_q(G)$ from (8.2)-(8.3), one may assume w.l.o.g. that X_i^c are projectors that all have the same rank, say, r. Then, for any given color $c \in [k]$, the matrices X_i^c $(i \in V)$ provide a feasible solution to $\xi_f(G)$ with value d/r. This shows $\xi_f(G) \leq d/r$. Finally, $d/r = k$ holds since by (8.2)-(8.3) we have $d = \text{rank}(I) = \sum_{c=1}^k \text{rank}(X_i^c) = kr$. □

In [PSS$^+$16, Prop. 5.11] it is shown that the projective rank can equivalently be defined as

$$\xi_f(G) = \inf\big\{\lambda : \exists \text{ finite-dimensional } C^*\text{-algebra } \mathcal{A} \text{ with tracial state } \tau,$$
$$X_i \in \mathcal{A} \text{ projector with } \tau(X_i) = \frac{1}{\lambda} \text{ for } i \in V,$$
$$X_iX_j = 0 \text{ for } \{i,j\} \in E\big\}.$$

Paulsen et al. [PSS$^+$16] also define the *tracial rank* $\xi_{tr}(G)$ of G as the parameter obtained by omitting in the above definition of $\xi_f(G)$ the restriction that $\mathcal{A}$ has to be finite-dimensional. The motivation for the parameter $\xi_{tr}(G)$ is that it lower bounds the *commuting* quantum chromatic number [PSS+16, Thm. 5.11]:

$$\xi_{tr}(G) \leq \chi_{qc}(G).$$

Using Theorems 4.5 and 4.6 (which we apply to $L/L(1)$ when L is not normalized), we obtain the following reformulations:

$$\xi_f(G) = \inf\big\{L(1) : L \in \mathbb{R}\langle \mathbf{x}\rangle^* \text{ tracial, symmetric, positive, } \mathrm{rank}(M(L)) < \infty, \\ L(x_i) = 1 \ (i \in V),\ L = 0 \text{ on } \mathcal{I}(\mathcal{H}_G)\big\},$$

and $\xi_{tr}(G)$ is obtained by the same program without the restriction $\mathrm{rank}(M(L)) < \infty$. In addition, we obtain that in this formulation of $\xi_f(G)$ we can equivalently optimize over all L that are conic combinations of trace evaluations at projectors $X_i \in \mathrm{S}^d$ (for some $d \in \mathbb{N}$) satisfying $X_iX_j = 0$ for all $\{i,j\} \in E$. If we restrict the optimization to conic combinations of *scalar* evaluations ($d = 1$) we obtain the fractional chromatic number. This shows that the projective rank can be seen as the noncommutative analogue of the fractional chromatic number, as was already observed in [MR16b, PSS+16].

The above formulations of the parameters $\xi_{tr}(G)$ and $\xi_f(G)$ in terms of linear functionals also show that they fit within the following hierarchy $\{\xi_r^{\mathrm{col}}(G)\}_{r\in\mathbb{N}\cup\{\infty\}}$, defined as the noncommutative tracial analogue of the hierarchy $\{\mathrm{las}_r^{\mathrm{col}}(G)\}_r$:

$$\xi_r^{\mathrm{col}}(G) = \inf\big\{L(1) : L \in \mathbb{R}\langle \mathbf{x}\rangle_{2r}^* \text{ tracial, symmetric, and positive,} \\ L(x_i) = 1 \ (i \in V),\ L = 0 \text{ on } \mathcal{I}_{2r}(\mathcal{H}_G)\big\}.$$

Again, $\xi_*^{\mathrm{col}}(G)$ is the parameter obtained by adding the constraint $\mathrm{rank}(M(L)) < \infty$ to the program defining $\xi_\infty^{\mathrm{col}}(G)$. By the above discussion the following holds.

Proposition 8.9. *For every graph G we have*

$$\xi_*^{\mathrm{col}}(G) = \xi_f(G) \leq \chi_q(G) \textit{ and } \xi_\infty^{\mathrm{col}}(G) = \xi_{tr}(G) \leq \chi_{qc}(G).$$

Using Lemma 4.15 one can verify that the parameters $\xi_r^{\mathrm{col}}(G)$ converge to $\xi_\infty^{\mathrm{col}}(G)$. Moreover, by Theorem 4.7, if $\xi_r^{\mathrm{col}}(G)$ admits a flat optimal solution, then we have $\xi_r^{\mathrm{col}} = \xi_*^{\mathrm{col}}(G)$. Also, the parameter $\xi_1^{\mathrm{col}}(G)$ coincides with $\mathrm{las}_1^{\mathrm{col}}(G) = \vartheta(\overline{G})$. Summarizing we have $\xi_\infty^{\mathrm{col}}(G) = \xi_{tr}(G) \leq \chi_{qc}(G)$ and the following chain of inequalities

$$\vartheta(\overline{G}) = \xi_1^{\mathrm{col}}(G) \leq \xi_r^{\mathrm{col}}(G) \leq \xi_\infty^{\mathrm{col}}(G) = \xi_{tr}(G) \leq \xi_*^{\mathrm{col}}(G) = \xi_f(G) \leq \chi_q(G).$$

Observe that the bounds $\mathrm{las}_r^{\mathrm{col}}(G)$ and $\xi_r^{\mathrm{col}}(G)$ remain below the fractional chromatic number $\chi_f(G)$, since $\xi_f(G) = \xi_*^{\mathrm{col}}(G) \leq \mathrm{las}_*^{\mathrm{col}}(G) = \chi_f(G)$. Hence, these bounds are weak if $\chi_f(G)$ is close to $\vartheta(\overline{G})$ and far from $\chi(G)$ or $\chi_q(G)$. In the classical setting this is the case, e.g., for the class of Kneser graphs $G = K(n,r)$, with vertex set the set of all r-subsets of $[n]$ and having an edge between any two disjoint r-subsets. By results of Lovász [Lov78, Lov79], the fractional chromatic number is n/r, which is known to be equal to $\vartheta(\overline{K(n,r)})$, while the chromatic number is

$n - 2r + 2$. In [GL08b] this was used as a motivation to define a new hierarchy of lower bounds $\{\Lambda_r(G)\}$ on the chromatic number that can go beyond the fractional chromatic number. In Section 8.3.3 we recall this approach and show that its extension to the tracial setting recovers the hierarchy $\{\gamma_r^{\mathrm{col}}(G)\}$ introduced in Section 8.3.1. We also show how a similar technique can be used to recover the hierarchy $\{\gamma_r^{\mathrm{stab}}(G)\}$.

A link between $\xi_r^{\mathrm{stab}}(G)$ and $\xi_r^{\mathrm{col}}(G)$

In [GL08b, Thm. 3.1] it is shown that the bounds $\mathrm{las}_r^{\mathrm{stab}}(G)$ and $\mathrm{las}_r^{\mathrm{col}}(G)$ satisfy

$$\mathrm{las}_r^{\mathrm{stab}}(G)\mathrm{las}_r^{\mathrm{col}}(G) \geq |V| \quad \text{for any } r \geq 1,$$

with equality if G is vertex-transitive. This extends a well-known property of the theta number (i.e., the case $r = 1$). The same property holds for the noncommutative analogues $\xi_r^{\mathrm{stab}}(G)$ and $\xi_r^{\mathrm{col}}(G)$.

Lemma 8.10. *For a graph $G = (V, E)$ and $r \in \mathbb{N} \cup \{\infty, *\}$ we have*

$$\xi_r^{\mathrm{stab}}(G)\xi_r^{\mathrm{col}}(G) \geq |V|,$$

with equality if G is vertex-transitive.

Proof. Let L be feasible for $\xi_r^{\mathrm{col}}(G)$. Then $\tilde{L} = L/L(1)$ provides a solution to $\xi_r^{\mathrm{stab}}(G)$ with value $\tilde{L}\big(\sum_{i\in V} x_i\big) = |V|/L(1)$, implying that $\xi_r^{\mathrm{stab}}(G) \geq |V|/L(1)$ and therefore $\xi_r^{\mathrm{stab}}(G)\xi_r^{\mathrm{col}}(G) \geq |V|$.

Assume G is vertex-transitive. Let L be a feasible solution for $\xi_r^{\mathrm{stab}}(G)$. As G is vertex-transitive we may assume (after symmetrization) that $L(x_i)$ takes a constant value. Set $L(x_i) =: 1/\lambda$ for all $i \in V$, so that the objective value of L for $\xi_r^{\mathrm{stab}}(G)$ is $|V|/\lambda$. Then $\tilde{L} = \lambda L$ provides a feasible solution for $\xi_r^{\mathrm{col}}(G)$ with value λ, implying $\xi_r^{\mathrm{col}}(G) \leq \lambda$. This shows $\xi_r^{\mathrm{col}}(G)\xi_r^{\mathrm{stab}}(G) \leq |V|$. □

For a vertex-transitive graph G, the inequality $\xi_f(G)\alpha_q(G) \leq |V|$ is shown in [MR16b, Lem. 6.5]; it can be recovered from the $r = *$ case of Lemma 8.10 and $\alpha_q(G) \leq \alpha_p(G)$.

Comparison to existing semidefinite programming bounds

By adding the constraints $L(x_i x_j) \geq 0$, for all $i, j \in V$, to the program defining $\xi_1^{\mathrm{col}}(G)$, we obtain the strengthened theta number $\vartheta^+(\overline{G})$ (from [Sze94]). Moreover, if we add the constraints

$$L(x_i x_j) \geq 0 \qquad \text{for } i \neq j \in V, \tag{8.10}$$

$$\sum_{j\in C} L(x_i x_j) \leq 1 \qquad \text{for } i \in V, \tag{8.11}$$

$$L(1) + \sum_{i\in C, j\in C'} L(x_i x_j) \geq |C| + |C'| \qquad \text{for } C, C' \text{ distinct cliques in } G \tag{8.12}$$

to the program defining the parameter $\xi_1^{\text{col}}(G)$, then we obtain the parameter $\xi_{\text{SDP}}(G)$, which is introduced in [PSS$^+$16, Thm. 7.3] as a lower bound on $\xi_{\text{tr}}(G)$. We will now show that the inequalities (8.10)–(8.12) are in fact valid for $\xi_2^{\text{col}}(G)$, which implies

$$\xi_2^{\text{col}}(G) \geq \xi_{\text{SDP}}(G) \geq \vartheta^+(\overline{G}).$$

For this, given a clique C in G, we define the polynomial

$$g_C := 1 - \sum_{i \in C} x_i \in \mathbb{R}\langle \mathbf{x} \rangle.$$

Then (8.11) and (8.12) can be reformulated as $L(x_i g_C) \geq 0$ and $L(g_C g_{C'}) \geq 0$, respectively, using the fact that $L(x_i) = L(x_i^2) = 1$ for all $i \in V$. Hence, to show that any feasible L for $\xi_2^{\text{col}}(G)$ satisfies (8.10)-(8.12), it suffices to show Lemma 8.11 below. Recall that a commutator is a polynomial of the form $[p,q] = pq - qp$ with $p, q \in \mathbb{R}\langle \mathbf{x} \rangle$. We denote by Θ_r the set of linear combinations of commutators $[p,q]$ with $\deg(pq) \leq r$.

Lemma 8.11. *Let C and C' be cliques in a graph G and let $i, j \in V$. Then we have*

$$g_C \in \mathcal{M}_2(\emptyset) + \mathcal{I}_2(\mathcal{H}_G), \text{ and } x_i x_j,\ x_i g_C,\ g_C g_{C'} \in \mathcal{M}_4(\emptyset) + \mathcal{I}_4(\mathcal{H}_G) + \Theta_4.$$

Proof. The claim $g_C \in \mathcal{M}_2(\emptyset) + \mathcal{I}_2(\mathcal{H}_G)$ follows from the identity

$$g_C = \underbrace{\Big(1 - \sum_{i \in C} x_i\Big)}_{g_C}{}^2 + \underbrace{\sum_{i \in C}(x_i - x_i^2) + \sum_{i \neq j \in C} x_i x_j}_{h} = g_C^2 + h, \tag{8.13}$$

where $h \in \mathcal{I}_2(\mathcal{H}_G)$. We also have

$$\begin{aligned} x_i x_j &= x_i x_j^2 x_i + x_j(x_i - x_i^2) + x_i^2(x_j - x_j^2) + [x_i, x_i x_j^2] + [x_i - x_i^2, x_j], \\ x_i g_C &= x_i g_C^2 x_i + g_C^2(x_i - x_i^2) + [x_i - x_i^2, g_C^2] + [x_i, x_i g_C^2], \end{aligned}$$

and, writing analogously $g_{C'} = g_{C'}^2 + h'$ with $h' \in \mathcal{I}_2(\mathcal{H}_G)$, we have

$$g_C g_{C'} = g_C g_{C'}^2 g_C + [g_C, g_C g_{C'}^2] + [h, g_{C'}^2] + g_C^2 h' + h h' + g_{C'}^2 h. \qquad \square$$

Example 8.12. Using the bound $\xi_{\text{SDP}}(G)$ it is shown in [PSS$^+$16, Thm. 7.4] that the tracial rank of the odd cycle C_{2n+1} on $2n+1$ vertices equals $(2n+1)/n$. That is, $\xi_{tr}(C_{2n+1}) = \xi_\infty^{\text{col}}(C_{2n+1}) = (2n+1)/n$. Combining this with Lemma 8.10 gives the inequality $n = \xi_\infty^{\text{stab}}(C_{2n+1}) \geq \alpha_{qc}(C_{2n+1})$. In fact, equality holds since $\alpha_{qc}(C_{2n+1}) \geq \alpha(C_{2n+1}) = n$. △

8.3.3 Links between $\gamma_r^{\text{col}}(G)$, $\xi_r^{\text{col}}(G)$, $\gamma_r^{\text{stab}}(G)$, and $\xi_r^{\text{stab}}(G)$

In this last section, we make the link between the two hierarchies $\{\xi_r^{\text{stab}}(G)\}$ (resp. $\{\xi_r^{\text{col}}(G)\}$) and $\{\gamma_r^{\text{stab}}(G)\}$ (resp. $\{\gamma_r^{\text{col}}(G)\}$). The key tool is the interpretation of the coloring and stability numbers in terms of certain graph products.

We start with the (quantum) coloring number. For an integer k, recall that the Cartesian product $G\square K_k$ of G and the complete graph K_k is the graph with vertex set $V\times[k]$, where two vertices (i,c) and (j,c') are adjacent if $(\{i,j\}\in E$ and $c=c')$ or $(i=j$ and $c\neq c')$. The following is a well-known reduction of the chromatic number $\chi(G)$ to the stability number of the Cartesian product $G\square K_k$:

$$\chi(G)=\min\big\{k\in\mathbb{N}:\alpha(G\square K_k)=|V|\big\}.$$

It was used in [GL08b] to define the following lower bounds on the chromatic number:

$$\Lambda_r(G)=\min\big\{k\in\mathbb{N}:\text{las}_r^{\text{stab}}(G\square K_k)=|V|\big\},$$

where it was also shown that $\text{las}_r^{\text{col}}(G)\leq\Lambda_r(G)\leq\chi(G)$ for all $r\geq 1$, with equality $\Lambda_{|V|}(G)=\chi(G)$. Hence the bounds $\Lambda_r(G)$ may go beyond the fractional chromatic number. This is the case for the above-mentioned Kneser graphs; see [GL08a] for other graph instances.

The above reduction from coloring to stability number has been extended to the quantum setting in [MR16b], where it is shown that

$$\chi_q(G)=\min\{k\in\mathbb{N}:\alpha_q(G\square K_k)=|V|\}.$$

It is therefore natural to use the upper bounds $\xi_r^{\text{stab}}(G\square K_k)$ on $\alpha_q(G\square K_k)$ in order to get the following lower bounds on the quantum coloring number:

$$\min\{k:\xi_r^{\text{stab}}(G\square K_k)=|V|\},\tag{8.14}$$

which are thus the noncommutative analogues of the bounds $\Lambda_r(G)$.

Observe that, for any $k\in\mathbb{N}$ and $r\in\mathbb{N}\cup\{\infty,*\}$, we have $\xi_r^{\text{stab}}(G\square K_k)\leq|V|$, which follows from Lemma 8.11 and the fact that the cliques $C_i=\{(i,c):c\in[k]\}$, for $i\in V$, cover all vertices in $G\square K_k$. Let

$$\mathcal{C}_{G\square K_k}=\big\{g_{C_i}:i\in V\big\},\quad\text{where }\ g_{C_i}=1-\sum_{c\in[k]}x_i^c,$$

denote the set of polynomials corresponding to these cliques. We now show that the parameter (8.14) in fact coincides with the parameter $\gamma_r^{\text{col}}(G)$ for all $r\in\mathbb{N}\cup\{\infty\}$.

For this observe first that the quadratic polynomials in the set $\mathcal{H}_{G,k}^{\text{col}}$ correspond precisely to the edges of $G\square K_k$, and that the projector constraints are included in $\mathcal{I}_2(\mathcal{H}_{G,k}^{\text{col}})$ (see (8.6)). Hence we have

$$\mathcal{I}_{2r}(\mathcal{H}_{G,k}^{\text{col}})=\mathcal{I}_{2r}(\mathcal{H}_{G\square K_k}\cup\mathcal{C}_{G\square K_k}).\tag{8.15}$$

We will also use the following result.

Lemma 8.13. *Let $r\in\mathbb{N}\cup\{\infty,*\}$ and assume L is feasible for $\xi_r^{\text{stab}}(G\square K_k)$. Then, we have $L(\sum_{i\in V,c\in[k]}x_i^c)=|V|$ if and only if $L=0$ on $\mathcal{I}_{2r}(\mathcal{C}_{G\square K_k})$.*

Proof. Assume $L=0$ on $\mathcal{I}_{2r}(\mathcal{C}_{G\square K_k})$. Then $0=\sum_{i\in V}L(g_{C_i})=|V|-L(\sum_{i,c}x_i^c)$.

Conversely assume that $0=L\big(\sum_{i\in V,c\in[k]}x_i^c\big)-|V|=\sum_{i\in V}L(g_{C_i})$. We will show $L=0$ on $\mathcal{I}_{2r}(\mathcal{C}_{G\square K_k})$. For this we first observe that $g_{C_i}-(g_{C_i})^2\in\mathcal{I}_2(\mathcal{H}_{G\square K_k})$

by (8.13). Hence $L(g_{C_i}) = L(g_{C_i}^2) \geq 0$, which, combined with $\sum_i L(g_{C_i}) = 0$, implies $L(g_{C_i}) = 0$ for all $i \in V$. Next we show $L(wg_{C_i}) = 0$ for all words w with degree at most $2r-1$, using induction on $\deg(w)$. The base case $w = 1$ holds by the above. Assume now $w = uv$, where $\deg(v) < \deg(u) \leq r$. Using the positivity of L, the Cauchy-Schwarz inequality gives $|L(uvg_{C_i})| \leq L(u^*u)^{1/2} L(v^* g_{C_i}^2 v)^{1/2}$. Note that it suffices to show $L(v^* g_{C_i} v) = 0$ since, using again (8.13), this implies $L(v^* g_{C_i}^2 v) = 0$ and thus $L(uvg_{C_i}) = 0$. We have $\deg(vv^*) < \deg(w)$, and therefore, using the tracial property of L and the induction assumption, we see that $L(v^* g_{C_i} v) = L(vv^* g_{C_i}) = 0$. □

Proposition 8.14. *For every graph G and $r \in \mathbb{N} \cup \{\infty\}$ we have*

$$\gamma_r^{\text{col}}(G) = \min\{k : \xi_r^{\text{stab}}(G \square K_k) = |V|\}.$$

Proof. Let L be a linear functional certifying $\gamma_r^{\text{col}}(G) \leq k$. Then, using (8.15) we see that L is feasible for $\xi_r^{\text{stab}}(G \square K_k)$ and Lemma 8.13 shows that $L(\sum_{i,c} x_i^c) = |V|$. This shows $\xi_r^{\text{stab}}(G \square K_k) \geq |V|$ and thus equality holds (since the reverse inequality always holds). Therefore, $\min\{k : \xi_r^{\text{stab}}(G \square K_k) = |V|\} \leq k$.

Conversely, assume $\xi_r^{\text{stab}}(G \square K_k) = |V|$. Since the optimum is attained, there exists a linear functional L feasible for $\xi_r^{\text{stab}}(G \square K_k)$ with $L(\sum_{i,c} x_i^c) = |V|$. Using Lemma 8.13 we can conclude that L is zero on $\mathcal{I}_{2r}(\mathcal{C}_{G \square K_k})$. Hence, in view of (8.15), L is zero on $\mathcal{I}_{2r}(\mathcal{H}_{G,k}^{\text{col}})$. This shows $\gamma_r^{\text{col}}(G) \leq k$. □

Note that the proof of Proposition 8.14 also works in the commutative setting; this shows that the sequence $\Lambda_r(G)$ corresponds to the usual Lasserre hierarchy for the feasibility problem defined by the equations (8.2)–(8.3), which is another way of showing $\Lambda_\infty(G) = \chi(G)$.

We now turn to the (quantum) stability number. For $k \in \mathbb{N}$, consider the graph product $K_k \star G$, with vertex set $[k] \times G$, and with an edge between two vertices (c,i) and (c',j) when $(c \neq c', i = j)$ or $(c = c', i \neq j)$ or $(c \neq c', \{i,j\} \in E)$. The product $K_k \star G$ coincides with the homomorphic product $K_k \ltimes \overline{G}$ used in [MR16b, Sec. 4.2], where it is shown that

$$\alpha_q(G) = \max\{k \in \mathbb{N} : \alpha_q(K_k \star G) = k\}.$$

This suggests using the upper bounds $\xi_r^{\text{stab}}(K_k \star G)$ on $\alpha_q(K_k \star G)$ to define the following upper bounds on $\alpha_q(G)$:

$$\max\{k \in \mathbb{N} : \xi_r^{\text{stab}}(K_k \star G) = k\}. \tag{8.16}$$

For each $c \in [k]$, the set $C^c = \{(c,i) : i \in V\}$ is a clique in $K_k \star G$, and we let

$$\mathcal{C}_{K_k \star G} = \{g_{C^c} : c \in [k]\}, \quad \text{where } g_{C^c} = 1 - \sum_{i \in V} x_c^i,$$

denote the set of polynomials corresponding to these cliques. As these k cliques cover the vertex set of $K_k \star G$, we can use Lemma 8.11 to conclude that $\xi_r^{\text{stab}}(K_k \star G) \leq k$ for all $r \in \mathbb{N} \cup \{\infty, *\}$.

Again, observe that the quadratic polynomials in the set $\mathcal{H}_{G,k}^{\text{stab}}$ correspond precisely to the edges of $K_k \star G$ and that we have

$$\mathcal{I}_{2r}(\mathcal{H}_{G,k}^{\text{stab}}) = \mathcal{I}_{2r}(\mathcal{H}_{K_k \star G} \cup \mathcal{C}_{K_k \star G}).$$

Based on this, one can show the analogue of Lemma 8.13: If L is feasible for the program $\xi_r^{\text{stab}}(K_k \star G)$, then we have $L(\sum_{i,c} x_c^i) = k$ if and only if $L = 0$ on $\mathcal{I}_{2r}(\mathcal{C}_{K_k \star G})$. This lemma can be used to show the following result, whose proof is analogous to that of Proposition 8.14 and thus omitted.

Proposition 8.15. *For every graph G and $r \in \mathbb{N} \cup \{\infty\}$ we have*

$$\gamma_r^{\text{stab}}(G) = \max\{k : \xi_r^{\text{stab}}(K_k \star G) = k\}.$$

We do not know whether the results of Propositions 8.14 and 8.15 hold for $r = *$, because we do not know whether the supremum is attained in the program defining the parameter $\xi_*^{\text{stab}}(\cdot) = \alpha_p(\cdot)$ (as was already observed in [Rob13, p. 120]). Hence we can only claim the inequalities

$$\gamma_*^{\text{col}}(G) \geq \min\{k : \xi_*^{\text{stab}}(G \Box K_k) = |V|\} \text{ and } \gamma_*^{\text{stab}}(G) \leq \max\{k : \xi_*^{\text{stab}}(K_k \star G) = k\}.$$

As mentioned above, we have $\text{las}_r^{\text{col}}(G) \leq \Lambda_r(G)$ for every integer $r \in \mathbb{N}$ [GL08b, Prop. 3.3]. This result extends to the noncommutative setting and the analogous result holds for the stability parameters. In other words the hierarchies $\{\gamma_r^{\text{col}}(G)\}$ and $\{\gamma_r^{\text{stab}}(G)\}$ refine the hierarchies $\{\xi_r^{\text{col}}(G)\}$ and $\{\xi_r^{\text{stab}}(G)\}$.

Proposition 8.16. *For every graph G and $r \in \mathbb{N} \cup \{\infty, *\}$ we have*

$$\xi_r^{\text{col}}(G) \leq \gamma_r^{\text{col}}(G) \text{ and } \xi_r^{\text{stab}}(G) \geq \gamma_r^{\text{stab}}(G).$$

Proof. We may restrict to $r \in \mathbb{N}$ since we have seen earlier that the inequalities hold for $r \in \{\infty, *\}$. The proof for the coloring parameters is similar to the proof of [GL08b, Prop. 3.3] in the classical case and thus we omit it. We now show $\xi_r^{\text{stab}}(G) \geq \gamma_r^{\text{stab}}(G)$. Set $k = \gamma_r^{\text{stab}}(G)$ and, using Proposition 8.15, let $L \in \mathbb{R}\langle x_c^i : i \in V, c \in [k]\rangle_{2r}^*$ be optimal for $\xi_r^{\text{stab}}(K_k \star G) = k$. That is, L is tracial, symmetric, positive, and satisfies $L(1) = 1$, $L(\sum_{i,c} x_c^i) = k$, and $L = 0$ on $\mathcal{I}(\mathcal{H}_{K_k \star G})$. It suffices now to construct a tracial symmetric positive linear form $\hat{L} \in \mathbb{R}\langle x_i : i \in V\rangle_{2r}^*$ such that $\hat{L}(1) = 1$, $\hat{L}(\sum_{i \in V} x_i) = k$, and $\hat{L} = 0$ on $\mathcal{I}_{2r}(\mathcal{H}_G)$, since this will imply $\xi_r^{\text{stab}}(G) \geq k$. For this, for any word $x_{i_1} \cdots x_{i_t}$ with degree $1 \leq t \leq 2r$, we define $\hat{L}(x_{i_1} \cdots x_{i_t}) := \sum_{c \in [k]} L(x_c^{i_1} \cdots x_c^{i_t})$, and we set $\hat{L}(1) = L(1) = 1$. Then, we have $\hat{L}(\sum_{i \in V} x_i) = k$. Moreover, one can easily check that $\hat{L}$ is indeed tracial, symmetric, positive, and vanishes on $\mathcal{I}_{2r}(\mathcal{H}_G)$. □

8.4 Discussion

Let us discuss some known separations between the quantum graph parameters and their classical analogues.

The separations between $\chi_q(G)$ and $\chi(G)$, and between $\alpha_q(G)$ and $\alpha(G)$, can be exponentially large in the number of vertices. This is the case for the graphs with vertex set $\{\pm 1\}^N$ for N a multiple of 4, where two vertices are adjacent if they are orthogonal [AHKS06, MR16b, MSS13]. These graphs are often called Hadamard graphs (notice that a clique of size N corresponds to a real Hadamard matrix, see Section 2.2).

Let us explain the separation between $\chi_q(G)$ and $\chi(G)$ for these graphs. Let $N \in \mathbb{N}$ and let $G_N = (V, E)$ be the graph where $V = \{\pm 1\}^N$ and

$$E = \big\{\{x, y\} \in V \times V : \langle x, y\rangle = \sum_{i=1}^N x_i y_i = 0\big\}.$$

We will first show that $\chi_q(G_N) \leq N$, for this we follow the argument given in [AHKS06].[2] We will construct a perfect strategy $P \in C_{q,s}([N]^2 \times V^2)$ for the quantum coloring game. We use the state $\psi = \frac{1}{\sqrt{N}} \sum_{i=1}^N e_i \otimes e_i \in \mathbb{C}^N \otimes \mathbb{C}^N$. To describe Alice's and Bob's POVMs it will be usefull to consider the unitary matrix $\Omega_N = \frac{1}{\sqrt{N}}(\omega_N^{ij})_{i,j\in[N]}$, where $\omega_N = e^{2\pi \mathbf{i}/N}$ is the Nth root of unity. This matrix is known as the *discrete Fourier transform.* For each question $x \in V$, Alice has a POVM $\{A_x^i\}_{i\in[N]}$, and for each question $y \in V$, Bob has a POVM $\{B_y^i\}_{i\in[N]}$, where

$$\begin{aligned} A_x^i &= \mathrm{Diag}(x)\Omega_N^* e_i e_i^* \Omega_N \mathrm{Diag}(x) \qquad i \in [N], \\ B_y^i &= \mathrm{Diag}(y)\Omega_N e_i e_i^* \Omega_N^* \mathrm{Diag}(y) \qquad i \in [N]. \end{aligned}$$

$$e_i \mapsto \frac{1}{\sqrt{N}} \sum_{i=1}^N \omega_N^{ij} e_j,$$

The claim is that the bipartite quantum correlation P corresponding to this state and these POVMs is a perfect strategy for the coloring game. To see this, we compute the probability that Alice and Bob output i and j respectively, when they are given questions $x, y \in V$:

$$\begin{aligned} P(i, j|x, y) &= \psi^*(A_x^i \otimes B_y^j)\psi \\ &= \frac{1}{N^3}\left|\sum_{k=1}^N x_k y_k \omega_N^{k(i-j)}\right|^2. \end{aligned}$$

In particular, the probability that Alice and Bob both answer $i \in [N]$ is

$$P(i, i|x, y) = \frac{1}{N^3}\left(\sum_{k=1}^N x_k y_k\right)^2.$$

Therefore, if Alice and Bob receive adjacent vertices x and y, that is, if $\sum_{k=1}^N x_k y_k = 0$, then the probability that they both answer color i equals zero. Similarly, if Alice

[2] The same argument was given earlier for the case where N is a power of 2. See [BCT99] for the same argument in the setting of graph coloring, and see [BCW98] for a similar argument in a different setting.

and Bob receive the same vertex $x = y$, then the above shows that $P(i, i|x, x) = \frac{1}{N}$ for each $i \in [N]$. It follows that $\sum_{i=1}^{N} P(i, i|x, x) = 1$ and therefore $P(i, j|x, x) = 0$ if $i \neq j$, that is, P is synchronous. Together this shows that P is indeed a perfect strategy for $\chi_q(G)$ and therefore $\chi_q(G) \leq N$.

In fact, one can show that $\chi_q(G_N) = N$. For this, we can use the theta number of the complement $\overline{G}$ of G. Recall that $\vartheta(\overline{G}) \leq \chi_q(G)$. It was shown in [MR16b, Prop. 4.2] that if N is divisible by 4, then $\vartheta(\overline{G_N}) = N$.

Finally, it follows from a result of Frankl and Rödl [FR87, Thm. 1.11] that for large enough N divisible by 4, the chromatic number of G_N is exponential in N. Informally, the result of Frankl and Rödl implies that there are no large independent sets in G_N. Therefore, the chromatic number needs to be large. Together, this shows that the ratio between $\chi(G)$ and $\chi_q(G)$ can be exponential in the number of vertices.

What about $\alpha(G)$ and $\alpha_q(G)$? In [MR16b] it is shown that the graphs G_N can also be used to show an exponential separation between $\alpha_q(G)$ and $\alpha(G)$.

Can we also separate the quantum parameters from their commuting operator analogues? This is still an open question. While it was recently shown that the sets $C_{q,s}(\Gamma)$ and $C_{qc,s}(\Gamma)$ can be different [DPP19], it is still not known whether there is a separation between the parameters $\chi_q(G)$ and $\chi_{qc}(G)$, and between $\alpha_q(G)$ and $\alpha_{qc}(G)$.

Let us finish by noting a remarkable property of the quantum chromatic number. It is easy to see that the chromatic number of a graph increases by 1 if we add a new vertex that is adjacent to all other vertices. Surprisingly, this is not true in general for the quantum chromatic number [MR16a].

Part II

Quantum algorithms & optimization

Chapter 9

Quantum algorithms

In this background chapter we introduce the basic concepts of quantum algorithms and we give an overview of the main quantum algorithms that will be used as subroutines in the subsequent chapters. For more details see for example [NC00], or the lecture notes [Wat11, dW11].

9.1 The basics

In Chapter 3 we have seen that the state of a quantum-mechanical system can be described by a unit vector in a Hilbert space, and that the allowed operations are applications of unitary operators and measurements. A quantum algorithm consists of precisely those operations: we start with an initial state ψ, we apply a unitary U to ψ and then we perform some m-outcome projective measurement $\{E_1, \dots, E_m\}$ to $U\psi$.[1] Below we introduce some notation and concepts that allow us to talk about the complexity of a quantum algorithm.

Dirac notation. The fundamental building block of a classical Boolean circuit is a bit, which is either 0 or 1. The quantum analogue is the quantum bit, a *qubit*, a superposition over two basis states, that is, a unit vector $\psi \in \mathbb{C}^2$. It will be useful to think of the standard basis vectors of $\mathbb{C}^2$ as '0' and '1'. To emphasize this, for quantum algorithms we will use the Dirac notation for the standard basis vectors of $\mathbb{C}^d$: $|0\rangle, \dots, |d-1\rangle$. The conjugate transpose of a vector $|\psi\rangle \in \mathbb{C}^d$ is denoted by $\langle\psi|$. We will use the shorthand notation $|\psi\rangle|\phi\rangle$ for the state $|\psi\rangle \otimes |\phi\rangle$. When $|\psi\rangle$ and $|\phi\rangle$ are standard basis vectors we sometimes even use the notation $|\psi, \phi\rangle$ for $|\psi\rangle|\phi\rangle$. From now on, unless explicitly stated otherwise, a vector $|\psi\rangle$ is assumed to be a normalized state: a unit vector in some complex Hilbert space.

Quantum circuits. We can describe a classical computation by a sequence of wires, which carry bits, and logical gates which act on those bits. For example we

[1] Note that a product of unitary operators is again a unitary operator. Also, without loss of generality we may assume that all measurements are deferred until the end [NC00, Section 4.4].

can apply the NOT gate to a bit b which transforms it into $b \oplus 1$, or we can take the OR of two bits a and b which evaluates to 0 if $a = b = 0$ and to 1 otherwise. Similarly a quantum computation can be described by wires, which now carry qubits, and some elementary gates that act on them. Two important gates are the *Hadamard gate* H and the *controlled-not gate* CNOT. The Hadamard gate acts on a single qubit $|b\rangle$ (where $b \in \{0,1\}$) as

$$H|b\rangle = \frac{|0\rangle + (-1)^b|1\rangle}{\sqrt{2}},$$

and the CNOT acts on two qubits $|a\rangle|b\rangle$ (where $a, b \in \{0,1\}$) as

$$\text{CNOT}\,|a\rangle|b\rangle = a|a\rangle|b \oplus 1\rangle + (1-a)|a\rangle|b\rangle$$

(i.e., if the control-qubit $|a\rangle$ is $|0\rangle$ then CNOT acts as the identity on the second qubit, otherwise it performs the NOT gate on it). As matrices the Hadamard and CNOT gate can be represented as follows (in the standard basis):

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad \text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

For example, the Hadamard gate can be used to create a uniform superposition over all 2^n standard basis vectors of $(\mathbb{C}^2)^{\otimes n}$:

$$H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{i_1,\dots,i_n \in \{0,1\}} |i_1\rangle \cdots |i_n\rangle = \frac{1}{\sqrt{2^n}} \sum_{i \in \{0,\dots,2^n-1\}} |i\rangle.$$

As another example, we can construct the EPR-pair that we have seen in Section 3.1 (an entangled state!) from 2 qubits initialized in $|0\rangle|0\rangle$ by first applying a Hadamard gate on the first qubit and then a CNOT gate on the second qubit where the first qubit acts as the control-qubit:

$$\begin{aligned} |0\rangle|0\rangle &\xrightarrow{H\otimes I} H|0\rangle|0\rangle = \frac{1}{\sqrt{2}}|0\rangle|0\rangle + \frac{1}{\sqrt{2}}|1\rangle|0\rangle \\ &\xrightarrow{\text{CNOT}} \frac{1}{\sqrt{2}}\,\text{CNOT}\,|0\rangle|0\rangle + \text{CNOT}\,\frac{1}{\sqrt{2}}|1\rangle|0\rangle = \frac{1}{\sqrt{2}}|0\rangle|0\rangle + \frac{1}{\sqrt{2}}|1\rangle|1\rangle. \end{aligned}$$

Complexity. The AND and NOT gate mentioned above are *universal* for classical computation, meaning that any Boolean function $f : \{0,1\}^n \to \{0,1\}$ can be computed using a circuit containing only AND and NOT gates. Is there also a small universal gate set for quantum computation? One can show that the set of all 1-qubit gates (unitaries acting on a single qubit) together with the CNOT gate is universal: any unitary matrix can be written as a product of such small unitaries [NC00, Section 4.5.2]. The set of all 1-qubit gates is unfortunately rather big, but it turns out that it can be very efficiently approximated using only the Hadamard gate and the 1-qubit phase gate $R_{\pi/4}$ defined as $R_{\pi/4}|0\rangle = |0\rangle$, $R_{\pi/4}|1\rangle = e^{i\pi/4}|1\rangle$. Indeed, the

Solovay-Kitaev theorem [NC00, App. 3] implies that any single qubit gate can be approximated up to error ε using only polylog$(1/\varepsilon)$ gates from the set $\{H, R_{\pi/4}\}$.

In the subsequent chapters, when we talk about gate complexity, we count the number of 1-qubit and 2-qubit quantum gates in a circuit.

The query model. Besides applications of these simple 2-qubit gates, it is often convenient to separately count the applications of certain "black-box" unitaries. We call an application of such a unitary a *query* and we refer to such a unitary as an *oracle.*

For example, for a bitstring $x \in \{0,1\}^n$ we can think of allowing access to the quantum analogue of the classical oracle that allows you to query a bit of x. That is, we may consider the *standard bit oracle*: the unitary O_x acting on $\mathbb{C}^n \otimes \mathbb{C}^2$ that is defined by

$$O_x : |i, b\rangle \mapsto |i, b \oplus x_i\rangle \qquad \text{for } i \in [n], \text{ and } b \in \{0,1\}.$$

Such an oracle can be used to test certain properties of the string x more efficiently than on a classical computer. For example, below we show how to find an index $i \in [n]$ such that $x_i = 1$ (if such an i exists) using only $\mathcal{O}(\sqrt{n})$ applications of O_x. Classically, $\Omega(n)$ queries of the form "what is x_i?" are needed to succeed with high probability. In other words, we can more efficiently compute the OR function $\mathrm{OR}(x) = x_1 \vee x_2 \vee \cdots \vee x_n$ on a quantum computer using queries to O_x. In Section 10.4 we study the *quantum query complexity* of Boolean functions $f : \{0,1\}^n \to \{0,1\}$. We show there how to express the minimum number of queries to O_x that are needed to compute $f(x)$ as a semidefinite program.

9.2 The fundamental building blocks

In this section we provide an overview of the (quantum) complexity of some useful subroutines: unstructured search, amplitude amplification, amplitude estimation, minimum finding, and singular value transformation. This is by no means a complete list, it is merely a collection of tools that will be used in the subsequent chapters.

9.2.1 Grover search

The problem: suppose we are given an $x \in \{0,1\}^n$. Our goal is to find an index $i \in [n]$ such that $x_i = 1$ (and output that no solution exists if $|x| = \|x\|_1 = 0$).

Theorem 9.1 ([Gro96, BBHT98]). *There exists a quantum algorithm that uses $\mathcal{O}(\sqrt{n})$ queries to O_x and $\mathcal{O}(\sqrt{n}\log(n))$ other gates, and, with probability at least $2/3$, outputs an $i \in [n]$ such that $x_i = 1$ if such an i exists.*

The algorithm roughly works as follows. It first constructs a uniform superposition over all indices i: the state $\frac{1}{\sqrt{n}} \sum_{i\in[n]} |i\rangle$. This state can be written as

$\alpha|\phi\rangle + \beta|\psi\rangle$ where

$$|\phi\rangle = \sum_{i\in[n]:x_i=1} \frac{1}{\sqrt{|x|}}|i\rangle$$
$$|\psi\rangle = \sum_{i\in[n]:x_i=0} \frac{1}{\sqrt{n-|x|}}|i\rangle,$$

and $\alpha = \sqrt{\frac{|x|}{n}}, \beta = \sqrt{\frac{n-|x|}{n}}$. That is, $|\phi\rangle$ corresponds to the 'good' indices and $|\psi\rangle$ to the remaining 'bad' indices. Then it uses successive applications of the oracle O_x (to mark 'good' indices with a phase of -1) and a reflection through the subspace spanned by the uniform superposition, together called the *Grover iterate*, to boost the amplitude α on $|\phi\rangle$ to some α' which is close to 1 in modulus.[2] The latter ensures that upon measuring in the computational basis, we end up with a 'good' index.

We point out that a success probability of at least $2/3$ does not seem much, especially if it used as a subroutine in a larger quantum algorithm. But a constant success probability of at least $2/3$ can be boosted to at least $1-\delta$ with only $\mathcal{O}(\log(1/\delta))$ repetitions (here $0 < \delta < 1/2$). The same will be true for the other quantum algorithms in this section, hence we usually only state the number of queries and gates needed to achieve a constant success probability $> 1/2$.

9.2.2 Amplitude amplification

The above procedure works since we have an oracle O_x to mark 'good' indices. Therefore it can be generalized to arbitrary quantum algorithms for which we can mark 'good' solutions. This procedure is called *amplitude amplification.*

Theorem 9.2 ([BHMT02]). *Suppose that U is a quantum algorithm acting on q qubits such that $U|0\rangle^{\otimes q} = \alpha|\phi\rangle + \beta|\psi\rangle$, where $\alpha, \beta \in \mathbb{C}$ and $\langle\psi|\phi\rangle = 0$, and that we have access to the unitary $R = 2|\phi\rangle\langle\phi| - I$. Then, using $\mathcal{O}(1/|\alpha|)$ applications of U, U^* and R, and $\mathcal{O}(q/|\alpha|)$ other gates, we can create a state $\alpha'|\phi\rangle + \beta'|\psi\rangle$ such that $|\alpha'| > 2/3$.*

Grover search can be seen as a special case where U creates a uniform superposition over all indices, which has amplitude $1/\sqrt{n}$ on each of the 'good' indices, and where R is O_x. When n is a power of 2, we can use $U = H^{\otimes \log(n)}$. Note that upon measuring the uniform superposition you would expect to see each good index with probability $(1/\sqrt{n})^2 = 1/n$. If there is only one 'good' index you would therefore expect to need to repeat the procedure n times to see a good solution. Classically this is indeed the case. The above two algorithms show that $\sqrt{n}$ repetitions suffice on a quantum computer.

[2] The number of successive applications depends on $|\alpha|$. This suggests that we need to know $|\alpha|$ up front. It turns out that this is not needed [BBHT98].

9.2.3 Amplitude estimation

The number of applications of U and U^* in the above procedure depends on the initial amplitude α on the 'good' indices. This suggests that we need to know $|\alpha|$, or a good approximation of it, up front. Fortunately $|\alpha|$ can be estimated very well using a procedure called *amplitude estimation* [BHMT02]. We state here an easy corollary of amplitude estimation that we will use.

Lemma 9.3 ([vAGGdW17, Lem. 9]). *Suppose we have a unitary U acting on q qubits such that $U|0\rangle^{\otimes q} = |0\rangle|\psi\rangle + |1\rangle|\phi\rangle$ with $\|\psi\|^2 = p \geq p_{\min}$ for some known bound $p_{\min}$, and $\|\phi\|^2 = 1 - p$. Let $\mu \in (0, 1]$ be the allowed multiplicative error in our estimation of p. Then, with $\mathcal{O}\left(\frac{1}{\mu\sqrt{p_{\min}}}\right)$ uses of U and U^* and using $\mathcal{O}\left(\frac{q}{\mu\sqrt{p_{\min}}}\right)$ gates on the q qubits, we obtain a scalar $\tilde{p}$ such that $|p - \tilde{p}| \leq \mu p$ with probability at least $4/5$.*

9.2.4 Minimum-finding

Dürr and Høyer [DH96] showed how to find the minimal value of a function f from $[n]$ to $\mathbb{R}$ using only $\mathcal{O}(\sqrt{n})$ queries to f. They did so by repeatedly using Grover search to find smaller and smaller elements of the range of f. Just as amplitude amplification generalizes Grover search, we can also generalize the minimum finding procedure. Here we describe the more general minimum-finding procedure given in [vAGGdW17, App. C].

Suppose we have a unitary U which acts on the all-zero state as

$$U|0\rangle^{\otimes q} = \sum_{k\in[n]} |\psi_k\rangle|x_k\rangle,$$

where the $|\psi_k\rangle$ are unnormalized states in $(\mathbb{C}^2)^{\otimes a}$. Assume that the states $|x_k\rangle \in (\mathbb{C}^2)^{\otimes b}$ are standard basis vectors so that we can interpret x_k as a real number written down with b bits of precision. (Notice that $a + b = q$.) Let X be the random variable on $\{x_k : k \in [n]\}$ such that $\Pr(X = x_k) = \langle\psi_k|\psi_k\rangle$. Our goal is to find the minimum value among the x_k's, using queries to U and U^*. We have the following result.

Theorem 9.4 (Generalized Minimum-Finding [vAGGdW17]). *Let U be a unitary, acting on q qubits, such that $U|0\rangle^{\otimes q} = \sum_{k=1}^n |\psi_k\rangle|x_k\rangle$. Let X be the random variable on $\{x_k : k \in [n]\}$ such that $\Pr(X = x_k) = \||\psi_k\rangle\|^2$. Let $x \geq \min_k x_k$. Using $M = \mathcal{O}(1/\sqrt{\Pr(X \leq x)})$ applications of U and U^* (and $\mathcal{O}(qM)$ other gates) we can obtain an x_i from the range of X that satisfies $x_i \leq x$ with probability at least $\frac{3}{4}$.*

Our procedure can find the minimum value x_{k^*} among the x_k's that have support in the second register, using roughly $\mathcal{O}(1/\|\psi_{k^*}\|)$ applications of U and U^*. Also, upon finding the minimal value x_{k^*} the procedure actually outputs the normalized state proportional to $|\psi_{k^*}\rangle|x_{k^*}\rangle$. This immediately gives the Dürr-Høyer result as a special case, if we take U to produce $U|0\rangle = \frac{1}{\sqrt{n}} \sum_{k=1}^n |k\rangle|f(k)\rangle$ using one query to f. Unlike Dürr-Høyer, we need not assume direct query access to the individual values $f(k)$.

We finish this section with an example application of the above generalized minimum-finding procedure: we show how to estimate the minimum eigenvalue of a Hermitian matrix A.

Finding the minimal eigenvalue of a Hermitian matrix. Let A be an $n \times n$ Hermitian matrix whose rows (and hence columns) each have at most s non-zero entries. Such a matrix is called *s-sparse.* We assume sparse oracle access to A as described in Section 9.3.2 below, and will count queries to these oracles.

Lemma 9.5 ([vAGGdW17, App. C]). *Suppose $A \in \mathrm{H}^n$ is given in s-sparse form. Let $A = \sum_{j=1}^{n} \lambda_j |\phi_j\rangle\langle\phi_j|$ be the spectral decomposition of A (which need not be known to the algorithm), with eigenvalues $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$. Suppose we are given a constant $K \in \mathbb{R}$ such that $\max_j |\lambda_j| \leq K$, and a precision $\varepsilon \in \mathbb{R}$ that satisfies $0 < \varepsilon \leq K/2$. Then we can obtain an estimate $\lambda \in \mathbb{R}$ that satisfies $|\lambda_1 - \lambda| \leq \varepsilon$ with probability at least $2/3$, using*

$$\mathcal{O}\left(\frac{Ks\sqrt{n}}{\varepsilon}\log^2\left(\frac{Kn}{\varepsilon}\right)\right) \text{ queries to } A \text{ and } \mathcal{O}\left(\frac{Ks\sqrt{n}}{\varepsilon}\log^{\frac{9}{2}}\left(\frac{Kn}{\varepsilon}\right)\right) \text{ gates.}$$

Proof idea. Here we only provide the general idea of the proof, for the details we refer to [vAGGdW17, App. C]. The general idea is as follows. We construct a unitary U which maps the all-zero state to $\sum_{k=1}^{n} |\psi_k\rangle|\lambda_k\rangle$ where the $|\psi_k\rangle$ are unnormalized states and $|\lambda_k\rangle$ is a binary encoding of λ_k, and then we apply the generalized minimum-finding procedure of Theorem 9.4.

The quantum algorithm acts on several registers, the first three of which are relevant to explain the main ideas and for the sake of exposition we ignore the other registers. The first two are $\log(n)$-qubit registers each. The last register will eventually correspond to the states $|\lambda_k\rangle$; it will contain sufficiently precise binary encodings of the eigenvalues λ_k, say using b bits (the precise value of b depends on the required precision ε). The algorithm works as follows. We first prepare the maximally entangled state on the first two registers, that is, we perform a unitary that acts as

$$|0\rangle^{\log(n)}|0\rangle^{\log(n)}|0\rangle^{b} \mapsto \frac{1}{\sqrt{n}}\sum_{j=0}^{n-1}|j\rangle|j\rangle|0\rangle^{b}.$$

We then use the invariance of maximally entangled states under transformations of the form $W \otimes \overline{W}$ for any $n \times n$ unitary W:[3]

$$\frac{1}{\sqrt{n}}\sum_{j=0}^{n-1}|j\rangle|j\rangle = \frac{1}{\sqrt{n}}\sum_{j=1}^{n}|\phi_j\rangle|\overline{\phi}_j\rangle,$$

and therefore

$$\frac{1}{\sqrt{n}}\sum_{j=0}^{n-1}|j\rangle|j\rangle|0\rangle^{b} = \frac{1}{\sqrt{n}}\sum_{j=1}^{n}|\phi_j\rangle|\overline{\phi}_j\rangle|0\rangle^{b}.$$

[3] To see this invariance, note that $\sum_{j=0}^{n-1}|j\rangle|j\rangle = \mathrm{vec}(I) = \mathrm{vec}(WW^*) = \sum_{j=1}^{n}\mathrm{vec}(w_j w_j^*) = \sum_{j=1}^{n}|w_j\rangle|\overline{w}_j\rangle$, where w_j is the jth column of W.

The state now contains the eigenvectors of A in the first register. We now apply a technique called *phase-estimation* (see, e.g., [NC00, Sec. 5.2]). Given a unitary U and an eigenvector $|u\rangle$ of U (and some work space), this technique allows to obtain an estimate $\tilde{\lambda}$ of the real value λ for which $U|u\rangle = e^{2\pi \mathbf{i}\lambda}|u\rangle$. For now let us assume that we have access to the unitary $e^{\pi \mathbf{i} A/K}$. Then our quantum algorithm continues by applying phase-estimation with respect to $e^{\pi \mathbf{i} A/K}$ to the first register, where we write the approximate phase $\tilde{\lambda}$ in the third register. That is, we perform the operation

$$\frac{1}{\sqrt{n}}\sum_{j=1}^{n}|\phi_j\rangle|\overline{\phi}_j\rangle|0\rangle^b \mapsto \frac{1}{\sqrt{n}}\sum_{j=1}^{n}|\phi_j\rangle|\overline{\phi}_j\rangle|\tilde{\lambda}_j\rangle.$$

Let us use U to denote the resulting unitary that acts on the all-zero state as

$$|0\rangle^{\log(n)}|0\rangle^{\log(n)}|0\rangle^b \mapsto \frac{1}{\sqrt{n}}\sum_{j=1}^{n}|\phi_j\rangle|\overline{\phi}_j\rangle|\tilde{\lambda}_j\rangle.$$

Notice that the latter state is precisely of the form required for the generalized minimum-finding procedure: the states $\frac{1}{\sqrt{n}}|\phi_j\rangle|\overline{\phi}_j\rangle$ are sub-normalized and the remaining register contains a binary encoding of $\tilde{\lambda}_k$, which we want to minimize over. Our algorithm thus computes the smallest eigenvalue of A by applying the generalized minimum-finding procedure with the unitary U.

It remains to show how to approximately implement the unitary $e^{\pi \mathbf{i} A/K}$, and how to account for the approximation errors coming from phase estimation and our imperfect implementation of $e^{\pi \mathbf{i} A/K}$. In the next section we will show how to apply the norm-decreasing operator e^{-A} for positive semidefinite matrices A, using similar techniques one can approximately implement $e^{\pi \mathbf{i} A/K}$. □

We note that a similar result was shown by Poulin and Wocjan [PW09], but, they assume access to (an approximation of) the unitary $e^{\pi \mathbf{i} A/K}$ instead of sparse access to A as we do here.

9.3 Matrix arithmetics using block-encodings

Suppose we are given a Hermitian matrix A and a univariate polynomial p and we wish to construct the matrix $p(A)$. There are several ways to construct $p(A)$. First, one can compute the matrices $A, A^2, \ldots, A^{\deg(p)}$ and then take the appropriate linear combination of these matrices to obtain $p(A)$. A second way is to compute the spectral decomposition $A = \sum_i \lambda_i v_i v_i^*$ and apply p to the eigenvalues to obtain $p(A) = \sum_i p(\lambda_i) v_i v_i^*$. The notions coincide. Each approach takes a polynomial number of arithmetic operations in the size of the matrix A. In this section we show how to compute $p(A)$ more efficiently on a quantum computer given a special kind of access to A (a 'block-encoding'), and we show how to efficiently create such access to A if the matrix A is sparse.

9.3.1 Singular value transformations

Suppose we have an operator A that acts on q qubits and satisfies $\|A\| \leq 1$. In what follows we let $n = 2^q$, so that A is an $n \times n$ matrix. Now suppose we want to perform the operation that maps states of the form $|0\rangle|\phi\rangle \in \mathbb{C}^2 \otimes \mathbb{C}^n$ to states $|0\rangle A|\phi\rangle + |1\rangle|\psi\rangle$ where we don't really care about how the sub-normalized state $|\psi\rangle$ looks like. This can be done by applying a $(q+1)$-qubit unitary that looks like

$$U = \begin{pmatrix} A & * \\ * & * \end{pmatrix}, \tag{9.1}$$

where the $*$'s represent $n \times n$ matrices, to the state $|0\rangle|\phi\rangle$. Since U encodes A in its top-left block, we call such a unitary a *block-encoding* of A. A block-encoding of A exists since $\|A\| \leq 1$. A block-encoding of A allows us to learn properties of A such as for instance the trace of A^*A: after applying U to the first $q+1$ qubits of the $(2q+1)$-qubit state $|0\rangle\frac{1}{\sqrt{n}}\sum_{i\in[n]}|i\rangle|i\rangle$, the state becomes

$$|0\rangle\frac{1}{\sqrt{n}}\sum_{i\in[n]} A|i\rangle|i\rangle,$$

and the probability of measuring '0' in the first qubit is given by $\mathrm{Tr}(A^*A)/n$.

Now suppose we have a block-encoding of A and some function $f : \mathbb{R} \to \mathbb{R}$ and we want to construct a block-encoding of $f(A)$. Here f acts on the singular values of A, that is, $f(A) = \sum_i f(\lambda_i)u_i v_i^*$ if $A = \sum_i \lambda_i u_i v_i^*$ is the singular value decomposition of A. Can we construct a block-encoding of $f(A)$ efficiently? The answer is yes to a certain extent: if f is sufficiently nice we can do so up to an additive and multiplicative error. To make this more precise we will need to generalize the notion of a block-encoding to allow for error, scaling, and the use of more auxiliary qubits. An alternative way to say that the unitary U from Equation (9.1) is a block-encoding is the following:

$$A = \begin{pmatrix} I_n \\ 0 \end{pmatrix}^T \begin{pmatrix} A & * \\ * & * \end{pmatrix} \begin{pmatrix} I_n \\ 0 \end{pmatrix} = (\langle 0| \otimes I)U(|0\rangle \otimes I).$$

In the definition below we allow the state $|0\rangle$ to be an a-qubit state instead of a single qubit state; and we allow both a multiplicative error (α) and an additive error (ε).

Definition 9.6 ([GSLW18, Def. 43]). *Suppose A is a q-qubit operator, $\alpha, \varepsilon \in \mathbb{R}_+$ and $a \in \mathbb{N}$, then the $(q+a)$-qubit unitary U is an (α, a, ε)*-block-encoding of A *if*

$$\|A - \alpha(\langle 0|^{\otimes a} \otimes I)U(|0\rangle^{\otimes a} \otimes I)\| \leq \varepsilon.$$

It follows that the unitary U of Eq. (9.1) is a $(1,1,0)$-block-encoding of A. A particularly useful result of Gilyén et al. [GSLW18] states that if we have a univariate polynomial p that is bounded in absolute by $1/2$ on the interval $[-1,1]$, then we can construct a block-encoding of $p(A)$ given access to a block-encoding of A.

Theorem 9.7 ([GSLW18, Thm. 56]). *Suppose that U is an (α, a, ε)-block-encoding of a Hermitian matrix A. Let $p \in \mathbb{R}[x]$ be a degree-d univariate polynomial such that $|p(x)| \leq 1/2$ for all $x \in [-1, 1]$. Given $\delta \geq 0$, there exists a quantum circuit (i.e., a unitary), which is a $(1, a+2, 4d\sqrt{\varepsilon/\alpha} + \delta)$-block-encoding of $p(A/\alpha)$. The circuit consists of d applications of U and U^*, a single application of a controlled-U gate, and $\mathcal{O}((a+1)d)$ other gates. Moreover, we can compute a description of such a circuit with a classical computer in time $\mathcal{O}(\mathrm{poly}(d, \log(1/\delta))$.*

We next record the useful fact that block-encodings of Hermitian matrices A and B can be combined to produce a block-encoding of AB.

Lemma 9.8 ([GSLW18, Lem. 53]). *Suppose that U is an (α, a, δ)-block-encoding of a q-qubit operator A, and V is a (β, b, ε)-block-encoding of a q-qubit operator B. Then, considering the Hilbert space $\mathbb{C}^{2^q} \otimes \mathbb{C}^{2^a} \otimes \mathbb{C}^{2^b}$, successively applying V to the first and third register, and U to the first and second, yields an $(\alpha\beta, a+b, \alpha\varepsilon + \beta\delta)$-block-encoding of AB.*

9.3.2 Sparse access to matrices

We now motivate the above singular value transformation techniques by showing how they can be used to compute block-encodings of smooth functions of Hermitian matrices to which we only have sparse access. In particular, in certain regimes, we show that this can be done more efficiently than by computing the singular value decomposition of the Hermitian matrix and then applying the smooth function. Let us first define the sparse access model.

Let A be an s-sparse (i.e., having at most s non-zero entries per row/column) Hermitian matrix acting on q qubits and let $n = 2^q$. We assume sparse black-box access to the elements of A in the following way: for input $(k, \ell) \in [n] \times [s]$ we can query the location and value of the ℓth non-zero entry in the kth row of the matrix A.

In the quantum setting this means we assume access to two oracles, as described in [BCK15]. We have an oracle O_I that calculates the function $\mathrm{index}_A : [n] \times [s] \to [n]$ that for input (k, ℓ) gives the column index of the ℓth non-zero element in the kth row of A. We assume this oracle computes the index "in place":

$$O_I|k, \ell\rangle = |k, \mathrm{index}_A(k, \ell)\rangle \qquad \text{for } k \in [n], \ell \in [s]. \tag{9.2}$$

(In the degenerate case where the kth row has fewer than ℓ non-zero entries, $\mathrm{index}_A(k, \ell)$ is defined to be ℓ together with some special symbol.) We also assume we can apply the inverse of O_I. Throughout we assume that the entries of A can each be represented using b bits and we furthermore assume access to an oracle O_A that returns a b-bit binary description of the entries of A:

$$O_A|k, i, z\rangle = |k, i, z \oplus A_{ki}\rangle \qquad \text{for } k, i \in [n], z \in \{0, 1\}^b. \tag{9.3}$$

When we count queries we make no distinction between O_I and O_A. We say that we make M queries to A if the number of applications of O_I plus the number of applications of O_A is upper bounded by M.

Lemma 9.9 ([GSLW18, Lem. 48 and Thm. 30])**.** *Let $A \in \mathbb{C}^{2^q \times 2^q}$ be a Hermitian operator that is s-sparse, satisfies $\|A\| \leq 1$, and to which we have access through the oracles described in* (9.2) *and* (9.3)*, and let $\varepsilon > 0$. Then we can implement a $(2, q+4, \varepsilon)$-block-encoding of A using $\mathcal{O}(s\log(s/\varepsilon))$ queries to A and $\mathcal{O}(sq\log(s/\varepsilon))$ 2-qubit gates.*

Combining the above block-encoding of A with Theorem 9.7 allows us to efficiently compute block-encodings of polynomials of A. Naturally we can combine this with polynomial approximations of more complicated functions f to provide block-encodings of $f(A)$. We finish this chapter with two examples that will be useful in Chapter 11.

Approximating the square-root function. We show how to efficiently construct a block-encoding of the matrix $\sqrt{1 + A/4}/4$, given sparse access to A. We do so by combining Lemma 9.9 and Theorem 9.7. For the latter it is necessary to have a good polynomial approximation of the function $\sqrt{1 + x/2}/4$ on the interval $[-1, 1]$. Notice that we consider the function $\sqrt{1 + x/2}/4$ instead of $\sqrt{1 + x/4}/4$, we do so since Lemma 9.9 only provides a block-encoding of A with $\alpha = 2$. We show below that a good polynomial approximation of $\sqrt{1 + x/2}/4$ can be obtained from its Taylor expansion around 0.

We have

$$\frac{\sqrt{1 + x/2}}{4} = \frac{1}{4}\sum_{k=0}^{\infty}\binom{1/2}{k}\left(\frac{x}{2}\right)^k \quad \text{whenever } |x| \leq 1.$$

Now, from the inequality

$$\left|\binom{1/2}{k}\right| = \left|\frac{\frac{1}{2}(\frac{1}{2}-1)\cdots(\frac{1}{2}-k+1)}{k!}\right| \leq 1,$$

it follows that for $d = \log(1/\delta)$ we have that the degree-d Taylor expansion is a δ-approximation on the interval $[-1, 1]$. One can verify that each Taylor expansion of the function $\sqrt{1 + x/2}/4$ around 0 is bounded in absolute value by $1/2$ on the interval $[-1, 1]$. Therefore, for any $0 < \delta \leq 1/2$, there exists a univariate polynomial p of degree $d = \mathcal{O}(\log(1/\delta))$ such that $|p(x) - \frac{\sqrt{1+x/2}}{4}| \leq \delta$ and $|p(x)| \leq \frac{1}{2}$ for all $x \in [-1, 1]$. We may therefore apply Theorem 9.7.

Lemma 9.10. *Let $A \in \mathbb{C}^{2^q \times 2^q}$ be a Hermitian operator that is s-row-sparse, satisfies $\|A\| \leq 1$, and to which we have access through the oracles described in* (9.2) *and* (9.3)*. Let $0 < \delta < 1/2 - \frac{\sqrt{3/2}}{4}$. Then we can implement a $(1, q+6, \delta)$-block-encoding of $\frac{\sqrt{1+A/4}}{4}$ with $\widetilde{\mathcal{O}}(s\log(1/\delta))$ queries to A and $\widetilde{\mathcal{O}}(sq\log(1/\delta))$ other 2-qubit gates. Moreover, we can compute a description of such a circuit with a classical computer in time $\mathcal{O}(\mathrm{poly}(\log(1/\delta)))$.*

Proof. Let $\varepsilon, \delta' > 0$ be constants to be determined later. We want the polynomial p to be a $\delta/2$-approximation of the function $\sqrt{1 + x/2}/4$ on the interval $[-1, 1]$.

Therefore we let p be the Taylor expansion of degree $d = \mathcal{O}(\log(1/\delta))$ of the function $\sqrt{1 + x/2}/4$ around 0. We now construct a block-encoding of $\sqrt{I + A/4}/4$. First construct a $(2, q{+}4, \varepsilon)$-block-encoding of A using Lemma 9.9. Then use Theorem 9.7 with this block-encoding and the polynomial p to construct a $(1, q+6, 4d\sqrt{\varepsilon/2} + \delta' + \delta/2)$-block-encoding of $\sqrt{1 + A/4}/4$ (here the linear $\delta/2$-term in the error comes from the polynomial approximation of $\sqrt{1 + x/2}/4$ up to error $\delta/2$). Finally pick $\delta' = \delta/10$ and $\varepsilon > 0$ such that $4d\sqrt{\varepsilon/2} + \delta' \leq \delta/2$, note that $\varepsilon = \Theta(\delta^2/\log(1/\delta)^2)$ suffices. The complexity statement follows from Lemma 9.9 and Theorem 9.7. □

Approximating the exponential function. Assume we are given sparse access to a Hermitian matrix A that satisfies $0 \preceq A \preceq KI$ for some known $K \in \mathbb{R}$. We show how to efficiently construct a block-encoding of $e^{-A}/4$. We again want to use Theorem 9.7. For that we need a polynomial on the interval $[-1, 1]$ that allows us to approximate the function e^{-x} on the interval $[0, K]$. First, we use the identity

$$\exp(-A) = \exp\left(-\frac{K}{2}\left(\frac{A - KI/2}{K/2} + I\right)\right),$$

to see that it suffices to obtain a good approximation of the function $\exp(-\frac{K}{2}(x{+}1))$ on the interval $[-1, 1]$. Next, as we did before for the function $\sqrt{1 + x/2}/4$, one can show that, for $\beta > 0$, the function $e^{-\beta(x+1)}/4$ can be δ-approximated on the interval $[-1, 1]$ by its Taylor expansion of degree $\mathcal{O}(\beta + \log(1/\delta))$ around 0. We thus use this with $\beta = K/2$.

Lemma 9.11. *Let $A \in \mathbb{C}^{2^q \times 2^q}$ be a Hermitian operator that is s-sparse, satisfies $0 \preceq A \preceq KI$ for some $K \in \mathbb{R}$, and to which we have access through the oracles described in* (9.2) *and* (9.3). *Let $0 < \delta < 1/4$. Then we can implement a $(1, q{+}6, \delta)$-block-encoding of $e^{-A}/4$ with $\widetilde{\mathcal{O}}(sK\log(1/\delta))$ queries to A and $\widetilde{\mathcal{O}}(sKq\log(1/\delta))$ other 2-qubit gates. Moreover, we can compute a description of such a circuit with a classical computer in time $\mathcal{O}(\mathrm{poly}(\log(K), \log(1/\delta)))$.*

Proof. Let $\varepsilon, \delta' > 0$ be constants to be determined later. We want the polynomial p to be a $\delta/2$-approximation of the function $e^{-\frac{K}{2}(x+1)}/4$ on the interval $[-1, 1]$. Therefore we let p be the Taylor expansion of degree $d = \mathcal{O}(K + \log(1/\delta))$ of the function $e^{-\frac{K}{2}(x+1)}/4$ around 0. We now construct a block-encoding of $e^{-A}/4$. First construct a $(2, q + 4, \varepsilon)$-block-encoding of $\frac{A-K/2 \cdot I}{K/2}$ using Lemma 9.9. Then use Theorem 9.7 with this block-encoding and the polynomial p to construct a $(1, q + 6, 4d\sqrt{\varepsilon/2} + \delta' + \delta/2)$-block-encoding of $e^{-A}/4$ (here the linear $\delta/2$-term in the error comes from the polynomial approximation of $e^{-\frac{K}{2}(x+1)}/4$ up to error $\delta/2$). Finally pick $\delta' = \delta/10$ and $\varepsilon > 0$ such that $4d\sqrt{\varepsilon/2} + \delta' \leq \delta/2$, note that $\varepsilon = \Theta(\delta^2/(K + \log(1/\delta))^2)$ suffices. The complexity statement follows from Lemma 9.9 and Theorem 9.7. □

Chapter 10

Quantum query complexity and semidefinite programming

This chapter is based on the paper "Semidefinite programming formulations for the completely bounded norm of a tensor", by S. Gribling and M. Laurent [GL19].

We can try to understand the power and limitations of quantum computers by determining how efficiently they can compute Boolean functions. Let us first give an informal introduction to this topic, we refer to Section 10.4 for formal definitions. Given a Boolean function $f : \{\pm 1\}^n \to \{\pm 1\}$, how many queries to an input $x \in \{\pm 1\}^n$ do we need in order to compute $f(x)$? Here, a classical query would be of the form "what is the ith bit of x?". We allow a quantum computer to make a superposition (over $i \in [n]$) of such queries. The minimum number of queries required to succeed with error probability $\leq 1/3$ is respectively the classical and quantum query complexity of the function. A first natural question is whether there is a difference between the notions of classical and quantum query complexity. Interestingly, the answer is yes for some class of functions. In Chapter 9 we have seen that there is a quantum algorithm (Grover's search) that computes the OR function, the function that is the logical OR of n bits, using $\mathcal{O}(\sqrt{n})$ quantum queries to the input string. It is not too hard to see that $\Theta(n)$ classical queries are needed.

The study of the classical and quantum query complexity of Boolean functions has a long history, we refer to for instance the survey [BW02] and the paper [ABDK16] for more information. In that long history, several general lower bound techniques and characterizations have been developed, see Section 10.4 for an overview. In this chapter we consider a recent characterization of quantum query complexity due to Arunachalam, Briët and Palazuelos [ABP19]. Let us first give a brief, high-level, summary of our results before explaining them in more detail in Section 10.1.

In this chapter we provide a new semidefinite programming characterization of the quantum query complexity of Boolean functions. Our new SDP characterization is based on a recent result of Arunachalam, Briët and Palazuelos [ABP19]. They showed that the quantum query complexity of a Boolean function can be characterized using tensors that have a completely bounded norm of at most one. Our main result is that the completely bounded norm of a t-tensor can be computed using an SDP involving matrices of size $\mathcal{O}(n^{\lceil t/2\rceil})$ and $\mathcal{O}(n^{2\lceil t/2\rceil})$ linear constraints. As an application of our result, the quantum query complexity of a Boolean function f can be obtained by checking feasibility of some SDPs. Using the duality theory of semidefinite programming we obtain a new type of certificates for large query complexity. We show that our class of certificates encompasses the linear programming certificates corresponding to the approximate degree of f and we propose an intermediate class of certificates based on second-order cone programming.

Organization. This chapter is organized as follows. We first explain our results in Section 10.1. We introduce some notation in Section 10.2. We then prove our main result in Section 10.3. In Section 10.4 we use our main result to derive a new SDP characterization of the quantum query complexity of Boolean functions. We compare our SDP to existing SDP characterizations of quantum query complexity in Section 10.4.3. Finally, using the duality theory of semidefinite programming, we obtain a new type of certificates for large query complexity in Section 10.5.

10.1 Our results

Throughout, we let $T = (T_{i_1,\dots,i_t}) \in \mathbb{R}^{n\times\cdots\times n}$ be a t-tensor acting on $\mathbb{R}^n$. The *completely bounded norm* of T, denoted $\|T\|_{\mathrm{cb}}$, is defined as

$$\sup\Big\{\Big\|\sum_{i_1,\dots,i_t=1}^{n} T_{i_1,\dots,i_t} U_1(i_1)\cdots U_t(i_t)\Big\| \;:\; d\in\mathbb{N},\; U_j(i)\in O(d) \text{ for } i\in[n], j\in[t]\Big\}. \tag{10.1}$$

Here $\|\cdot\|$ is the operator norm and $O(d)\subseteq\mathbb{R}^{d\times d}$ is the group of $d\times d$ orthogonal matrices. Note that in (10.1) one could equivalently optimize over *complex* unitary matrices $U_j(i)$.

We show that $\|T\|_{\mathrm{cb}}$ can be expressed as the optimal value of a semidefinite program (SDP). This SDP involves matrices of size $\mathcal{O}(n^{\lceil t/2\rceil})$ and $\mathcal{O}(n^{2\lceil t/2\rceil})$ linear constraints, so that an additive ε-approximation of its optimal value can be obtained in time $\mathrm{poly}(n^t, \log(1/\varepsilon))$ (see Theorem 10.5 in Section 10.3).

To put this result in perspective, if we replace the product $U_1(i_1)\cdots U_t(i_t)$ by the Kronecker (or tensor) product $U_1(i_1)\otimes\cdots\otimes U_t(i_t)$ then we obtain the *jointly completely bounded norm* of T. It is known that there is a one-to-one correspondence between the jointly completely bounded norm of a t-tensor and the entangled bias of an associated t-partite XOR game (see, e.g., [PV16]). The latter can be computed in polynomial time when $t = 2$ [Tsi87] (as we have seen in Section 3.3.1), but it is an NP-hard problem to give any constant-factor multiplicative approximation of the entangled bias of a 3-partite XOR game [Vid16]. Hence the jointly completely bounded norm of a 3-tensor is hard to approximate up to any constant factor.

As we have mentioned before, the main motivation for our study of the completely bounded norm of a tensor comes from a connection to the quantum query complexity of Boolean functions that was recently shown in [ABP19]. In Section 10.4 we explain that connection and the corollaries of our SDP characterization of $\|T\|_{\mathrm{cb}}$ in more detail. For now, let us mention that we obtain a new SDP characterization of the quantum query complexity of Boolean functions. This is not the first SDP characterization of quantum query complexity. Previously, two other semidefinite programming characterizations of quantum query complexity were given in [BSS03, HLŠ07] using a different approach. For total functions on n bits, these two SDPs have matrix variables of size 2^n while our SDP has a matrix variable of size $\Theta(n^t)$ where t is the number of queries. Thus, for small query complexity (constant) the matrix variable in our SDP is much smaller. In Section 10.4.3 we compare the three SDPs in more detail.

Finally, we point out that the notion of completely bounded norm of a tensor considered in this chapter differs from the notion considered in the work of Watrous [Wat09].

10.2 Preliminaries

For two sets of vectors $\{x_1, \dots, x_k\}$ and $\{y_1, \dots, y_\ell\}$ we use the shorthand notation $\mathrm{Gram}(\{x_i\}, \{y_j\})$ for the Gram matrix of the $k + \ell$ vectors $x_1, \dots, x_k, y_1, \dots, y_\ell$, which has the block structure:

$$\mathrm{Gram}(\{x_i\}, \{y_j\}) = \begin{pmatrix} (\langle x_i, x_j\rangle) & (\langle x_i, y_j\rangle) \\ (\langle y_i, x_j\rangle) & (\langle y_i, y_j\rangle) \end{pmatrix}.$$

We will use the following lemma, which follows from a well-known isometry property of the Euclidean space; we give a short proof for completeness.

Lemma 10.1. *Let $x_1, \dots, x_k, y_1, \dots, y_k \in \mathbb{R}^d$ for some $d \in \mathbb{N}$. If $\langle x_i, x_j\rangle = \langle y_i, y_j\rangle$ for all $i, j \in [k]$, then there exists a matrix $U \in O(d)$ such that $Ux_i = y_i$ for all $i \in [k]$.*

Proof. We may assume that both sets $\{x_1, \dots, x_k\}$ and $\{y_1, \dots, y_k\}$ are linearly independent (since, for any $\lambda \in \mathbb{R}^k$, $\sum_i \lambda_i x_i = 0$ if and only if $\sum_i \lambda_i y_i = 0$, as $\|\sum_i \lambda_i x_i\|^2 = \|\sum_i \lambda_i y_i\|^2$). We may also assume that $k = d$ (else consider vectors $x_{k+1}, \dots, x_d \in \mathbb{R}^d$ forming an orthonormal basis of $\mathrm{Span}(x_1, \dots, x_k)^\perp$ and analogously for the y_i's). Now it follows from the assumption $(\langle x_i, x_j\rangle)_{i,j=1}^d = (\langle y_i, y_j\rangle)_{i,j=1}^d$ that the linear map U such that $Ux_i = y_i$ for $i \in [d]$ is orthogonal. □

Throughout we let e denote the all-ones vector (of appropriate size). For an integer t, we use the shorthand notation $\binom{[n]}{\leq t}$ for $\{S \subseteq [n] : |S| \leq t\}$. In what follows we use tensors, matrices, and vectors indexed by tuples $(i_1, \dots, i_t) \in [n]^t$. We will use the notation $\underline{i}$ to denote such a tuple, whose length (here t) will be clear from the context, and we let $\underline{i}\,\underline{j} = (i_1, \dots, i_t, j_1, \dots, j_s)$ denote the concatenation of two tuples $\underline{i} = (i_1, \dots, i_t)$ and $\underline{j} = (j_1, \dots, j_s)$. We may view a tensor $T \in \mathbb{R}^{n \times \cdots \times n}$ either as a map from $[n]\times\cdots\times[n]$ to $\mathbb{R}$ given by $\underline{i}\mapsto T_{\underline{i}}$, or as a multilinear form on $\mathbb{R}^n\times\cdots\times\mathbb{R}^n$ given by $(z_1,\dots,z_t)\mapsto T(z_1,\dots,z_t)=\sum_{i_1,\dots,i_t=1}^n T_{i_1,\dots,i_t}z_1(i_1)\cdots z_t(i_t)$. We use the $(n+1)$-dimensional *Lorentz cone* $\mathcal{L}^{n+1}=\{(w,v)\in\mathbb{R}\times\mathbb{R}^n : w\geq\|v\|_2\}$.

10.3 SDPs for the completely bounded norm

In this section we provide semidefinite programming reformulations of the completely bounded norm of a tensor. We first explain the main idea for building such a program, which essentially follows by using an adaptation of Lemma 10.1, and then we indicate how to design a more economical SDP, using smaller matrices and fewer constraints.

10.3.1 Basic construction of an SDP formulation

Recall that the operator norm of a matrix A is defined by $\|A\|=\max_{v:\|v\|=1}\|Av\|$, or, equivalently, by $\|A\|=\sup_{u,v:\|u\|=\|v\|=1}\langle u,Av\rangle$. Using the latter definition we can reformulate the completely bounded norm $\|T\|_{\mathrm{cb}}$ of a t-tensor T as the optimal value of the following program:

$$\begin{aligned}\|T\|_{\mathrm{cb}} \;=\; \sup\quad & \sum_{i_1,\dots,i_t=1}^{n} T_{i_1,\dots,i_t}\,\langle u, U_1(i_1)\cdots U_t(i_t)v\rangle \\ \text{s.t.}\quad & d\in\mathbb{N},\; u,v\in\mathbb{R}^d \text{ unit},\; U_j(i)\in O(d) \text{ for } i\in[n], j\in[t]\end{aligned} \tag{10.2}$$

We now show how to use Lemma 10.1 to characterize vectors that can be written as $U_1(i_1)\cdots U_t(i_t)v$, where v is a unit vector and $U_j(i)$ are orthogonal matrices, in terms of their Gram matrix.

Lemma 10.2. *Let $\{v_{\underline{i}}\}_{\underline{i}=(i_1,\dots,i_t)\in[n]^t}$ be a set of unit vectors in $\mathbb{R}^d$. There exist orthogonal matrices $U_j(i)\in O(d)$ for $j\in[t], i\in[n]$ and a unit vector $v\in\mathbb{R}^d$ such that*

$$v_{\underline{i}}=U_1(i_1)\cdots U_t(i_t)v \qquad \textit{for all } \underline{i}=(i_1,\dots,i_t)\in[n]^t,$$

if and only if

$$\langle v_{\underline{i}\,\underline{j}}, v_{\underline{i}\,\underline{k}}\rangle=\langle v_{\underline{i}'\,\underline{j}}, v_{\underline{i}'\,\underline{k}}\rangle \quad \textit{for all } \ell\in[t-1], \textit{ and indices } \underline{i},\underline{i}'\in[n]^\ell, \underline{j},\underline{k}\in[n]^{t-\ell}. \tag{10.3}$$

Proof. The 'only if' part is easy: for any indices $\underline{i}\in[n]^\ell$ and $\underline{j}=(j_{\ell+1},\dots,j_t)$, $\underline{k}=(k_{\ell+1},\dots,k_t)\in[n]^{t-\ell}$ we have

$$\langle v_{\underline{i}\,\underline{j}}, v_{\underline{i}\,\underline{k}}\rangle=\langle U_{\ell+1}(j_{\ell+1})\cdots U_t(j_t)v, U_{\ell+1}(k_{\ell+1})\cdots U_t(k_t)v\rangle,$$

which is independent of $\underline{i}$. We show the 'if' part by induction on $t\geq 1$. Assume first $t=1$ (in which case condition (10.3) is void). By assumption, the vectors $v_1,\dots,v_n$ are unit vectors. Then pick a unit vector $v\in\mathbb{R}^d$ and for each $i\in[n]$ let $U(i)\in O(d)$ be such that $U(i)v=v_i$ (which exists by Lemma 10.1). Assume now that $t \geq 2$. Fix the index $1 \in [n]$ and for any $i_1 \in [n] \setminus \{1\}$ consider the two sets of vectors

$$\{v_{1\,\underline{i}} : \underline{i} \in [n]^{t-1}\} \qquad \text{and} \qquad \{v_{i_1\,\underline{i}} : \underline{i} \in [n]^{t-1}\}.$$

Observe that it follows from condition (10.3) (case $\ell = 1$) that

$$\langle v_{1\,\underline{j}}, v_{1\,\underline{k}}\rangle = \langle v_{i_1\,\underline{j}}, v_{i_1\,\underline{k}}\rangle \qquad \text{for all } \underline{j}, \underline{k} \in [n]^{t-1}.$$

Hence we may apply Lemma 10.1: there exists an orthogonal matrix $U_1(i_1) \in O(d)$ such that

$$v_{i_1\,\underline{i}} = U_1(i_1) v_{1\,\underline{i}} \quad \text{for all } \underline{i} \in [n]^{t-1}.$$

We can now apply the induction hypothesis to the vectors $v_{1\,\underline{i}}$ ($\underline{i} \in [n]^{t-1}$). Since they satisfy (10.3) (with t replaced by $t-1$) it follows that there exist orthogonal matrices $U_2(i_2), \ldots, U_t(i_t) \in O(d)$ and a unit vector $v \in \mathbb{R}^d$ such that

$$v_{1\,\underline{i}} = U_2(i_2) \cdots U_t(i_t) v \quad \text{for all } \underline{i} \in [n]^{t-1}.$$

Combining the above two relations we obtain

$$v_{i_1,\ldots,i_t} = U_1(i_1) v_{1\,\underline{i}} = U_1(i_1) \cdots U_t(i_t) v \qquad \text{for all } \underline{i} \in [n]^{t-1}.$$

This concludes the proof. □

We are now ready to give an equivalent SDP formulation for the program (10.2). First, in view of Lemma 10.2, we can rewrite (10.2) as

$$\sup\Big\{ \sum_{\underline{i} \in [n^t]} T_{\underline{i}} \langle u, v_{\underline{i}}\rangle : d \in \mathbb{N}, u, v_{\underline{i}} \in \mathbb{R}^d \text{ unit vectors satisfying (10.3)}\Big\}. \tag{10.4}$$

Consider now the Gram matrix of the vectors $u, v_{\underline{i}}$ (for $\underline{i} \in [n]^t$):

$$X = \mathrm{Gram}(\{u\}, \{v_{\underline{i}}\}_{\underline{i} \in [n]^t}) \in \mathrm{S}_+^{1+n^t}.$$

Let $A_1, \ldots, A_{m_0} \in \mathrm{S}^{1+n^t}$ be such that the linear constraints $\mathrm{Tr}(A_i X) = 0$ (for $i \in [m_0]$) enforce condition (10.3) on X (namely, the fact that the entry $X_{\underline{i}\,\underline{j}, \underline{i}\,\underline{k}}$ does not depend on the choice of $\underline{i}$), and define the operator

$$\mathcal{A}_0(X) = (\mathrm{Tr}(A_1 X), \ldots, \mathrm{Tr}(A_{m_0} X)).$$

One can show that the number of linear constraints that is necessary to enforce condition (10.3) on X is $m_0 = \sum_{\ell=1}^{t-1} (n^\ell - 1)\binom{n^{t-\ell}}{2} \leq \binom{n^t}{2}$, where the last inequality follows from the fact that each entry in the bottom-right $n^t \times n^t$ principal submatrix of X appears in at most one equation. In addition, let the matrix $C_0(T) \in \mathrm{S}^{1+n^t}$ be the block-matrix whose first diagonal block is indexed by 0 (corresponding to u) and whose second diagonal block is indexed by the tuples $\underline{i} \in [n]^t$, defined as

$$C_0(T) = \frac{1}{2}\left(\begin{array}{c|c} 0 & \ldots\ T_{\underline{i}}\ \ldots \\ \hline \vdots & \\ T_{\underline{i}} & 0 \\ \vdots & \end{array}\right). \tag{10.5}$$

It follows that

$$\langle C_0(T), X\rangle = \sum_{\underline{i}\in[n]^t} T_{\underline{i}}\ X_{0,\underline{i}} = \sum_{\underline{i}\in[n]^t} T_{\underline{i}}\ \langle u, v_{\underline{i}}\rangle$$

is precisely the objective function in the program (10.4) (and thus of (10.2)). Consider now the following pair of primal/dual semidefinite programs:

$$\begin{aligned} \max\quad & \langle C_0(T), X\rangle & \min\quad & \langle e, \lambda\rangle \\ \text{s.t.}\quad & X \in \mathrm{S}_+^{1+n^t} & \text{s.t.}\quad & \lambda \in \mathbb{R}^{1+n^t}, y \in \mathbb{R}^{m_0} \\ & \operatorname{diag}(X) = e,\ \mathcal{A}_0(X) = \mathbf{0} & & \operatorname{Diag}(\lambda) + \mathcal{A}_0^*(y) - C_0(T) \in \mathrm{S}_+^{1+n^t} \end{aligned} \tag{10.6}$$

It follows from the above discussion that the optimal value of the primal problem equals $\|T\|_{\mathrm{cb}}$. Observe that both the primal and the dual are strictly feasible (for the primal the identity matrix provides a strictly feasible solution). Hence, strong duality holds and the optima in both primal and dual are equal and attained (justifying the use of max and min). In other words this shows:

Theorem 10.3. *The completely bounded norm $\|T\|_{\mathrm{cb}}$ of a t-tensor T acting on $\mathbb{R}^n$ is given by any of the two semidefinite programs in* (10.6)*. Moreover, the supremum in definition* (10.1) *is attained and one may restrict the optimization to size $d \leq 1 + n^t$.*

We also observe that the primal SDP in (10.6) involves a matrix of size $\mathcal{O}(n^t)$ and has $\mathcal{O}(n^{2t})$ linear constraints. Hence, for fixed constant t, the optimal values can be approximated up to an additive error ε in time polynomial in n and $\log(1/\varepsilon)$.

10.3.2 Reducing the size of the semidefinite program

To obtain a more efficient semidefinite programming representation of the completely bounded norm of a t-tensor, we fix an integer $s \in [t]$ and use the following observation:

$$\langle u, U_1(i_1)\cdots U_t(i_t)v\rangle = \langle U_s(i_s)^* \cdots U_1(i_1)^* u, U_{s+1}(i_{s+1})\cdots U_t(i_t)v\rangle.$$

We characterized in Lemma 10.2 the vectors of the form $U_{s+1}(i_{s+1})\cdots U_t(i_t)v$, where $v \in \mathbb{R}^d$ is a unit vector and $U_j(i) \in O(d)$ for $i \in [n]$ and $j \in \{s+1, \dots, t\}$, as the unit vectors $v_{\underline{b}} \in \mathbb{R}^d$ (for $\underline{b} \in [n]^{t-s}$) satisfying the condition:

$$\langle v_{\underline{i}\,\underline{j}}, v_{\underline{i}\,\underline{k}}\rangle = \langle v_{\underline{i}'\,\underline{j}}, v_{\underline{i}'\,\underline{k}}\rangle \quad \textit{for all } \ell \in [t-s-1],\ \textit{and indices } \underline{i}, \underline{i}' \in [n]^\ell, \underline{j}, \underline{k} \in [n]^{t-s-\ell}. \tag{10.7}$$

Analogously to Lemma 10.2 we have the following characterization for the vectors of the form $U_s(i_s)^* \cdots U_1(i_1)^* u$, where $u \in \mathbb{R}^d$ is a a unit vector and $U_j(i) \in O(d)$ for $i \in [n]$ and $j \in [s]$. (Note that $U^* \in O(d)$ if and only if $U \in O(d)$).

Lemma 10.4. *Let $\{u_{\underline{a}}\}_{\underline{a}=(i_1,\dots,i_s)\in[n]^s}$ be a set of unit vectors in $\mathbb{R}^d$. There exist orthogonal matrices $U_j(i) \in O(d)$ for $j \in [s], i \in [n]$ and a unit vector $u \in \mathbb{R}^d$ such that*

$$u_{\underline{a}} = U_s(i_s)\cdots U_1(i_1)u \qquad \textit{for all } \underline{a} = (i_1, \dots, i_s) \in [n]^s,$$

if and only if

$$\langle u_{\underline{j}\,\underline{i}}, u_{\underline{k}\,\underline{i}}\rangle = \langle u_{\underline{j}\,\underline{i}'}, u_{\underline{k}\,\underline{i}'}\rangle \quad \text{for all } \ell \in [s-1], \text{ and indices } \underline{i}, \underline{i}' \in [n]^{\ell}, \underline{j}, \underline{k} \in [n]^{s-\ell}. \tag{10.8}$$

We can now rewrite the program (10.2) as an SDP using matrices of size $n^s + n^{t-s}$. Indeed, by the above, program (10.2) can be equivalently rewritten as

$$\sup\Big\{ \textstyle\sum_{\underline{a}\in[n]^s, \underline{b}\in[n]^{t-s}} T_{\underline{a}\,\underline{b}} \langle u_{\underline{a}}, v_{\underline{b}}\rangle : \quad \begin{array}{l} v_{\underline{b}} \in \mathbb{R}^d \text{ unit vectors satisfying (10.7)}, \\ u_{\underline{a}} \in \mathbb{R}^d \text{ unit vectors satisfying (10.8)} \end{array} \Big\}. \tag{10.9}$$

Consider now the Gram matrix of the vectors $\{u_{\underline{a}}\}$ and $\{v_{\underline{b}}\}$:

$$X = \mathrm{Gram}(\{u_{\underline{a}}\}_{\underline{a}\in[n]^s}, \{v_{\underline{b}}\}_{\underline{b}\in[n]^{t-s}}).$$

Let $A_1, \ldots, A_{m_s} \in \mathrm{S}^{n^s+n^{t-s}}$ be such that the linear constraints $\mathrm{Tr}(A_i X) = 0$ (for $i \in [m_s]$) enforce the conditions (10.7) and (10.8), and define the operator

$$\mathcal{A}_s(X) = (\mathrm{Tr}(A_1 X), \ldots, \mathrm{Tr}(A_{m_s} X)).$$

Observe that X has size $n^s + n^{t-s}$, which is minimized when selecting $s = \lfloor t/2 \rfloor$. Moreover, the number of linear constraints satisfies

$$m_s = \sum_{\ell=1}^{s-1} (n^{\ell} - 1)\binom{n^{s-\ell}}{2} + \sum_{\ell=1}^{t-s-1} (n^{\ell} - 1)\binom{n^{t-s-\ell}}{2} \le \binom{n^s}{2} + \binom{n^{t-s}}{2} < \binom{n^t}{2}. \tag{10.10}$$

Let $C_s(T) \in \mathrm{S}^{n^s+n^{t-s}}$ be the block-matrix whose first diagonal block is indexed by tuples $\underline{a} \in [n]^s$ and whose second diagonal block is indexed by tuples $\underline{b} \in [n]^{t-s}$, given by

$$C_s(T) = \frac{1}{2}\begin{pmatrix} 0 & M(T) \\ M(T)^* & 0 \end{pmatrix}, \tag{10.11}$$

where $M(T) \in \mathbb{R}^{n^s \times n^{t-s}}$ has entries $M(T)_{\underline{a},\underline{b}} := T_{\underline{a}\,\underline{b}}$. Note that when selecting $s = 0$ the matrix in (10.11) coincides with the matrix in (10.5). It follows that

$$\langle C_s(T), X\rangle = \sum_{\underline{a}\in[n]^s,\, \underline{b}\in[n]^{t-s}} T_{\underline{a}\,\underline{b}} X_{\underline{a},\underline{b}} = \sum_{\underline{a}\in[n]^s,\, \underline{b}\in[n]^{t-s}} T_{\underline{a}\,\underline{b}} \langle u_{\underline{a}}, v_{\underline{b}}\rangle$$

is the objective function of the program (10.9) (and thus of (10.2)). Then we can define the pair of primal/dual semidefinite programs

$$\begin{array}{llll} \max & \langle C_s(T), X\rangle & \min & \langle e, \lambda\rangle \\ \text{s.t.} & X \in \mathrm{S}_+^{n^s+n^{t-s}} & \text{s.t.} & \lambda \in \mathbb{R}^{n^s+n^{t-s}}, y \in \mathbb{R}^m \\ & \mathrm{diag}(X) = e,\ \mathcal{A}_s(X) = \mathbf{0} & & \mathrm{Diag}(\lambda) + \mathcal{A}_s^*(y) - C_s(T) \in \mathrm{S}_+^{n^s+n^{t-s}} \end{array} \tag{10.12}$$

whose optimal values provide as before the completely bounded norm $\|T\|_{\mathrm{cb}}$.

Theorem 10.5. *The completely bounded norm* $\|T\|_{\rm cb}$ *of a t-tensor* T *acting on* $\mathbb{R}^n$ *is given by any of the two semidefinite programs in* (10.12). *Moreover, the supremum in definition* (10.1) *is attained and one may restrict the optimization to size* $d \leq n^s + n^{t-s}$ *for any integer* $s \in [t]$.

If we select $s = \lfloor t/2 \rfloor$ the primal program in (10.12) involves a matrix variable of size $n^{\lfloor t/2 \rfloor} + n^{\lceil t/2 \rceil}$ and it has $\mathcal{O}(n^{2\lceil t/2 \rceil})$ affine constraints. This represents a significant size reduction with respect to the program in (10.6) (corresponding to the choice $s = 0$), which involves a matrix variable of size $1 + n^t$ and $\mathcal{O}(n^{2t})$ affine constraints.

10.4 SDP characterization of the quantum query complexity of Boolean functions

In this section we illustrate the relevance of the above results through the connection established recently in [ABP19] between the completely bounded norm of tensors and the quantum query complexity of Boolean functions. After a brief recap on the quantum query complexity of Boolean functions, we give a new SDP characterization for the quantum query complexity $Q_\varepsilon(f)$ of a Boolean function f. We then compare our SDP to the known SDP characterizations. Finally we use our SDP to derive a new type of certificates for large quantum query complexity: $Q_\varepsilon(f) > t$.

10.4.1 Quantum query complexity

We are given a domain $D \subseteq \{\pm 1\}^n$ and a Boolean function $f : D \to \{\pm 1\}$. The function is called *total* when $D = \{\pm 1\}^n$ and *partial* otherwise. The task is to compute the value $f(x)$ for an input $x \in D$ while having access to x only through some oracle. The objective is to compute $f(x)$ using the smallest possible number of oracle calls on a worst-case input x, and the least such number is called the *classical/quantum query complexity* of the function f. See for instance [BW02] for a survey on query complexity and [ABDK16] for a more recent overview of the relation between classical and quantum query complexity.

In the classical case, the oracle consists of querying the value of the entry x_i for a selected index $i \in [n]$. In the quantum case, an oracle query to x is defined as an application of the *phase oracle* O_x, which is the diagonal unitary operator acting on $\mathbb{C}^{n+1}$ defined by $O_x = \mathrm{Diag}(x_1, \dots, x_n, 1)$.[1] A *t-query quantum algorithm* can be described by a Hilbert space $\mathcal{H} = \mathbb{C}^{n+1} \otimes \mathbb{C}^d$ (for some $d \in \mathbb{N}$), a sequence of unitaries $U_0, \dots, U_t$ acting on $\mathcal{H}$, two Hermitian positive semidefinite operators P_{+1}, P_{-1} on $\mathcal{H}$ satisfying $P_{+1} + P_{-1} = I$, and a unit vector $v \in \mathcal{H}$. The algorithm

[1] The above described classical oracle can be seen as applying the function $C_x : [n] \times \{\pm 1\} \to [n] \times \{\pm 1\}$, defined by $(i, b) \mapsto (i, x_i b)$, to the input $(i, 1)$, it is the *standard bit oracle* that we have seen in Section 9.1. In the quantum setting C_x corresponds to applying the operator $\mathrm{Diag}(x) \oplus -\mathrm{Diag}(x)$ which acts on $\mathbb{C}^n \otimes \mathbb{C}^2$. For this section, in the quantum setting it will be more convenient to work with the phase oracle $O_x = \mathrm{Diag}(x, 1)$ that acts on $\mathbb{C}^{n+1}$. It is well known that for quantum query algorithms the two oracles are equivalent, see also [BSS03, AAI$^+$16].

starts in the state v and alternates between applying a unitary U_j and the oracle O_x. The final state of the algorithm on input $x \in D$ is

$$\psi_x := U_t(O_x \otimes I_d)U_{t-1}(O_x \otimes I_d)U_{t-2} \cdots U_1(O_x \otimes I_d)U_0 v.$$

The algorithm concludes by measuring ψ_x with respect to the POVM $\{P_{+1}, P_{-1}\}$, which means that it outputs $+1$ with probability $\psi_x^* P_{+1} \psi_x$ and -1 with probability $\psi_x^* P_{-1} \psi_x$. The expected output of the algorithm is therefore given by

$$\psi_x^*(P_{+1} - P_{-1})\psi_x. \tag{10.13}$$

Given $\varepsilon \geq 0$ the *bounded-error quantum query complexity* of f, denoted as $Q_\varepsilon(f)$, is the smallest number of queries a quantum algorithm must make such that, for all $x \in D$, it computes $f(x)$ with probability at least $1 - \varepsilon$. The key fact that we will use later is that for a quantum algorithm which computes $f(x)$ with probability at least $1 - \varepsilon$ we have

$$|\psi_x^*(P_{+1} - P_{-1})\psi_x - f(x)| = 2\psi_x^* P_{-f(x)} \psi_x \leq 2\varepsilon \qquad \text{for all } x \in D.$$

Determining the quantum query complexity of a given function f is non-trivial. Understanding the quantum query complexity of specific functions can roughly be done in one of two ways: via the *polynomial method* from [BBC$^+$01] or via the *adversary method* from [Amb02]. Both methods have been used to provide lower bounds on the quantum query complexity. The adversary method was strengthened in [HLŠ07] to the *general adversary method*, which provides a parameter $\mathrm{ADV}^{\pm}(f)$, satisfying $Q_\varepsilon(f) = \Theta(\mathrm{ADV}^{\pm}(f))$ for any fixed $\varepsilon \in (0, \frac{1}{2})$ [HLŠ07, Rei09, Rei11, LMR$^+$11].[2] The polynomial method has been strengthened only very recently in [ABP19] and is shown there to provide an exact characterization of the quantum query complexity. Since the polynomial method is the most relevant to our work we explain it in some more detail below.

The polynomial method is based on the following observation made in [BBC$^+$01]: for any t-query quantum algorithm, Equation (10.13) in fact defines an n-variate polynomial p with degree at most $2t$ such that $p(x) = \psi_x^*(P_{+1} - P_{-1})\psi_x$ equals the expected value of the returned sign of the algorithm for an input $x \in D$. For inputs $x \in \{\pm 1\}^n \setminus D$ we can also run our quantum algorithm, it will still output a sign, so we have $|p(x)| \leq 1$ (and thus also $|p(x)| \leq 1 + 2\varepsilon$) for all $x \in \{\pm 1\}^n$. This motivated considering the *ε-approximate degree of f*, $\deg_\varepsilon(f)$, defined by

$$\begin{aligned} \deg_\varepsilon(f) = \min \quad & t \\ \text{s.t.} \quad & \exists\, n\text{-variate polynomial } p \text{ with } \deg(p) \leq t \\ & |p(x) - f(x)| \leq 2\varepsilon \quad \forall x \in D, \\ & |p(x)| \leq 1 + 2\varepsilon \quad \forall x \in \{\pm 1\}^n. \end{aligned} \tag{10.14}$$

[2] To be more precise, for all $\varepsilon \in (0, \frac{1}{2})$, we have $\frac{1-2\sqrt{\varepsilon(1-\varepsilon)}}{2}\mathrm{ADV}^{\pm}(f) \leq Q_\varepsilon(f) = \mathcal{O}(\log(1/\varepsilon)\mathrm{ADV}^{\pm}(f))$, where the first inequality is shown in [HLŠ07] and the second one in [Rei11].

Then, as shown in [BBC$^+$01], it follows from the above that the approximate degree of f provides a lower bound on $Q_\varepsilon(f)$:

$$\deg_\varepsilon(f) \leq 2Q_\varepsilon(f).$$

In [AA15] (see also [AAI$^+$16]) the observation is made that (10.13) can be used to define a $2t$-tensor $T \in \mathbb{R}^{(n+1)\times\cdots\times(n+1)}$ by using different input strings at the successive queries. More precisely, for any $(z_1,\ldots,z_{2t}) \in \mathbb{R}^{n+1}\times\ldots\times\mathbb{R}^{n+1}$, we can define

$$T(z_1,\ldots,z_{2t}) = v^*U_0^*\widetilde{O}_{z_1}\cdots\widetilde{O}_{z_t}U_t^*(P_{+1}-P_{-1})U_t\widetilde{O}_{z_{t+1}}U_{t-1}\widetilde{O}_{z_{t+2}}\cdots\widetilde{O}_{z_{2t}}U_0v, \tag{10.15}$$

where $\widetilde{O}_z = O_z \otimes I_d$, so that $T((x,1),\ldots,(x,1)) = \psi_x^*(P_{+1}-P_{-1})\psi_x$ equals the expected value of the returned sign of the quantum algorithm for all $x \in D$. Note that T is in fact bounded on the entire hypercube: T satisfies the inequalities $|T(z_1,\ldots,z_{2t})| \leq 1$ for all $z_1,\ldots,z_{2t} \in \{\pm1\}^{n+1}$. This led to the following notion of *block-multilinear approximate degree*, bm-deg$_\varepsilon(f)$, defined by

$$\begin{aligned} \text{bm-deg}_\varepsilon(f) = \min \quad & t \\ \text{s.t.} \quad & \exists\, t\text{-tensor } T \text{ acting on } \mathbb{R}^{n+1}, \\ & |T((x,1),\ldots,(x,1)) - f(x)| \leq 2\varepsilon \qquad \forall x \in D, \\ & |T(z_1,\ldots,z_t)| \leq 1 \quad \forall z_1,\ldots,z_t \in \{\pm1\}^{n+1}. \end{aligned} \tag{10.16}$$

Notice that if T is a t-tensor that is feasible for the program (10.16), then the degree-t polynomial p defined by $p(x) = T((x,1),\ldots,(x,1))$ is feasible for the program (10.14), and thus we have

$$\deg_\varepsilon(f) \leq \text{bm-deg}_\varepsilon(f) \leq 2Q_\varepsilon(f).$$

In the recent work [ABP19] it is shown that the $2t$-tensor in (10.15) in fact has completely bounded norm at most 1. In addition, the authors of [ABP19] also show the converse: the existence of a $2t$-tensor $T \in \mathbb{R}^{(n+1)\times\cdots\times(n+1)}$ that satisfies $\|T\|_{\text{cb}} \leq 1$ and $|T((x,1),\ldots,(x,1)) - f(x)| \leq 2\varepsilon$ for all $x \in D$, ensures the existence of a t-query quantum algorithm that outputs the correct sign with probability at least $1-\varepsilon$ for all $x \in D$.[3] That is, if such a $2t$-tensor exists, then $Q_\varepsilon(f) \leq t$. This leads to the notion of *completely bounded approximate degree*, cb-deg$_\varepsilon(f)$, defined by[4]

$$\begin{aligned} \text{cb-deg}_\varepsilon(f) = \min \quad & t \\ \text{s.t.} \quad & \exists\, t\text{-tensor } T \text{ acting on } \mathbb{R}^{n+1}, \\ & |T((x,1),\ldots,(x,1)) - f(x)| \leq 2\varepsilon \qquad \forall x \in D, \\ & \|T\|_{\text{cb}} \leq 1. \end{aligned} \tag{10.17}$$

[3] In [ABP19] the result is stated using a tensor $T \in \mathbb{R}^{2n\times\cdots\times2n}$. The dimension $2n$ corresponds to the fact that a controlled-phase gate (acting on $\mathbb{C}^{2n}$) is unitarily equivalent to the standard bit oracle. When we allow the quantum algorithm to use additional workspace we obtain the same query complexity measure working with the oracle $O_x = \text{Diag}(x,1)$ (acting on $\mathbb{C}^{n+1}$). See also [BSS03, AAI$^+$16].

[4] Note that [ABP19] use the same definition except that they consider the least t such that there exists a $2t$-tensor with these properties.

Notice that $\|T\|_{\mathrm{cb}} \leq 1$ implies that $|T(z_1,\dots,z_t)| \leq 1$ for all $z_1,\dots z_t \in \{\pm 1\}^{n+1}$. Therefore we have

$$\deg_\varepsilon(f) \leq \text{bm-}\deg_\varepsilon(f) \leq \text{cb-}\deg_\varepsilon(f) \leq 2Q_\varepsilon(f).$$

As mentioned above, the last inequality is in fact an equality up to rounding:

Theorem 10.6 ([ABP19, Cor. 1.5]). *For a Boolean function $f : D \to \{-1,1\}$ and $\varepsilon \geq 0$, we have*

$$Q_\varepsilon(f) = \lceil \text{cb-}\deg_\varepsilon(f)/2 \rceil.$$

The completely bounded degree thus gives a much tighter characterization of quantum query complexity than the general adversary method. Indeed, the general adversary method only characterizes $Q_\varepsilon(f)$ for $\varepsilon > 0$, and moreover it only does so up to logarithmic factors in $1/\varepsilon$ (see footnote 2 of this chapter), while the completely bounded degree is exact for all $\varepsilon \geq 0$.

10.4.2 New semidefinite reformulation

Using our earlier results in Section 10.3 about the completely bounded norm of a tensor, we can express the completely bounded approximate degree cb-$\deg_\varepsilon(f)$ using semidefinite programming. To certify the inequality $\|T\|_{\mathrm{cb}} \leq 1$ we can use the dual SDP in (10.12) as follows: $\|T\|_{\mathrm{cb}} \leq 1$ if and only if

$$\exists \lambda \in \mathbb{R}^{N_s}, y \in \mathbb{R}^{m_s} \text{ such that } \langle e, \lambda \rangle \leq 1 \text{ and } \mathrm{Diag}(\lambda) + \mathcal{A}_s^*(y) - C_s(T) \in \mathrm{S}_+^{N_s}.$$

Here, we may choose s to be any integer $0 \leq s \leq \lfloor t/2 \rfloor$, so that N_s is given by $(n+1)^s + (n+1)^{t-s}$ and m_s by (10.10) (with n replaced by $n+1$). We may then use the fact that the constraints: $|T((x,1),\dots,(x,1)) - f(x)| \leq 2\varepsilon$ for all $x \in D$, can be written as linear constraints on the coefficients of T to reformulate (10.17) using semidefinite programming. To make it more apparent that $T((x,1),\dots,(x,1))$ is a linear combination of the coefficients of T, recall that by definition $T(z,\dots,z) = \sum_{i_1,\dots,i_t=1}^{n+1} T_{i_1,\dots,i_t} z_{i_1} \cdots z_{i_t}$ for all $z \in \mathbb{R}^{n+1}$. It follows that the parameter cb-$\deg_\varepsilon(f)$ can be reformulated as the smallest integer $t \in \mathbb{N}$ for which the following SDP admits a feasible solution:

$$\begin{aligned} &\text{cb-}\deg_\varepsilon(f) = \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (10.18)\\ &\quad \min \quad t \\ &\quad \text{s.t.} \quad \exists\, t\text{-tensor } T \in \mathbb{R}^{(n+1)\times\dots\times(n+1)}, \lambda \in \mathbb{R}^{(n+1)^s+(n+1)^{t-s}}, y \in \mathbb{R}^{m_s} \\ &\qquad\qquad \Big| \sum_{i_1,\dots,i_t=1}^{n+1} T_{i_1,\dots,i_t} z_{i_1}\cdots z_{i_t} - f(x) \Big| \leq 2\varepsilon \text{ for } x \in D,\ z = (x,1) \in \{\pm 1\}^{n+1} \\ &\qquad\qquad \langle e, \lambda \rangle \leq 1 \\ &\qquad\qquad \mathrm{diag}(\lambda) + \mathcal{A}_s^*(y) - C_s(T) \in \mathrm{S}_+^{(n+1)^s+(n+1)^{t-s}}. \end{aligned}$$

Recall that, due to Theorem 10.5, we may choose s to be any integer $0 \leq s \leq \lfloor t/2 \rfloor$.

10.4.3 Relation to known SDPs for quantum query complexity

The above SDP (10.18) is not the first SDP that characterizes the quantum query complexity. The parameter $\mathrm{ADV}^{\pm}(f)$ provided by the general adversary method in [HLŠ07] mentioned in the previous section can also be written as an SDP. Even earlier, Barnum, Saks, and Szegedy [BSS03] formulated another SDP characterization for the quantum query complexity. Like ours, the SDP in [BSS03] expresses $Q_\varepsilon(f)$ as the smallest integer $t \in [n]$ for which there exist some positive semidefinite matrices satisfying some linear (in)equalities. Both the Barnum-Saks-Szegedy SDP and the general adversary method SDP are derived by considering the behavior of a quantum algorithm on pairs of different inputs; the matrix variables should be seen as the Gram matrices of vectors associated to the quantum algorithm. Instead, as explained before, our SDP fits in the framework of the polynomial method where we only consider the expected output of the quantum algorithm on different inputs. There are three main differences between these three SDP characterizations that we will highlight below.

First, solutions to either the Barnum-Saks-Szegedy SDP or the general adversary method SDP can be turned into quantum query algorithms, while a solution to our SDP only proves the existence of a quantum algorithm (it is not clear how to directly derive a quantum algorithm from it, as far as we know). We do not know how to construct a quantum algorithm from a solution to our SDP because the proof of Theorem 10.6 given in [ABP19, Cor. 1.5] relies on a factorization theorem due to Christensen and Sinclair [CS87], which, as far as we know, does not have a constructive proof.

A second difference is the size of the matrix variables involved in the various SDPs, which we have summarized in Table 10.1. We want to highlight the difference in block size between the three SDPs. Using our SDP one can certify the quantum query complexity t of a Boolean function using a single matrix of size $\Theta(n^t)$, while both the general adversary method SDP and the Barnum-Saks-Szegedy SDP use several matrix variables of size $|D|$ (which is 2^n for total functions). We mention that for $\varepsilon = 0$ our SDP for cb-$\deg_\varepsilon(f)$ simplifies: the matrix variable remains of size $\Theta(n^t)$, there is only one linear inequality, and the number of linear equalities remains $\Theta(n^{2t})$. Indeed, since the equations $\sum_{i_1,\dots,i_t=1}^{n+1} T_{i_1,\dots,i_t} z_{i_1} \cdots z_{i_t} = f(x)$ (for $x \in D, z = (x,1)$) involve $\Theta(n^t)$ real variables (the coefficients of T), there are at most $\Theta(n^t)$ linear equalities that are linearly independent, and we only need to impose linearly independent equality constraints.

A third difference is the fact that the adversary method SDP characterizes the quantum query complexity only up to a multiplicative factor, while both our SDP and the Barnum-Saks-Szegedy SDP give an exact characterization.

10.5 Lower bounds on quantum query complexity

We now turn our attention to providing lower bounds on the quantum query complexity. Given a fixed integer $t \in \mathbb{N}$, finding the smallest scalar $\varepsilon \geq 0$ such that

	# blocks	block size	# lin. ineq.	# equations
$\mathrm{ADV}^{\pm}(f)$	n	$\lvert D\rvert$	0	$\lvert f^{-1}(1)\rvert \cdot \lvert f^{-1}(0)\rvert$
BSS	$nt+2$	$\lvert D\rvert$	$\lvert D\rvert$	$\Theta(t\cdot \lvert D\rvert^2)$
$\text{cb-deg}_\varepsilon(f)$	1	$\Theta(n^t)$	$2\lvert D\rvert+1$	$\Theta(n^{2t})$

Table 10.1: A comparison of the size of the general adversary method SDP $\mathrm{ADV}^{\pm}(f)$, the Barnum-Saks-Szegedy SDP (BSS), and our SDP for $\text{cb-deg}_\varepsilon(f)$. The latter two are feasibility problems whose size depends on the number of queries t (which means we consider $\text{cb-deg}_\varepsilon(f) = 2t$). When viewed as a block-diagonal SDP, the first column specifies the number of blocks and the second one the size of the blocks, the third column gives the number of linear inequalities on entries of these blocks and the fourth one the number of linear equations.

$\deg_\varepsilon(f) \leq t$ can be expressed as a linear program. The duality theory of linear programming can therefore be used to provide tight lower bounds on the approximate degree $\deg_\varepsilon(f)$. Likewise, as we explain in this section, our SDP formulation of $\text{cb-deg}_\varepsilon(f)$ and the duality theory of semidefinite programming can be used to give tight lower bounds on $\text{cb-deg}_\varepsilon(f)$.

This section is organized as follows. We first rewrite the program expressing $\text{cb-deg}_\varepsilon(f)$ in a form that permits to give certificates for $\text{cb-deg}_\varepsilon(f) > t$. These certificates take the form of feasible solutions to a certain SDP. We show how linear programming certificates for $\deg_\varepsilon(f)$ can be seen as SDP solutions with a specific structure. We then define an intermediate class of certificates based on second-order cone programming.

10.5.1 Semidefinite programming certificates

Let $D \subseteq \{\pm 1\}^n$ and let $f : D \to \{\pm 1\}$. A certificate for $\text{cb-deg}_\varepsilon(f) > t$ can be given as follows. We now fix t and consider the following minimization problem (derived from the program (10.18), setting $s = 0$)[5]

$$\begin{aligned} \min \quad & 2\varepsilon \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (10.19)\\ \text{s.t.} \quad & T \in \mathbb{R}^{(n+1)\times\ldots\times(n+1)} \text{ a } t\text{-tensor}, \lambda \in \mathbb{R}^{1+(n+1)^t}, y \in \mathbb{R}^m, \varepsilon \in \mathbb{R} \\ & \Big| \sum_{i_1,\ldots,i_t=1}^{n+1} T_{i_1,\ldots,i_t} z_{i_1} \cdots z_{i_t} - f(x) \Big| \leq 2\varepsilon \quad \text{for all } x \in D, z = (x,1) \in \{\pm 1\}^{n+1} \\ & \langle e, \lambda \rangle \leq 1 \\ & \mathrm{diag}(\lambda) + \mathcal{A}_0^*(y) - C_0(T) \in \mathrm{S}_+^{1+(n+1)^t} \end{aligned}$$

[5] Note that we use the less efficient, but easier, SDP-formulation of the completely bounded norm (10.6). In (10.19) we could have just as easily used the more efficient formulation, but in Section 10.5.3 it will be convenient to work with the less efficient formulation.

Using semidefinite programming duality theory we can formulate its dual. After simplification the dual reads as follows:

$$\begin{aligned}
\max \quad & -w + \sum_{x\in D} \phi(x) f(x) \qquad (10.20)\\
\text{s.t.} \quad & \phi = (\phi(x))_{x\in D} \in \mathbb{R}^D, X \in \mathrm{S}_+^{1+(n+1)^t}, w \in \mathbb{R}\\
& \sum_{x\in D} |\phi(x)| = 1\\
& \operatorname{diag}(X) = w\cdot e\\
& \mathcal{A}_0(X) = \mathbf{0}\\
& X_{0,\underline{i}} = \sum_{\substack{x\in D\\ z=(x,1)}} \phi(x) z_{i_1}\cdots z_{i_t} \qquad \text{for all } \underline{i} = (i_1,\ldots,i_t) \in [n+1]^t
\end{aligned}$$

Note that there is no duality gap since the dual program (10.20) is strictly feasible. A tuple (ϕ, X, w) that forms a feasible solution to (10.20) with objective value strictly larger than 2ε is an *SDP certificate* for $\text{cb-deg}_\varepsilon(f) > t$.

We remark that, for total functions, i.e., when $D = \{\pm1\}^n$, the constraint on $X_{0,\underline{i}}$ says that $X_{0,\underline{i}}$ should be equal to a certain Fourier coefficient of the function ϕ. To see this we briefly recall the basic relevant facts of Fourier analysis on the Boolean cube; we refer to, for instance, [O'D14, Wol08] for more information.

Fourier analysis on the Boolean cube. For functions $f, g : \{\pm1\}^n \to \mathbb{R}$ we define the inner product $\langle f, g\rangle = \frac{1}{2^n}\sum_{x\in\{\pm1\}^n} f(x)g(x)$. Then the character functions $\chi_S(x) := \prod_{i\in S} x_i$ ($S \subseteq [n]$) form an orthonormal basis with respect to this inner product. Any function $f : \{\pm1\}^n \to \mathbb{R}$ can be expressed in this basis as $f(x) = \sum_{S\subseteq[n]} \widehat{f}(S)\chi_S(x)$, where $\widehat{f}(S) = \langle f, \chi_S\rangle$ are the Fourier coefficients of f.

We need one more definition in order to point out a link between the last constraint of program (10.20) and the Fourier coefficients of ϕ. For a tuple $\underline{i} = (i_1,\ldots,i_t) \in [n+1]^t$ let $S_{\underline{i}}$ denote the set of indices $k \in [n]$ that occur an odd number of times within the multiset $\{i_1,\ldots,i_t\}$. Note the identity

$$\sum_{x\in D, z=(x,1)} \phi(x) z_{i_1}\cdots z_{i_t} = \sum_{x\in D} \phi(x) \prod_{k\in S_{\underline{i}}} x_k = \sum_{x\in D} \phi(x)\chi_{S_{\underline{i}}}(x). \qquad (10.21)$$

Hence, in the case $D = \{\pm1\}^n$, we have

$$\sum_{x\in\{\pm1\}^n, z=(x,1)} \phi(x) z_{i_1}\cdots z_{i_t} = 2^n \widehat{\phi}(S_{\underline{i}}) \qquad (10.22)$$

and thus the last constraint in program (10.20) says that $X_{0,\underline{i}} = 2^n\widehat{\phi}(S_{\underline{i}})$ for all indices $\underline{i} \in [n+1]^t$.

Adding redundant inequalities to (10.19)**.** In the next section we will show how the SDP certificates for the completely bounded approximate degree generalize the linear programming certificates corresponding to the approximate degree. To do so, it will be useful to state an equivalent form of (10.20) derived by adding redundant inequalities to the primal problem. Recall that for a tensor T the norm constraint $\|T\|_{\mathrm{cb}} \leq 1$ implies that $\left|\sum_{i_1,\dots,i_t=1}^{n+1} T_{i_1,\dots,i_t} z_{i_1} \cdots z_{i_t}\right| = |T(z,\dots,z)| \leq 1$ for all $z \in \{\pm 1\}^{n+1}$. As the last two constraints of (10.19) ensure $\|T\|_{\mathrm{cb}} \leq 1$, it follows that the conditions

$$\left|\sum_{i_1,\dots,i_t=1}^{n+1} T_{i_1,\dots,i_t} z_{i_1} \cdots z_{i_t}\right| \leq 1 + 2\varepsilon \text{ for all } x \notin D, z = (x,1) \tag{10.23}$$

are redundant for (10.19). If we add these inequalities to (10.19) and then take the dual, then we obtain

$$\begin{aligned}
\max \quad & -w + \sum_{x \in D} \phi(x) f(x) - \sum_{x \notin D} |\phi(x)| \qquad\qquad (10.24)\\
\text{s.t.} \quad & \phi = (\phi(x))_{x \in \{\pm 1\}^n} \in \mathbb{R}^{\{\pm 1\}^n}, X \in \mathrm{S}_+^{1+(n+1)^t}, w \in \mathbb{R}\\
& \sum_{x \in \{\pm 1\}^n} |\phi(x)| = 1\\
& \mathrm{diag}(X) = w \cdot e\\
& \mathcal{A}_0(X) = \mathbf{0}\\
& X_{0,\underline{i}} = 2^n \widehat{\phi}(S_{\underline{i}}) \quad \text{ for all } \underline{i} = (i_1,\dots,i_t) \in [n+1]^t
\end{aligned}$$

Notice that strong duality holds between the above program (10.24) and the program defined by (10.19) and (10.23). In particular it follows that the optimal value of program (10.24) equals that of program (10.20). Using complementary slackness, we can say slightly more when the optimal value is strictly positive.

Lemma 10.7. *If the optimal value of the program* (10.24) *is strictly positive, then any optimal solution* (ϕ, X, w) *to* (10.24) *satisfies* $\phi(x) = 0$ *for* $x \notin D$.

Proof. Suppose that the above program (10.24) has an optimal solution (ϕ, X, w) with strictly positive objective value. Then, by strong duality, the program defined by (10.19) and (10.23) has an optimal solution $(T, \lambda, y, \varepsilon)$ with $\varepsilon > 0$. Since ε is strictly positive, there will be a strictly positive slack in all the inequalities (10.23). By complementary slackness this means that the variables in the dual corresponding to these inequalities must be equal to zero for an optimal solution, that is, $\phi(x) = 0$ for all $x \notin D$. □

Note that any tuple (ϕ, X, w) that is feasible for the program in (10.24) and satisfies $\phi(x) = 0$ for $x \notin D$ is in fact feasible for the program in (10.20).

10.5.2 Linear programming certificates: approximate degree

Let $f : D \to \{\pm 1\}$ be given. Given a fixed degree t, the smallest $\varepsilon \geq 0$ for which there exists a polynomial p of degree at most t that satisfies $\sup_{x \in D} |f(x) - p(x)| \leq 2\varepsilon$

and $\sup_{x\notin D}|p(x)| \leq 1+2\varepsilon$ can be determined using the following pair of linear programs:

$$
\begin{array}{llll}
\min & 2\varepsilon & \max & \displaystyle\sum_{x\in D} f(x)\phi(x) - \sum_{x\notin D}|\phi(x)| \\
\text{s.t.} & |p(x)-f(x)| \leq 2\varepsilon \quad \text{for } x\in D & \text{s.t.} & \displaystyle\sum_{x\in\{\pm1\}^n} |\phi(x)| = 1 \\
 & |p(x)| \leq 1+2\varepsilon \quad \text{for } x\notin D & & \phi(x)\in\mathbb{R} \quad \text{for } x\in\{\pm1\}^n \\
 & c=(c_S)\in\mathbb{R}^{\binom{[n]}{\leq t}}, \varepsilon\in\mathbb{R} & & \widehat{\phi}(S)=0 \quad \text{for } S\in\binom{[n]}{\leq t}. \\
 & p = \displaystyle\sum_{S\in\binom{[n]}{\leq t}} c_S\chi_S & &
\end{array}
\tag{10.25}
$$

For a fixed $\varepsilon \geq 0$, a polynomial ϕ that is a feasible solution to the maximization problem in (10.25) with objective value strictly larger than 2ε is called a *dual polynomial* for f, and it is a certificate for $\deg_\varepsilon(f) > t$. Note that by LP duality such a certificate exists whenever $\deg_\varepsilon(f) > t$. Dual polynomials have been used to give tight bounds on the approximate degree of many Boolean functions, see for example [Špa08, She13, BT13, BKT18].

Feasible solutions to the maximization problem in (10.25) provide feasible solutions to the SDP in (10.24). This gives a "direct" proof that dual polynomials give lower bounds on quantum query complexity.

Lemma 10.8. *Let $f : D \to \mathbb{R}$ and let $\phi : \{\pm1\}^n \to \mathbb{R}$ be a feasible solution to* (10.25) *with objective value strictly larger than* 2ε*, then* cb-$\deg_\varepsilon(f) > t$.

Proof. Observe that the tuple $(\phi, X=0, w=0)$ forms a feasible solution to (10.20) with objective value strictly larger than ε. Indeed, $X=0$ is positive semidefinite, it satisfies $\operatorname{diag}(X) = 0 = w\cdot e$ and $\mathcal{A}_0(X)=0$. Moreover, the condition $\widehat{\phi}(S)=0$ for all $S\in\binom{[n]}{\leq t}$ ensures that, for all $\underline{i}\in[n+1]^t$,

$$X_{0,\underline{i}} = 2^n\widehat{\phi}(S_{\underline{i}}) = 0$$

since $|S_{\underline{i}}| \leq t$. □

10.5.3 Second-order cone programming certificates

In (the proof of) Lemma 10.8 we have seen that the linear programming certificates of $\deg_\varepsilon(f) > t$ correspond to SDP certificates $(\phi, X, w) = (\phi, 0, 0)$ using the all-zeroes matrix $X=0$ in (10.24). Here we consider a more general class of SDP certificates (ϕ, X, w) where X and w still have an easy structure: those certificates for which we can take $X = \left(\begin{smallmatrix} w & v^T \\ v & wI\end{smallmatrix}\right)$ for some vector $v\in\mathbb{R}^{(n+1)^t}$ and real number w. This is based on the following observation.

Lemma 10.9. *Let $w \in \mathbb{R}$ and $v \in \mathbb{R}^{(n+1)^t}$. The matrix $X = \begin{pmatrix} w & v^T \\ v & wI \end{pmatrix}$ satisfies $\mathcal{A}(X) = 0$. Moreover, $X \in \mathrm{S}_+^{1+(n+1)^t}$ if and only if $w \geq \|v\|_2$.*

Proof. First note that $\mathcal{A}_0(X) = 0$ is trivially satisfied by X. Indeed, $\mathcal{A}_0$ ignores the first row and column of X and, for all $\ell \in [t-1]$, $\underline{i}, \underline{i}' \in [n+1]^\ell, \underline{j}, \underline{k} \in [n+1]^{t-\ell}$, we have that

$$X_{\underline{i}\,\underline{j},\underline{i}\,\underline{k}} = X_{\underline{i}'\,\underline{j},\underline{i}'\,\underline{k}} = \begin{cases} w & \text{if } \underline{j} = \underline{k} \\ 0 & \text{else.} \end{cases}$$

Second, by considering the Schur complement of X with respect to its upper-left corner, we have $X \in \mathrm{S}_+^{1+(n+1)^t}$ if and only if either $X = 0$, or $w > 0$ and $w - v^T v/w \geq 0$. □

By restricting our attention to feasible solutions of the above form, the program (10.24) reduces to the following second-order cone program:

$$\begin{aligned} \max \quad & -w + \sum_{x \in D} \phi(x) f(x) - \sum_{x \notin D} |\phi(x)| \\ \text{s.t.} \quad & \phi = (\phi(x))_{x \in \{\pm 1\}^n} \in \mathbb{R}^{\{\pm 1\}^n}, w \in \mathbb{R} \\ & \sum_{x \in \{\pm 1\}^n} |\phi(x)| = 1, \quad w \geq 2^n \sqrt{\sum_{\underline{i} \in [n+1]^t} \widehat{\phi}(S_{\underline{i}})^2} \end{aligned} \tag{10.26}$$

This second-order cone program involves the $(1 + (n+1)^t)$-dimensional Lorentz cone. However, by counting the number of tuples $\underline{i}$ for which $S_{\underline{i}}$ equals a given set S we can reduce to dimension $\left|\binom{[n]}{\leq t}\right| = \sum_{k=0}^t \binom{n}{k}$. Indeed, for each subset $S \subseteq [n]$ with $|S| \leq t$ let $\mathcal{I}_S$ denote the set of tuples $\underline{i} \in [n+1]^t$ for which $S_{\underline{i}} = S$. One can verify that

$$|\mathcal{I}_S| = \sum_{\substack{k_1, \ldots, k_n \in \mathbb{N}, \\ \sum_i k_i \leq t, \\ k_l \text{ is odd for } l \in S, \\ k_l \text{ is even for } l \notin S}} \frac{t!}{k_1! \cdots k_n! (t - \sum_i k_i)!}.$$

Then by construction

$$\sum_{\underline{i} \in [n+1]^t} \widehat{\phi}(S_{\underline{i}})^2 = \sum_{S \in \binom{[n]}{\leq t}} |\mathcal{I}_S| \widehat{\phi}(S_{\underline{i}})^2$$

and therefore (10.26) is equivalent to the following pair of primal/dual second-order cone programs:

$$
\begin{array}{lll@{\qquad}ll}
\min & 2\varepsilon & & \max & -w + \sum_{x\in D} f(x)\phi(x) - \sum_{x\notin D} |\phi(x)| \\
\text{s.t.} & c = (c_S)_{S\in\binom{[n]}{\leq t}} \in \mathbb{R}^{\binom{[n]}{\leq t}}, \varepsilon \in \mathbb{R} & & \text{s.t.} & \phi = (\phi(x))_{x\in\{\pm1\}^n} \in \mathbb{R}^{\{\pm1\}^n}, w \in \mathbb{R} \\
& |p(x) - f(x)| \leq 2\varepsilon \text{ for } x \in D & & & \sum_{x\in\{\pm1\}^n} |\phi(x)| = 1 \\
& |p(x)| \leq 1 + 2\varepsilon \text{ for } x \notin D & & & v = \left(2^n \sqrt{|\mathcal{I}_S|}\, \widehat{\phi}(S)\right)_{S\in\binom{[n]}{\leq t}} \\
& \sum_{S\in\binom{[n]}{\leq t}} \frac{c_S^2}{|I_S|} \leq 1 & & & w \geq \|v\|_2 \\
& p = \sum_{S\in\binom{[n]}{\leq t}} c_S \chi_S & & &
\end{array}
\tag{10.27}
$$

We note that strong duality holds since both the primal and dual are strictly feasible.

Lemma 10.10. *If the optimal value of* (10.27) *is strictly larger than* 2ε*, then*

$$\text{cb-deg}_\varepsilon(f) > t.$$

Hence, the above forms a strengthening of the polynomial method. Indeed, any ϕ that is feasible for the maximization program in (10.25) (with objective $> 2\varepsilon$) will have low-degree Fourier coefficients equal to zero and therefore ($\phi, w = 0, v = 0$) will be feasible for the maximization program in (10.27) (with objective $> 2\varepsilon$). Also, notice that compared to (10.25) the primal here has the additional constraint that the coefficients of the approximating polynomial have to be normalized (w.r.t. a weighted 2-norm).

Chapter 11

Quantum algorithms for semidefinite programming

This chapter is based on the paper "Quantum SDP-solvers: Better upper and lower bounds", by J. van Apeldoorn, A. Gilyén, S. Gribling, R. de Wolf [vAGGdW17]. Some of the key ideas needed to provide the better upper bounds the title suggests have been generalized by Gilyén et al. [GSLW18], we have seen some of these generalizations in Section 9.3. Here we use those generalizations to provide a cleaner presentation of the results of [vAGGdW17].

After seeing many applications of semidefinite programming in the preceding chapters, we turn our attention to solving semidefinite programs using quantum computers. The first contribution in this direction was due to Brandão and Svore in 2016 [BS17]. They provided a quantum algorithm for solving semidefinite programs, which in some regimes is faster than the best-possible classical algorithms in terms of the dimension n of the problem and the number m of constraints, but worse in terms of various other parameters. This chapter is based on [vAGGdW17], the first work to improve on the results of Brandão and Svore, where we improve their algorithm in several ways, getting better dependence on those other parameters. Subsequent progress in the same framework has been made in [BKL$^+$17, vAG18a], which we briefly discuss in Section 11.5.

To be more concrete, let us recall the formulation of a pair of primal-dual semidefinite programs, and define some useful parameters. Given a set of matrices $C, A_1, \ldots, A_m \in \mathrm{S}^n$ and a vector $b \in \mathbb{R}^m$ we can define a pair of semidefinite programs, a *primal* (P) and a *dual* (D):[1]

$$\begin{array}{llll} (P) & \max & \langle C, X\rangle & \\ & \text{s.t.} & X \in \mathrm{S}^n_+ & \\ & & \mathrm{Tr}(A_j X) \leq b_j \quad j \in [m] & \end{array} \qquad \begin{array}{lll} (D) & \min & \langle b, y\rangle \\ & \text{s.t.} & y \in \mathbb{R}^m_+ \\ & & \sum_{j=1}^{m} y_j A_j - C \in \mathrm{S}^n_+ \end{array} \tag{11.1}$$

For the sake of normalization, let us assume that the operator norm of each of the matrices $C, A_1, \ldots, A_m$ is at most one. A special class of semidefinite programs is formed by *linear programs*, those SDPs for which all matrices involved are diagonal. Under assumptions that will be satisfied everywhere in this chapter, strong duality applies: the primal and dual SDP (11.1) will have the same optimal value OPT. To talk about the complexity of SDP-solvers, let us define some parameters. Let s be the sparsity of the input matrices: the maximal number of non-zero entries per row (and hence also per column) of the input matrices. Let R be an upper bound on the trace of an optimal X. Let r be an upper bound on $\|y\|_1$ for an optimal y to the dual. Let $\varepsilon > 0$ be the desired additive error with which we want to approximate OPT. Assume that the rows and columns of the matrices of SDP (11.1) can be accessed as adjacency lists: we can query, say, the ℓth non-zero entry of the kth row of matrix A_j in constant time (this is the same sparse access model that we have seen in Section 9.3.2). One can define 'solving' an SDP in different ways. At the very least an SDP-solver should produce an additive approximation of the value OPT. On top of that, one can require a solver to output a primal or dual point that is feasible, or nearly feasible, with the stated objective value. The algorithms stated below provide at least an approximation of OPT, but they differ in the additional output. Our algorithm (see Theorem 11.1) will provide, with high probability, a feasible solution y to the dual that is optimal up to an additive error ε.

One way to divide SDP-solvers into two categories is by looking at the dependence of their runtime on R, r, and $1/\varepsilon$.

The first class of SDP-solvers has a runtime that scales polylogarithmically in these parameters. This class of SDP-solvers encompasses for instance the ellipsoid method [GLS81], which is mainly of theoretical importance, and interior point methods [NN94], which are used in practice to solve SDPs. One can show that interior point methods can solve SDPs in time

$$\mathcal{O}(\sqrt{n}m(m^2 + mn^2 + n^3)L),$$

where L is a measure of the size of the instance [BTN01, Sec. 6.6]. The dependence on m and n becomes prohibitive even for moderate size SDPs.

The second class of SDP-solvers, often based on first-order methods, often provides a better runtime in terms of m and n, at the expense of a polynomial dependence on R, r, and $1/\varepsilon$. In this chapter we focus on the matrix version of the

[1] Note that we slightly deviate from the presentation in (1.1) by allowing inequality constraints in the primal instead of equality constraints.

multiplicative weights update method due to Arora and Kale [AK16].[2] A typical classical runtime for SDP-solvers in this framework is of the form

$$\mathcal{O}\left(nms \cdot \mathrm{poly}\left(\frac{Rr}{\varepsilon}\right)\right),$$

which can provide a faster algorithm for SDPs with small values of Rr/ε. The framework of Arora and Kale should really be seen as a meta-algorithm, because it does not specify how to implement a certain crucial step, let us call this 'the oracle' for now.[3] They themselves provide oracles that are optimized for special cases. For example for the MAXCUT SDP, they obtain a solver with near-linear runtime $\widetilde{\mathcal{O}}\big(|E|/\varepsilon^5\big)$ in the number of edges of the graph. For the sake of comparison, let us note that in [vAGGdW17] we show that one can get a general classical SDP-solver in their framework with complexity[4]

$$\widetilde{\mathcal{O}}\left(nms\left(\frac{Rr}{\varepsilon}\right)^4 + ns\left(\frac{Rr}{\varepsilon}\right)^7\right).$$

The first quantum SDP-solver of Brandão and Svore achieved a runtime of

$$\widetilde{\mathcal{O}}\left(\sqrt{mn}s^2\mathrm{poly}\left(\frac{Rr}{\varepsilon}\right)\right),$$

where the degree of the polynomial term is at least 32. Note that compared to the classical runtime this provides a quadratic improvement in terms of the dependence on m and n. We subsequently modified their algorithm. These modifications both simplify and speed up the quantum SDP-solver, resulting in complexity

$$\widetilde{\mathcal{O}}\left(\sqrt{mn}s^2\left(\frac{Rr}{\varepsilon}\right)^8\right).$$

The dependence on m, n, and s is the same as in Brandão-Svore, but our dependence on R, r, and $1/\varepsilon$ is substantially better. Note that each of the three parameters R, r, and $1/\varepsilon$ now occurs with the same 8th power in the complexity. This is no coincidence: as we show in [vAGGdW17, App. E], these three parameters can all be traded for one another, in the sense that we can massage the SDP to make each one of them small at the expense of making the others proportionally bigger. These trade-offs suggest we should actually think of Rr/ε as *one* parameter of the

[2] See also [AHK12] for a subsequent survey; the same algorithm was independently discovered around the same time in the context of learning theory [TRW05, WK12]. In the optimization community first-order methods for semidefinite programming have been considered for instance in [Ren16, Ren19].

[3] We provide a complete overview of the Arora-Kale method in Section 11.2.1. We refer to that section for definitions and details. 'The oracle' should not be confused with the oracle access to the input data.

[4] Here, and in the rest of this chapter, the notation $\widetilde{\mathcal{O}}(\cdot)$ is used to hide polylogarithmic factors in n, m, s, r, R and the desired additive error ε.

primal-dual pair of SDPs, not three separate parameters. For the special case of LPs, we can improve the runtime to

$$\widetilde{\mathcal{O}}\left(\sqrt{mn}\left(\frac{Rr}{\varepsilon}\right)^5\right).$$

Finally, in terms of upper bounds on the complexity of SDP solving, we mention that the current state of the art is due to van Apeldoorn and Gilyén [vAG18a] who provide an algorithm with a runtime of $\widetilde{\mathcal{O}}\Big((\sqrt{m}+\sqrt{n}\frac{Rr}{\varepsilon})s\big(\frac{Rr}{\varepsilon}\big)^4\Big)$. We briefly discuss their result in Section 11.5.

Limitations of our approach. Given that the runtime of our algorithm depends polynomially on the factor Rr/ε, a natural question is how big this term can be, or needs to be. In other words, for a fixed SDP (i.e., fixed R and r), what is the error up to which we can efficiently solve the SDP? As we will argue, sometimes the 'natural' choice of error is inverse polynomial in n and m, which negates our 'speed=up'. Let us briefly sketch why this is the case. As we will see, the output of our algorithm is a vector $y \in \mathbb{R}^m_+$ such that $\sum_{j=1}^m y_j A_j - C \succeq 0$ and $|\langle b, y\rangle - \mathrm{OPT}| \leq \varepsilon$. The vector y will be very sparse, it will have $\mathcal{O}(T)$ non-zero entries where $T = \mathcal{O}\Big(\big(\frac{Rr}{\varepsilon}\big)^2 \ln(n)\Big)$ is the number of iterations of our algorithm. Such sparse vectors have some advantages, for example they take much less space to store than arbitrary $y \in \mathbb{R}^m$. In fact, to get a sublinear running time in terms of m, this is necessary. However, this sparsity of the algorithm's output also points to a weakness of these methods: if *every* ε-optimal dual-feasible vector y has many non-zero entries, then the number of iterations needs to be large. For example, if every ε-optimal dual-feasible vector y has $\Omega(m)$ non-zero entries, then these methods require $T = \Omega(m)$ iterations before they can reach an ε-optimal dual-feasible vector. Since $T = \mathcal{O}\Big(\big(\frac{Rr}{\varepsilon}\big)^2 \ln(n)\Big)$ this would imply that $\frac{Rr}{\varepsilon} = \Omega(\sqrt{m/\ln(n)})$, and hence many classical SDP-solvers would have a better complexity than our quantum SDP-solver. As we show in Section 11.3, this will naturally be the case for families of SDPs that have a lot of symmetry.

Lower bounds. What about lower bounds for quantum SDP-solvers? Brandão and Svore already proved that a quantum SDP-solver has to make $\Omega(\sqrt{n}+\sqrt{m})$ queries to the input matrices, for some SDPs. Their lower bound is for a family of SDPs where $s, R, r, 1/\varepsilon$ are all constant, and is by reduction from a search problem. Somewhat surprisingly, the subsequent work in [BKL$^+$17, vAG18a] shows that this lower bound is in fact tight, in the setting where $s, R, r, 1/\varepsilon$ are all constant.

Here we step away from this regime. We prove lower bounds that are quantitatively stronger in m and n, but for SDPs with non-constant R and r. The key idea is to consider a Boolean function F on $N = abc$ input bits that is the composition of an a-bit majority function with a b-bit OR function that is composed with a c-bit majority function. The known quantum query complexities of majority and OR, combined with composition properties of the adversary lower bound, imply that every quantum algorithm that computes this function requires $\Omega(a\sqrt{b}c)$

queries. We define a family of LPs, with constant $1/\varepsilon$ but non-constant r and R, such that constant-error approximation of OPT computes F. Choosing a, b, and c appropriately, this implies a lower bound of

$$\Omega\Big(\sqrt{\max\{n,m\}}(\min\{n,m\})^{3/2}\Big)$$

queries to the entries of the input matrices for quantum LP-solvers. Since LPs are SDPs with sparsity $s = 1$, we get the same lower bound for quantum SDP-solvers. If m and n are of the same order, this lower bound is $\Omega(mn)$, the same scaling with mn as the classical general instantiation of Arora-Kale (11). In particular, this shows that we cannot have an $O(\sqrt{mn})$ upper bound without simultaneously having polynomial dependence on Rr/ε. The value of Rr/ε in the proof of our lower bound implies that for the case $m \approx n$, this polynomial dependence has to be at least $(Rr/\varepsilon)^{1/4}$.

Organization. This chapter is structured as follows. We first provide an informal overview of the Arora-Kale framework for solving SDPs in Section 11.1. This allows us to point out where the quantum improvements come from in Section 11.1.1. We then give a formal proof of our quantum SDP-solver in Section 11.2. We then proceed by highlighting the limitations of quantum SDP-solvers. First we consider SDP-solvers in the Arora-Kale framework (that are not tuned to specific classes of SDPs): we show that the inherent sparsity of the provided solutions puts a lower bound on the runtime for SDPs whose good solutions are dense (Section 11.3). We then prove some general lower bounds on the runtime of quantum LP-solvers and therefore quantum SDP-solvers (Section 11.4). Finally, in Section 11.5 we describe subsequent progress.

11.1 Basic approach

Arora and Kale [AK16] showed how to approximate OPT using a matrix version of the "multiplicative weights update" method. In Section 11.2.1 we will describe their framework in more detail, but in order to describe our result we will start with an overly simplified sketch here. The algorithm goes back and forth between candidate solutions to the primal SDP and to the corresponding dual SDP. Recall that under assumptions that will be satisfied everywhere in this chapter, strong duality applies: the primal and dual SDP (11.1) will have the same optimal value OPT. The algorithm does a binary search for OPT by trying different guesses α for it. Suppose we have fixed some α, and want to find out whether α is bigger or smaller than OPT. This is now a feasibility problem and we will try to construct a feasible solution to the dual with objective value at most α or show that it does not exist. Start with some candidate solution $X^{(1)}$ for the primal, for example a multiple of the identity matrix ($X^{(1)}$ has to be psd but need not be a feasible solution to the primal). This $X^{(1)}$ induces the following polytope:

$$\begin{aligned}\mathcal{P}_\varepsilon(X^{(1)}) := \{y \in \mathbb{R}^m : {} & b^T y \leq \alpha, \\ & \mathrm{Tr}\left(\Big(\sum_{j=1}^m y_j A_j - C\Big) X^{(1)}\right) \geq -\varepsilon, \\ & y \geq 0\}.\end{aligned} \tag{11.2}$$

This polytope can be thought of as a relaxation of the feasible region of the dual SDP with the extra constraint that OPT $\leq \alpha$: instead of requiring that $\sum_j y_j A_j - C$ is psd, we merely require that its inner product with the particular psd matrix $X^{(1)}$ is not too negative. The algorithm then calls an "oracle" that provides a $y^{(1)} \in \mathcal{P}_\varepsilon(X^{(1)})$, or outputs "fail" if $\mathcal{P}_0(X^{(1)})$ is empty (how to efficiently implement such an oracle depends on the application). In the "fail" case we know there is no dual-feasible y with objective value $\leq \alpha$, so we can increase our guess α for OPT, and restart. In case the oracle produced a $y^{(1)}$, this is used to define a Hermitian matrix $H^{(1)}$ and a new candidate solution $X^{(2)}$ for the primal, which is proportional to $e^{-H^{(1)}}$. Then the oracle for the polytope $\mathcal{P}_\varepsilon(X^{(2)})$ induced by this $X^{(2)}$ is called to produce a candidate $y^{(2)} \in \mathcal{P}_\varepsilon(X^{(2)})$ for the dual (or "fail"), this is used to define $H^{(2)}$ and $X^{(3)}$ proportional to $e^{-H^{(2)}}$, and so on.

Surprisingly, the average of the dual candidates $y^{(1)}, y^{(2)}, \ldots$ converges to a nearly-dual-feasible solution. Let w^* be the "width" of the oracle for a certain SDP: the maximum of $\left\|\sum_{j=1}^m y_j A_j - C\right\|$ over all psd matrices X and all vectors y that the oracle may output for the corresponding polytope $\mathcal{P}_\varepsilon(X)$. In general we will not know the width of an oracle exactly, but only an upper bound $w \geq w^*$, that may depend on the SDP; this is, however, enough for the Arora-Kale framework. In Section 11.2.1 we will show that without loss of generality we can assume the oracle returns a y such that $\|y\|_1 \leq r$ (recall that r is an upper bound on $\|y\|_1$ for an optimal y to the dual). Because we assumed $\|A_j\|, \|C\| \leq 1$, we have $w^* \leq r + 1$ as an easy width-bound. General properties of the multiplicative weights update method guarantee that after $T = \widetilde{\mathcal{O}}(w^2 R^2/\varepsilon^2)$ iterations, if no oracle call yielded "fail", then the vector $\frac{1}{T}\sum_{t=1}^T y^{(t)}$ is close to dual-feasible and satisfies $b^T y \leq \alpha$. This vector can then be turned into a dual-feasible solution by tweaking its first coordinate, certifying that OPT $\leq \alpha + \varepsilon$, and we can decrease our guess α for OPT accordingly.

The framework of Arora and Kale is really a meta-algorithm, because it does not specify how to implement the oracle. They themselves provide oracles that are optimized for special cases, which allows them to give a very low width-bound for these specific SDPs. As mentioned before, for example, for the MAXCUT SDP, they obtain a solver with near-linear runtime in the number of edges of the graph. They also observed that the algorithm can be made more efficient by not explicitly calculating the matrix $X^{(t)}$ in each iteration: the algorithm can still be made to work if instead of providing the oracle with $X^{(t)}$, we feed it good estimates of $\mathrm{Tr}(A_j X^{(t)})$ and $\mathrm{Tr}(C X^{(t)})$.

11.1.1 Quantum improvements

The Brandão-Svore quantum SDP-solver. The key idea of the Brandão-Svore algorithm is to take the Arora-Kale approach and to replace two of its steps by more efficient quantum subroutines. First, given a vector $y^{(t-1)}$, it turns out one can use "Gibbs sampling" to prepare the new primal candidate $X^{(t)} \propto e^{-H^{(t-1)}}$ *as a* $\log(n)$*-qubit quantum state* $\rho^{(t)} := X^{(t)} / \operatorname{Tr}(X^{(t)})$ in much less time than needed to compute $X^{(t)}$ as an $n \times n$ matrix. Second, one can efficiently implement the oracle for $\mathcal{P}_\varepsilon(X^{(t)})$ based on a number of copies of $\rho^{(t)}$, using those copies to estimate $\operatorname{Tr}(A_j \rho^{(t)})$ and $\operatorname{Tr}(A_j X^{(t)})$ when needed (note that $\operatorname{Tr}(A\rho)$ is the expectation value of operator A for the quantum state ρ). This is based on something called "Jaynes's principle". The resulting oracle is weaker than what is used classically, in the sense that it outputs a sample $j \sim y_j / \|y\|_1$ rather than the whole vector y. However, such sampling still suffices to make the algorithm work (it also means we can assume the vector $y^{(t)}$ to be quite sparse).

Our SDP-solver. Following Brandão and Svore, we make a quantum algorithm out of the Arora-Kale framework by giving a quantum implementation of the oracle. Our first observation is that the polytope $\mathcal{P}_\varepsilon(X)$ is extremely simple: it has only two constraints and therefore, if it is non-empty, then all its vertices have at most 2 non-zero coordinates.

A first naive approach would then be to find all vertices by solving $\Theta(m^2)$ linear systems of size 2×2 (this also determines if $\mathcal{P}_\varepsilon(X)$ is non-empty). Here each linear system is determined by values of the form $\operatorname{Tr}(A_j X)$, b_j and $\operatorname{Tr}(CX)$, and thus we can decide if $\mathcal{P}_\varepsilon(X)$ is non-empty with $\Theta(m^2)$ queries to such values.

We use a more sophisticated approach to show that $\Theta(m)$ classical queries suffice. Our approach is amenable to a quantum speed-up: we show that only $\Theta(\sqrt{m})$ quantum queries are needed. In particular, we show how to reduce the problem of finding a $y \in \mathcal{P}_\varepsilon(X)$ with $\|y\|_1 \leq r$ to finding a convex combination of points $(\operatorname{Tr}(A_j X), b_j)$ $(j \in [m])$ that lies within a certain region of the plane. The geometry of that region implies that if such a convex combination exists, then there exists such a convex combination of only two points $(\operatorname{Tr}(A_j X), b_j)$.[5] We show how to find such a convex combination using the generalized minimum-finding procedure presented in Theorem 9.4. This procedure uses $\widetilde{\mathcal{O}}(\sqrt{m})$ calls to an oracle that provides $\operatorname{Tr}(A_j X)$ and b_j. We then proceed to show that a quantum algorithm can compute $\operatorname{Tr}(A_j X)$ more efficiently than a classical computer.

[5] This in turn implies that the output of our oracle will be a 2-sparse vector in each iteration. Independently of us, Ben-David, Eldar, Garg, Kothari, Natarajan, and Wright (at MIT), and separately Ambainis observed that in the special case where all b_j are at least 1, the oracle can even be made 1-sparse, and the one entry can be found using one Grover search over m points (in both cases personal communication 2017). The same happens implicitly in our oracle in this case. However, in general 2 non-zero entries are necessary in y.

11.2 An improved quantum SDP-solver

Here we describe our quantum SDP-solver in more detail. In Section 11.2.1 we describe the framework designed by Arora and Kale for solving semidefinite programs. As in the recent work by Brandão and Svore, we use this framework to design an efficient quantum algorithm for solving SDPs. In particular, we show that the key subroutine needed in the Arora-Kale framework can be implemented efficiently on a quantum computer. Our implementation uses different techniques than the quantum algorithm of Brandão and Svore, allowing us to obtain a faster algorithm. The techniques required for this subroutine are developed in Sections 11.2.2 and 11.2.3. In Section 11.2.4 we put everything together to prove the main theorem of this section (the notation is explained below):

Theorem 11.1. *Instantiating Meta-Algorithm 1 using the trace calculation algorithm from Section 11.2.2 and the oracle from Section 11.2.3 (with width-bound* $w := r + 1$*), and using this to do a binary search for* $\mathrm{OPT} \in [-R, R]$ *(using different guesses* α *for* OPT*), gives a quantum algorithm for solving SDPs of the form* (11.1)*, which (with high probability) produces a feasible solution* y *to the dual program which is optimal up to an additive error* ε*, and uses*

$$\widetilde{\mathcal{O}}\left(\sqrt{nm}s^2\left(\frac{Rr}{\varepsilon}\right)^8\right)$$

queries to the input matrices and the same order of other gates.

Notation/Assumptions. We use log to denote the logarithm in base 2. We denote the all-zero matrix and vector by 0. Throughout we assume each element of the input matrices can be represented by a bitstring of size $\mathrm{poly}(\log n, \log m)$ (in particular this means that the input contains only rational entries!). We use s to denote the sparsity of the input matrices, that is, the maximum number of non-zero entries in a row (or column) of any of the matrices $C, A_1, \ldots, A_m$ is s. Recall that for normalization purposes we assume $\|A_1\|, \ldots, \|A_m\|, \|C\| \leq 1$. We assume throughout that the optimal value of both the primal and the dual is attained and that their values are equal. We furthermore assume that $A_1 = I$ and $b_1 = R$, that is, the trace of primal-feasible solutions is bounded by R (and hence also the trace of primal-optimal solutions is bounded by R). The analogous quantity for the dual SDP, an upper bound on $\sum_{j=1}^m y_j$ for an optimal dual solution y, will be denoted by r. However, we do not add the constraint $\sum_{j=1}^m y_j \leq r$ to the dual. We will assume $r \geq 1$. In Section 11.3 it will be necessary to work with the best possible upper bounds: we let R^* be the smallest trace of an optimal solution to the primal SDP (11.1), and we let r^* be the smallest ℓ_1-norm of an optimal solution to the dual.

Unless specified otherwise, we always consider *additive* error. In particular, an ε-optimal solution to an SDP will be a feasible solution whose objective value is within additive error ε of the optimum.

Input oracles. We assume sparse black-box access to the elements of the matrices $C, A_1, \dots, A_m$ defined in the following way: for input $(j,k,\ell) \in (\{0\} \cup [m]) \times [n] \times [s]$ we can query the location and value of the ℓth non-zero entry in the kth row of the matrix A_j (where $j = 0$ would indicate the C matrix).

Specifically in the quantum case, similar to (9.2) and (9.3), we assume access to an oracle O_I that calculates the $\mathrm{index}_{A_j} : [n] \times [s] \to [n]$ function, which for input (k,ℓ) gives the column index of the ℓth non-zero element in the kth row of A_j. We assume this oracle computes the index "in place":

$$O_I|j,k,\ell\rangle = |j,k,\mathrm{index}_{A_j}(k,\ell)\rangle. \tag{11.3}$$

(In the degenerate case where the kth row has fewer than ℓ non-zero entries, $\mathrm{index}_{A_j}(k,\ell)$ is defined to be ℓ together with some special symbol.) We also assume we can apply the inverse of O_I.

We also need another oracle O_M, returning a bitstring representation of $(A_j)_{ki}$ for any $j \in \{0\} \cup [m]$ and $k, i \in [n]$:

$$O_M|j,k,i,z\rangle = |j,k,i,z \oplus (A_j)_{ki}\rangle. \tag{11.4}$$

Computational model. As our computational model, we assume a slight relaxation of the usual quantum circuit model: a classical control system that can run quantum subroutines. We limit the classical control system so that its number of operations is at most a polylogarithmic factor bigger than the gate complexity of the quantum subroutines, i.e., if the quantum subroutines use C gates, then the classical control system may not use more than $\mathcal{O}(C \operatorname{polylog}(C))$ elementary operations.

When we talk about gate complexity, we count the number of 2-qubit quantum gates needed for implementation of the quantum subroutines. Additionally, we assume that there exists a unit-cost QRAM gate that allows us to store and retrieve qubits in a memory, by means of a swap of two registers indexed by another register:

$$\mathrm{QRAM} : |i, x, r_1, \dots, r_K\rangle \mapsto |i, r_i, r_1, \dots, r_{i-1}, x, r_{i+1}, \dots, r_K\rangle,$$

where the registers $r_1, \dots, r_K$ are only accessible through this gate. The QRAM gate can be seen as a quantum analogue of pointers in classical computing. The only place where we need QRAM is for a data structure that allows efficient access to the non-zero entries of a sum of sparse matrices [vAGGdW17, App. D]; for the special case of LP-solving it is not needed.

11.2.1 The Arora-Kale framework for solving SDPs

In this section we give a short introduction to the Arora-Kale framework for solving semidefinite programs. We refer to [AK16, AHK12] for a more detailed description and omitted proofs.

The key building block is the Matrix Multiplicative Weights (MMW) algorithm introduced by Arora and Kale in [AK16]. The MMW algorithm can be seen as a strategy for you in a game between you and an adversary. We first introduce the game. There is a number of rounds T. In each round you present a density matrix ρ to an adversary, the adversary replies with a loss matrix M satisfying $-I \preceq M \preceq I$. After each round you have to pay $\mathrm{Tr}(M\rho)$. Your objective is to pay as little as possible. The MMW algorithm is a strategy for you that allows you to lose not too much, in a sense that is made precise below. In Algorithm 1 we state the MMW algorithm, the following theorem shows the key property of the output of the algorithm.

Input Parameter $\eta \leq 1$, number of rounds T.

Rules In each round player 1 (you) presents a density matrix ρ, player 2 (the adversary) replies with a matrix M satisfying $-I \preceq M \preceq I$.

Output A sequence of symmetric $n \times n$ matrices $M^{(1)}, \ldots, M^{(T)}$ satisfying $-I \preceq M^{(t)} \preceq I$, for $t \in [T]$, and a sequence of $n \times n$ psd matrices $\rho^{(1)}, \ldots, \rho^{(T)}$ satisfying $\mathrm{Tr}\big(\rho^{(t)}\big) = 1$ for $t \in [T]$.

Strategy of player 1:

Take $\rho^{(1)} := I/n$
In round t:

1. Show the density matrix $\rho^{(t)}$ to the adversary.
2. Obtain the loss matrix $M^{(t)}$ from the adversary.
3. Update the density matrix as follows:

$$\rho^{(t+1)} := \exp\left(-\eta \sum_{\tau=1}^{t} M^{(\tau)}\right) \Bigg/ \mathrm{Tr}\left(\exp\left(-\eta \sum_{\tau=1}^{t} M^{(\tau)}\right)\right)$$

Algorithm 1: Matrix Multiplicative Weights (MMW) Algorithm

Theorem 11.2 ([AK16, Thm. 3.1]). *For every adversary, the sequence of density matrices $\rho^{(1)}, \ldots, \rho^{(T)}$ constructed using the Matrix Multiplicative Weights Algorithm 1 satisfies*

$$\sum_{t=1}^{T} \mathrm{Tr}\Big(M^{(t)}\rho^{(t)}\Big) \leq \lambda_{\min}\left(\sum_{t=1}^{T} M^{(t)}\right) + \eta \sum_{t=1}^{T} \mathrm{Tr}\Big((M^{(t)})^2\rho^{(t)}\Big) + \frac{\ln(n)}{\eta}.$$

Arora and Kale use the MMW algorithm to construct an SDP-solver. For that, they construct an adversary who promises to satisfy an additional condition: in each round t, the adversary returns a matrix $M^{(t)}$ whose trace inner product with the density matrix $\rho^{(t)}$ is non-negative. The above theorem shows that then, after T rounds, the average of the adversary's responses satisfies the stronger condition that its smallest eigenvalue is not too negative: $\lambda_{\min}\Big(\frac{1}{T}\sum_{t=1}^{T} M^{(t)}\Big) \geq -\eta - \frac{\ln(n)}{\eta T}$. More explicitly, the MMW algorithm is used to build a vector $y \geq 0$ such that

$\frac{1}{T}\sum_{t=1}^{T} M^{(t)}$ is proportional to $\sum_{j=1}^{m} y_j A_j - C$:

$$\frac{1}{T}\sum_{t=1}^{T} M^{(t)} \propto \sum_{j=1}^{m} y_j A_j - C,$$

and $b^T y \leq \alpha$. It then follows that the smallest eigenvalue of the matrix $\sum_{j=1}^{m} y_j A_j - C$ is only slightly below zero and the objective value $b^T y$ is at most α. Since $A_1 = I$, increasing the first coordinate of y makes the smallest eigenvalue of $\sum_j y_j A_j - C$ bigger. By the above we know how much the minimum eigenvalue has to be shifted to make the matrix positive semidefinite: $-\eta - \frac{\ln(n)}{\eta T}$. So $\overline{y} = y + \big(-\eta - \frac{\ln(n)}{\eta T}\big)e_1$ is dual feasible. With the right choice of parameters it can be shown that $\overline{y}$ satisfies $b^T\overline{y} \leq \alpha + \varepsilon$. In order to present the algorithm formally, we require some definitions.

Given a candidate solution $X \succeq 0$ for the primal problem (11.1) and a parameter $\varepsilon \geq 0$, define the polytope

$$\begin{aligned}\mathcal{P}_\varepsilon(X) := \{y \in \mathbb{R}^m :\ & b^T y \leq \alpha,\\ & \mathrm{Tr}\Big(\Big(\sum_{j=1}^{m} y_j A_j - C\Big)X\Big) \geq -\varepsilon,\\ & y \geq 0\}.\end{aligned}$$

One can verify the following:

Lemma 11.3 ([AK16, Lemma 4.2])**.** *If for a given candidate solution $X \succeq 0$ the polytope $\mathcal{P}_0(X)$ is empty, then a scaled version of X is primal-feasible and of objective value at least α.*

The Arora-Kale framework for solving SDPs uses the MMW algorithm where the role of the adversary is taken by an ε-approximate oracle whose role is to either provide a $y \in \mathcal{P}_\varepsilon(X)$, or certify that $\mathcal{P}_0 = \emptyset$, see Algorithm 2.

Input An $n \times n$ psd matrix X, a parameter $\alpha \in [-R, R]$, and the description of an SDP as in (11.1).

Output Either the $\mathsf{Oracle}_\varepsilon$ returns a vector y from the polytope $\mathcal{P}_\varepsilon(X)$ or it outputs "fail". It may only output fail if $\mathcal{P}_0(X) = \emptyset$.

Algorithm 2: Definition of an ε-approximate $\mathsf{Oracle}_\varepsilon$ for maximization SDPs

As we will see later, the runtime of the Arora-Kale framework depends on a property of the oracle called the *width*:

Definition 11.4 (*Width* of $\mathsf{Oracle}_\varepsilon$)**.** *The* width *of $\mathsf{Oracle}_\varepsilon$ for an SDP is the smallest $w^* \geq 0$ such that for every $X \succeq 0$ and $\alpha \in [-R, R]$, the vector y returned by $\mathsf{Oracle}_\varepsilon$ satisfies $\Big\|\sum_{j=1}^{m} y_j A_j - C\Big\| \leq w^*$.*

In practice, the width of an oracle is not always known. However, it suffices to work with an upper bound $w \geq w^*$: as we can see in Meta-Algorithm 1, the purpose of the width is to rescale the matrix $M^{(t)}$ in such a way that it forms a valid response for the adversary in the MMW algorithm. The following theorem shows the correctness of the Arora-Kale primal-dual meta-algorithm for solving SDPs, stated in Meta-Algorithm 1:

Input The input matrices and reals of SDP (11.1) and trace bound R. The current guess α of the optimal value. An additive error tolerance $\varepsilon > 0$. An $\frac{\varepsilon}{3}$-approximate oracle $\mathsf{Oracle}_{\varepsilon/3}$ as in Algorithm 2 with width-bound w.

Output Either "Lower" and a vector $\overline{y} \in \mathbb{R}^m_+$ feasible for (11.1) with $b^T\overline{y} \leq \alpha + \varepsilon$ or "Higher" and a symmetric $n \times n$ matrix X that, when scaled suitably, is primal-feasible with objective value at least α.

```
$T := \left\lceil \frac{9w^2R^2\ln(n)}{\varepsilon^2} \right\rceil$.
$\eta := \sqrt{\frac{\ln(n)}{T}}$.
$\rho^{(1)} := I/n$
for $t = 1, \ldots, T$ do
    Run $\mathsf{Oracle}_{\varepsilon/3}$ with $X^{(t)} = R\rho^{(t)}$.
    if $\mathsf{Oracle}_{\varepsilon/3}$ outputs "fail" then
        return "Higher" and a description of $X^{(t)}$.
    end if
    Let $y^{(t)}$ be the vector generated by $\mathsf{Oracle}_{\varepsilon/3}$.
    Set $M^{(t)} = \frac{1}{w}\left(\sum_{j=1}^m y_j^{(t)} A_j - C\right)$.
    Define $H^{(t)} = \sum_{\tau=1}^t M^{(\tau)}$.
    Update the state matrix as follows:
        $\rho^{(t+1)} := \exp\left(-\eta H^{(t)}\right) / \operatorname{Tr}\left(\exp\left(-\eta H^{(t)}\right)\right)$.
end for
If $\mathsf{Oracle}_{\varepsilon/3}$ does not output "fail" in any of the $T$ rounds, then output the dual solution $\overline{y} = \frac{\varepsilon}{R}e_1 + \frac{1}{T}\sum_{t=1}^T y^{(t)}$ where $e_1 = (1, 0, \ldots, 0) \in \mathbb{R}^m$.
```

Meta-Algorithm 1: Primal-Dual Algorithm for solving SDPs

Theorem 11.5 ([AK16, Theorem 4.7]). *Suppose we are given an SDP of the form* (11.1) *with input matrices $A_1 = I, A_2, \ldots, A_m$ and C having operator norm at most* 1, *and input reals $b_1 = R, b_2, \ldots, b_m$. Assume Meta-Algorithm 1 does not output "fail" in any of the rounds, then the returned vector $\overline{y}$ is feasible for the dual* (11.1) *with objective value at most $\alpha + \varepsilon$. If $\mathsf{Oracle}_{\varepsilon/3}$ outputs "fail" in the t-th round then a suitably scaled version of $X^{(t)}$ is primal-feasible with objective value at least α.*

The SDP-solver uses $T = \left\lceil \frac{9w^2R^2\ln(n)}{\varepsilon^2} \right\rceil$ iterations. In each iteration several steps have to be taken. The most expensive two steps are computing the matrix exponential of the matrix $-\eta H^{(t)}$ and the application of the oracle. Note that the only purpose of computing the matrix exponential is to allow the oracle to compute the values $\mathrm{Tr}(A_jX)$ for all j and $\mathrm{Tr}(CX)$, since the polytope depends on X only through those values. To obtain faster algorithms it is important to note, as was done already by Arora and Kale, that the primal-dual algorithm also works if we provide a (more accurate) oracle with *approximations* of $\mathrm{Tr}(A_jX)$. Let $a_j := \mathrm{Tr}(A_j\rho) = \mathrm{Tr}(A_jX)/\mathrm{Tr}(X)$ and $c := \mathrm{Tr}(C\rho) = \mathrm{Tr}(CX)/\mathrm{Tr}(X)$. Then, given a list of reals $\tilde{a}_1, \ldots, \tilde{a}_m, \tilde{c}$ and a parameter $\theta \geq 0$, such that $|\tilde{a}_j - a_j| \leq \theta$ for all j, and $|\tilde{c} - c| \leq \theta$, we define the polytope

$$\begin{aligned}\tilde{\mathcal{P}}(\tilde{a}_1, \ldots, \tilde{a}_m, \tilde{c} - (r+1)\theta) := \{y \in \mathbb{R}^m : {} & b^T y \leq \alpha, \\ & \sum_{j=1}^m y_j \leq r, \\ & \sum_{j=1}^m \tilde{a}_j y_j \geq \tilde{c} - (r+1)\theta \\ & y \geq 0\}.\end{aligned}$$

For convenience we will denote $\tilde{a} = (\tilde{a}_1, \ldots, \tilde{a}_m)$ and $c' := \tilde{c} - (r+1)\theta$. Notice that $\tilde{\mathcal{P}}$ also contains a new type of constraint: $\sum_j y_j \leq r$. Recall that r is defined as a positive real such that there exists an optimal solution y to SDP (11.1) with $\|y\|_1 \leq r$. Hence, using that $\mathcal{P}_0(X)$ is a *relaxation* of the feasible region of the dual (with bound α on the objective value), we may restrict our oracle to return only such y:

$$\mathcal{P}_0(X) \neq \emptyset \Rightarrow \mathcal{P}_0(X) \cap \{y \in \mathbb{R}^m : \sum_{j=1}^m y_j \leq r\} \neq \emptyset.$$

The benefit of this restriction is that an oracle that always returns a vector with bounded ℓ_1-norm automatically has a width $w^* \leq r+1$, due to the assumptions on the norms of the input matrices. The downside of this restriction is that the analogue of Lemma 11.3 does not hold for $\mathcal{P}_0(X) \cap \{y \in \mathbb{R}^m : \sum_j y_j \leq r\}$ (instead of $\mathcal{P}_0(X)$).

The following shows that an oracle that always returns a vector $y \in \tilde{\mathcal{P}}(\tilde{a}, c')$ if one exists, is a $4Rr\theta$-approximate oracle as defined in Algorithm 2.

Lemma 11.6. *Let $\tilde{a}_1, \ldots, \tilde{a}_m$ and $\tilde{c}$ be θ-approximations of $\mathrm{Tr}(A_1\rho), \ldots, \mathrm{Tr}(A_m\rho)$ and $\mathrm{Tr}(C\rho)$, respectively, where $X = R\rho$. Then the following holds:*

$$\mathcal{P}_0(X) \cap \{y \in \mathbb{R}^m : \sum_{j=1}^m y_j \leq r\} \subseteq \tilde{\mathcal{P}}(\tilde{a}, c') \subseteq \mathcal{P}_{4Rr\theta}(X).$$

Proof. First, suppose $y \in \mathcal{P}_0(X) \cap \{y \in \mathbb{R}^m : \sum_j y_j \leq r\}$. We then have $y \in \tilde{\mathcal{P}}(\tilde{a}, c')$ because

$$\sum_{j=1}^m \tilde{a}_j y_j - \tilde{c} \geq \sum_{j=1}^m (\tilde{a}_j - \mathrm{Tr}(A_j\rho)) y_j - (\tilde{c} - \mathrm{Tr}(C\rho)) \geq -\theta\|y\|_1 - \theta \geq -(r+1)\theta,$$

where we first subtracted $\sum_{j=1}^m \operatorname{Tr}(A_j\rho)y_j - \operatorname{Tr}(C\rho) \geq 0$, and then used the triangle inequality.

Next, suppose $y \in \tilde{\mathcal{P}}(\tilde{a}, c')$. We show that $y \in \mathcal{P}_{4Rr\theta}(X)$. Indeed, since $|\operatorname{Tr}(A_j\rho) - \tilde{a}_j| \leq \theta$ we have

$$\operatorname{Tr}\left(\left(\sum_{j=1}^m y_jA_j - C\right)\rho\right) \geq \left(\sum_{j=1}^m \tilde{a}_jy_j + \tilde{c}\right) - (r+1)\theta \geq -(2+r+\|y\|_1)\theta \geq -4r\theta$$

where the last inequality used our assumptions $r \geq 1$ and $\|y\|_1 \leq r$. Hence

$$\operatorname{Tr}\left(\left(\sum_{j=1}^m y_jA_j - C\right)X\right) \geq -4r\operatorname{Tr}(X)\theta = -4Rr\theta,$$

where for the equality we use $\operatorname{Tr}(X) = R$. □

We have now seen the Arora-Kale framework for solving SDPs. To obtain a quantum SDP-solver it remains to provide a quantum oracle subroutine. By the above discussion it suffices to set $\theta = \varepsilon/(12Rr)$, since with that choice of θ we have $\mathcal{P}_{4Rr\theta}(X) = \mathcal{P}_{\varepsilon/3}(X)$, and to use an oracle that is based on θ-approximations of $\operatorname{Tr}(A\rho)$ (for $A \in \{A_1, \ldots, A_m, C\}$). In Section 11.2.2 below we first give a quantum algorithm for approximating $\operatorname{Tr}(A\rho)$ efficiently. Then, in Section 11.2.3, we provide an oracle using those estimates. The oracle will be based on a simple geometric idea and can be implemented both on a quantum computer and on a classical computer (of course, resulting in different runtimes). In Section 11.2.4 we conclude with an overview of the runtime of our quantum SDP-solver. We want to stress that our solver is meant to work for any SDP. In particular, our oracle does not use the structure of a specific SDP. As we will show in Section 11.3, any oracle that works for all SDPs necessarily has a large width-bound. To obtain quantum speedups for a *specific* class of SDPs it will be necessary to develop oracles tuned to that problem, we view this as an important direction for future work.

11.2.2 Approximating $\operatorname{Tr}(A\rho)$ using a quantum algorithm

In this section we give an efficient quantum algorithm to approximate quantities of the form $\operatorname{Tr}(A\rho)$. We are going to work with Hermitian matrices $A, H \in \mathbb{C}^{n\times n}$, such that ρ is the Gibbs state $e^{-H}/\operatorname{Tr}(e^{-H})$. That is, we want to estimate

$$\operatorname{Tr}(A\rho) = \frac{\operatorname{Tr}(Ae^{-H})}{\operatorname{Tr}(e^{-H})}. \tag{11.5}$$

Note the analogy with quantum physics: in physics terminology $\operatorname{Tr}(A\rho)$ is simply called the “expectation value” of A for a quantum system in a thermal state corresponding to H.

The general approach is to separately estimate $\operatorname{Tr}(Ae^{-H})$ and $\operatorname{Tr}(e^{-H})$, and then to use the ratio of these estimates as an approximation of $\operatorname{Tr}(A\rho)$. Both estimations are done using state preparation to prepare a pure state with a ‘flag’ (a

1-qubit register), such that the probability that the flag is 0 is proportional to the quantity we want to estimate, and then to use amplitude estimation to estimate that probability (see Section 9.2.3). For example, to estimate $\mathrm{Tr}\big(Ae^{-H}\big)$ we could create a unitary U such that $U|0\rangle = |0\rangle|\psi\rangle + |1\rangle|\Phi\rangle$, where $|\psi\rangle$ is a subnormalized state such that $\||\psi\rangle\|^2 = \mathrm{Tr}\big(Ae^{-H}\big)$, and then use the amplitude estimation procedure (Lemma 9.3) to estimate $\||\psi\rangle\|^2 = \mathrm{Tr}\big(Ae^{-H}\big)$. To do so efficiently requires a lower bound on $\||\psi\rangle\|^2$. To give such a lower bound it suffices to have good control over the largest eigenvalue of e^{-H} and the smallest eigenvalue of A. We first show how to control the largest eigenvalue of e^{-H} and we then mention how we can assume a lower bound on the smallest eigenvalue of "A".

The largest eigenvalue of e^{-H} we equals $e^{-\lambda_{\min}(H)}$, we thus need to control the smallest eigenvalue of H. The first observation to make is that Equation (11.5) is invariant under shifting H by a multiple of the identity, that is, for the matrix $H_+ = H - \lambda_{\min}(H)I$ we have

$$\frac{\mathrm{Tr}\big(Ae^{-H}\big)}{\mathrm{Tr}(e^{-H})} = \frac{\mathrm{Tr}\big(Ae^{-H_+}\big)}{\mathrm{Tr}(e^{-H_+})},$$

since $e^{-H_+} = e^{\lambda_{\min}(H)}e^{-H}$. Hence, if we can efficiently find the smallest eigenvalue of H (approximately) then we can control the largest eigenvalue of e^{-H}.

We now show how to control the smallest eigenvalue of the second matrix in the trace inner product. It is clear that $\mathrm{Tr}(A\rho)$ is not invariant under shifting A, but it turns out that we can also get an additive approximation of $\mathrm{Tr}(A\rho)$ by combining multiplicative approximations of $\mathrm{Tr}\Big(\frac{I+A/4}{4}e^{-H}\Big)$ and $\mathrm{Tr}\big(e^{-H}\big)/4$. The important observation here is that both these traces are the inner product of a matrix whose largest eigenvalue we can control (e^{-H}) and a matrix whose smallest eigenvalue is at least $1/8$ (either $\frac{I+A/4}{4}$ or $I/4$).

The remainder of this section consists of two parts. We first provide a rigorous analysis of the above approach. We show that we can obtain an additive θ-approximation to $\mathrm{Tr}(A\rho) = \mathrm{Tr}\big(Ae^{-H}\big)/\,\mathrm{Tr}\big(e^{-H}\big)$ using $\widetilde{\mathcal{O}}\Big(\frac{\sqrt{n}dK}{\theta}\Big)$ queries to A and H, given that A is s-sparse and satisfies $\|A\| \leq 1$ and H is d-sparse and satisfies $\|H\| \leq K$ (and assuming $s \leq d$). The proof will use several of the techniques we have seen in Chapter 9 as black-boxes. We then unpack these boxes to a certain extent for the special case where all matrices are diagonal, showing an improved bound of $\widetilde{\mathcal{O}}\Big(\frac{\sqrt{n}}{\theta}\Big)$ queries, which is the relevant case for LP-solving.

General approach

To start, consider the following lemma about the multiplicative approximation error of a ratio of two real numbers that are given by multiplicative approximations:

Lemma 11.7. *Let $0 \leq \theta \leq 1$ and let $\alpha, \tilde{\alpha}, \beta, \tilde{\beta}$ be positive real numbers such that $|\alpha - \tilde{\alpha}| \leq \alpha\theta/3$ and $|\beta - \tilde{\beta}| \leq \beta\theta/3$. Then*

$$\left|\frac{\alpha}{\beta} - \frac{\tilde{\alpha}}{\tilde{\beta}}\right| \leq \theta\frac{\alpha}{\beta}$$

Proof. The inequality can be proven as follows

$$\begin{aligned}\left|\frac{\alpha}{\beta}-\frac{\tilde{\alpha}}{\tilde{\beta}}\right| &= \left|\frac{\alpha\tilde{\beta}-\tilde{\alpha}\beta}{\beta\tilde{\beta}}\right| = \left|\frac{\alpha\tilde{\beta}-\alpha\beta+\alpha\beta-\tilde{\alpha}\beta}{\beta\tilde{\beta}}\right| \\ &\leq \left|\frac{\alpha\tilde{\beta}-\alpha\beta}{\beta\tilde{\beta}}\right| + \left|\frac{\alpha\beta-\tilde{\alpha}\beta}{\beta\tilde{\beta}}\right| \leq \frac{\alpha\theta}{3\tilde{\beta}}+\frac{\alpha\theta}{3\tilde{\beta}} \leq \theta\frac{\alpha}{\beta}\end{aligned}$$

where the last step used $\tilde{\beta} \geq \frac{2}{3}\beta$. □

Corollary 11.8. *Let A be such that $\|A\| \leq 1$. A multiplicative $\theta/15$-approximation of both $\mathrm{Tr}\big((I+A/4)e^{-H}\big)/n$ and $\mathrm{Tr}\big(e^{-H}\big)/n$ can be turned into an additive θ-approximation of $\frac{\mathrm{Tr}\left(Ae^{-H}\right)}{\mathrm{Tr}(e^{-H})}$.*

Proof. According to Lemma 11.7, by dividing the two multiplicative approximations we get a multiplicative $\theta/5$-approximation of

$$\frac{\mathrm{Tr}\big((I+A/4)e^{-H}\big)}{\mathrm{Tr}(e^{-H})} = 1 + \frac{\mathrm{Tr}\big(\frac{A}{4}e^{-H}\big)}{\mathrm{Tr}(e^{-H})}. \tag{11.6}$$

Notice that $1+\frac{\mathrm{Tr}\left(\frac{A}{4}e^{-H}\right)}{\mathrm{Tr}(e^{-H})}$ can be upper bounded by $5/4$ (using that $\|A\| \leq 1$), which implies that the ratio is an additive $\theta/4$-approximation of (11.6). By subtracting 1 from this ratio and multiplying the result by 4 we thus obtain an additive θ-approximation to $\mathrm{Tr}\big(Ae^{-H}\big)/\,\mathrm{Tr}\big(e^{-H}\big)$. □

It thus suffices to approximate both quantities from the corollary separately. Notice that both are of the form $\mathrm{Tr}\big((I+A/4)e^{-H}\big)/n$, the first with the actual A, the second with $A=0$. In Lemma 11.9 we first show that we can estimate such a trace using a block-encoding of $B=\sqrt{I+A/4}e^{-H/2}$. Moreover, we can do so efficiently provided that $\mathrm{Tr}(B^*B)=\Omega(1)$.

Recall that a $(1,a,\varepsilon)$-block-encoding of a matrix $B \in \mathbb{C}^n$ is a $(n+2^a)\times(n+2^a)$ unitary of the form

$$\begin{pmatrix}\widetilde{B} & * \\ * & *\end{pmatrix}$$

where $\|B-\widetilde{B}\| \leq \varepsilon$, and the $*$'s represent matrices of an appropriate size, see Definition 9.6 for the precise definition.

Lemma 11.9. *Let U be a $(1,a,\varepsilon)$-block-encoding of a matrix $B \in \mathbb{C}^{n\times n}$ that satisfies $\mathrm{Tr}(B^*B)=\Omega(1)$. Let $0<\mu\leq 1$ and assume $\varepsilon \leq \mu\,\mathrm{Tr}(B^*B)/(4n)$. A multiplicative μ-approximation of $\mathrm{Tr}(B^*B)/n$ can be computed using $\widetilde{\mathcal{O}}\Big(\frac{\sqrt{n}}{\mu}\Big)$ applications of U and U^*, while using the same order of other gates.*

Proof. Let us define $\widetilde{B}=(\langle 0|^a\otimes I)U(|0\rangle^a\otimes I)$ such that $\|B-\widetilde{B}\| \leq \varepsilon$. Let $|\psi\rangle=\frac{1}{\sqrt{n}}\sum_{i\in[n]}|i\rangle|i\rangle$. Then, as we have seen before

$$\|(B\otimes I)|\psi\rangle\|^2 = \mathrm{Tr}(B^*B)/n.$$

Since $\|B - \widetilde{B}\| \leq \varepsilon$ we have $\Big| \|(B \otimes I)|\psi\rangle\| - \|(\widetilde{B} \otimes I)|\psi\rangle\| \Big| \leq \varepsilon$. Using this and the fact that $\|B\|, \|\widetilde{B}\| \leq 1$, we find that

$$\Big| \|(B \otimes I)|\psi\rangle\|^2 - \|(\widetilde{B} \otimes I)|\psi\rangle\|^2 \Big| \leq 2\varepsilon.$$

Let $\widetilde{U}$ be the circuit that first maps $|0\ldots0\rangle \mapsto \frac{1}{\sqrt{n}} \sum_{i\in[n]} |i\rangle|i\rangle$ and then applies $\widetilde{B}$ to the first register using U. That is, U acts on the all-zero state as

$$\widetilde{U}|0\ldots0\rangle = |0\rangle^a (\widetilde{B} \otimes I)|\psi\rangle + |\Phi\rangle,$$

where $(\langle 0|^a \otimes I)|\Phi\rangle = 0$. By the assumption on the eigenvalues of B and the value of ε, we have that $\|(\widetilde{B} \otimes I)|\psi\rangle\|^2 = \Omega(1/n)$. Therefore, applying amplitude estimation with the unitary $\widetilde{U}$ leads to a multiplicative $\mu/2$-approximation of $\|(\widetilde{B} \otimes I)|\psi\rangle\|^2$ using $\mathcal{O}(\frac{\sqrt{n}}{\mu})$ queries to $\widetilde{U}$ (and thus to U), see Lemma 9.3.

Finally we observe that if p is a multiplicative $\mu/2$-approximation of $\|(\widetilde{B} \otimes I)|\psi\rangle\|^2$, then

$$\begin{aligned} |p - \|(B \otimes I)|\psi\rangle\|^2| &\leq |p - \|(\widetilde{B} \otimes I)|\psi\rangle\|^2| + 2\varepsilon \\ &\leq \mu/2 \|(\widetilde{B} \otimes I)|\psi\rangle\|^2 + 2\varepsilon \\ &\leq \mu/2 (\|(B \otimes I)|\psi\rangle\|^2 + 2\varepsilon) + 2\varepsilon. \end{aligned}$$

Hence, since ε is such that $\mu/2 \cdot 2\varepsilon + 2\varepsilon \leq \mu/2 \|(B \otimes I)|\psi\rangle\|^2$, it follows that p is a multiplicative μ-approximation of $\|(B \otimes I)|\psi\rangle\|^2 = \mathrm{Tr}(B^*B)/n$. □

It remains to show how to implement a block-encoding of $\sqrt{I + A/4}e^{-H/2}$ with the desired properties efficiently. In order to have an $\Omega(1)$-lower bound on $\mathrm{Tr}\big((I + A/4)e^{-H}\big)$ it suffices to make sure that the smallest eigenvalues of $I + A/4$ is at least a constant, and that the largest eigenvalue of e^{-H} is at least a constant. By the assumption on the norm of A, the smallest eigenvalue of $I + A/4$ will be at least $3/4$. As we have remarked before, we may shift H by a scalar multiple of the identity. If we shift H such that $H \succeq 0$ but $H \not\succeq I$, then the largest eigenvalue of e^{-H} will be larger than $1/e$. Let us show how to efficiently implement a block-encoding of $\sqrt{I + A/4}e^{-H/2}$, assuming that $H \succeq 0$.

Lemma 11.10. *Let $A, H \in \mathbb{C}^{n\times n}$ ($n = 2^q$) be Hermitian matrices such that $\|A\| \leq 1$, $\|H\| \leq K$ for a known bound $K > 1$, and $H \succeq 0$. Assume A is s-sparse and H is d-sparse with $s \leq d$. Let $\varepsilon > 0$. We can implement a $(1, 2(q+6), 2\varepsilon)$-block-encoding of $\frac{\sqrt{I+A/4}}{4}\frac{e^{-H/2}}{4}$ using $\widetilde{\mathcal{O}}(dK)$ queries to A and H, while using the same order of other gates.*

Proof. Let U_A and U_H be $(1, q+6, \varepsilon)$-block-encodings of $\frac{\sqrt{I+A/4}}{4}$ and $e^{-H/2}/4$ respectively, constructed using Lemma 9.10 and Lemma 9.11. These block-encodings can be created using $\widetilde{\mathcal{O}}(dK)$ queries to A and H. Using Lemma 9.8 we can combine these block-encodings into a $(1, 2(q+6), 2\varepsilon)$-block-encoding $U_{A,H}$ of

$$B := \frac{\sqrt{I + A/4}}{4}\frac{e^{-H/2}}{4}.$$ □

Theorem 11.11. *Let $A, H \in \mathbb{C}^{n\times n}$ be Hermitian matrices such that $\|A\| \leq 1$ and $\|H\| \leq K$ for a known bound $K > 1$. Assume A is s-sparse and H is d-sparse with $s \leq d$. An additive θ-approximation of*

$$\mathrm{Tr}(A\rho) = \frac{\mathrm{Tr}\big(Ae^{-H}\big)}{\mathrm{Tr}(e^{-H})}$$

can be computed using $\widetilde{\mathcal{O}}\Big(\frac{\sqrt{n}dK}{\theta}\Big)$ queries to A and H, while using the same order of other gates.

Proof. Start by computing an estimate $\tilde{\lambda}_{\min}$ of $\lambda_{\min}(H)$, the minimum eigenvalue of H, up to additive error $1/2$ using Lemma 9.5. We define $H_+ := H-(\tilde{\lambda}_{\min}-1/2)I$, so that $\lambda_{\min}(H_+) \in [0, 1]$. We then apply Lemma 11.10 and Lemma 11.9 twice, once to the matrices A, H_+ and once to the matrices $0, H_+$, to obtain multiplicative $\theta/15$-approximations of $\mathrm{Tr}\big((I + A/4)e^{-H}\big)/n$ and $\mathrm{Tr}\big(e^{-H}\big)/n$, which we can combine to an additive θ-approximation of $\mathrm{Tr}(A\rho)$ using Corollary 11.8. The complexity statement follows from the applied lemmas. □

The special case of diagonal matrices – for LP-solving

In this section we consider diagonal matrices, assuming oracle access to H of the following form:

$$O_H|i\rangle|z\rangle = |i\rangle|z \oplus H_{ii}\rangle$$

and similarly for A. Notice that this kind of oracle can easily be constructed from the general sparse matrix oracle (11.4) that we assume access to.

Lemma 11.12. *Let $A, H \in \mathbb{R}^{n\times n}$ be diagonal matrices such that $\|A\| \leq 1$ and $H \succeq 0$, and let $\varepsilon > 0$ be an error parameter. Then there exists a unitary $\tilde{U}_{A,H}$ such that*

$$\left|\left\|(\langle 0| \otimes I)\tilde{U}_{A,H}|0\dots0\rangle\right\|^2 - \mathrm{Tr}\bigg(\frac{I + A/2}{4n}e^{-H}\bigg)\right| \leq \varepsilon,$$

which uses 1 quantum query to A and H and $\mathcal{O}(\log^{\mathcal{O}(1)}(1/\varepsilon) + \log(n))$ other gates.

Proof. First we prepare the state $\sum_{i=1}^n |i\rangle/\sqrt{n}$ with $\mathcal{O}(\log(n))$ one- and two-qubit gates. If n is a power of 2 we do this by applying $\log_2(n)$ Hadamard gates on $|0\rangle^{\otimes \log_2(n)}$; in the general case it is still possible to prepare the state $\sum_{i=1}^n |i\rangle/\sqrt{n}$ with $\mathcal{O}(\log(n))$ two-qubit gates, for example by preparing the state $\sum_{i=1}^k |i\rangle/\sqrt{k}$ for $k = 2^{\lceil \log_2(n)\rceil}$ and then using (exact) amplitude amplification in order to remove the $i > n$ from the superposition.

Then we query the values of H and A to get the state $\sum_{i=1}^n |i\rangle|H_{ii}\rangle|A_{ii}\rangle/\sqrt{n}$. Using these binary values we apply a finite-precision arithmetic circuit to prepare

$$\frac{1}{\sqrt{n}}\sum_{i=1}^{n} |i\rangle|H_{ii}\rangle|A_{ii}\rangle|\beta_i\rangle,$$

where $\beta_i := \arcsin\bigg(\sqrt{\frac{1+A_{ii}/2}{4}e^{-H_{ii}} + \delta_i}\bigg)/\pi$, and $|\delta_i| \leq \varepsilon$. Note that the error δ_i comes from writing down only a finite number of bits $b_1.b_2b_3\dots b_{\log(8/\varepsilon)}$ of

$\frac{1+A_{ii}/2}{4}e^{-H_{ii}}$. Due to our choice of A and H, we know that β_i lies in $[0,1]$. We proceed by first adding an ancilla qubit initialized to $|1\rangle$ in front of the state, then we apply $\log(8/\varepsilon)$ controlled rotations to this qubit: for each j such that $b_j = 1$ we apply a rotation by angle $\pi 2^{-j}$. In other words, if $b_1 = 1$, then we rotate $|1\rangle$ fully to $|0\rangle$. If $b_2 = 1$, then we rotate halfway, and we proceed further by halving the angle for each subsequent bit. We will end up with a normalized version of the state:

$$\sum_{i=1}^{n}\left(\sqrt{\frac{1+A_{ii}/2}{4}e^{-H_{ii}}+\delta_i}|0\rangle+\sqrt{1-\frac{1+A_{ii}/2}{4}e^{-H_{ii}}-\delta_i}|1\rangle\right)|i\rangle|A_{ii}\rangle|H_{ii}\rangle|\beta_i\rangle.$$

It is now easy to see that the squared norm of the $|0\rangle$-part of the normalized state is as required:

$$\left\|\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\sqrt{\frac{1+A_{ii}/2}{4}e^{-H_{ii}}+\delta_i}|i\rangle\right\|^2=\frac{\mathrm{Tr}\big((I+A/2)e^{-H}\big)}{4n}+\sum_{i=1}^{n}\frac{\delta_i}{n},$$

which is an additive ε-approximation since $\left|\sum_{i=1}^{n}\frac{\delta_i}{n}\right| \leq \varepsilon$. □

Corollary 11.13. *Let $A, H \in \mathbb{R}^{n\times n}$ be diagonal matrices, with $\|A\| \leq 1$, let $\theta \in (0,1]$. An additive θ-approximation of*

$$\mathrm{Tr}(A\rho)=\frac{\mathrm{Tr}\big(Ae^{-H}\big)}{\mathrm{Tr}(e^{-H})}$$

can be computed using $\mathcal{O}\Big(\frac{\sqrt{n}}{\theta}\Big)$ queries to A and H and $\widetilde{\mathcal{O}}\Big(\frac{\sqrt{n}}{\theta}\Big)$ other gates.

Proof. Let $\varepsilon > 0$ be a constant to be determined later. Since H is a diagonal matrix, its eigenvalues are exactly its diagonal entries. Using the quantum minimum-finding algorithm of Dürr and Høyer [DH96] (see also Section 9.2.4) one can find (with high success probability) the minimum $\lambda_{\min}$ of the diagonal entries using $\mathcal{O}(\sqrt{n})$ queries to the matrix elements. Let $H_+ := H - \lambda_{\min}I$. Lemma 11.12 applied to A and H_+ shows that there exists a unitary U such that $U|0\ldots 0\rangle = |0\rangle|\psi\rangle + |1\rangle|\phi\rangle$ where $|\psi\rangle$ and $|\phi\rangle$ are subnormalized states such that

$$\left|\||\psi\rangle\|^2-\frac{\mathrm{Tr}\big((I+A/2)e^{-H_+}\big)}{4n}\right|\leq\varepsilon.$$

Assuming that $\varepsilon < \frac{1}{2}\frac{\mathrm{Tr}\left((I+A/2)e^{-H_+}\right)}{4n}$, this implies that

$$\||\psi\rangle\|^2\geq\frac{1}{2}\frac{\mathrm{Tr}\big((I+A/2)e^{-H_+}\big)}{4n}=\Omega(1/n),$$

where we use that $\lambda_{\min}(H_+) \in [0,1]$. Therefore, using the amplitude estimation procedure of Lemma 9.3, we can find a $\theta/18$-multiplicative approximation p of $\||\psi\rangle\|^2$ using $\widetilde{\mathcal{O}}\Big(\frac{\sqrt{n}}{\theta}\Big)$ applications of U and U^*. As in the proof of Lemma 11.9 one can show that this $\theta/18$-multiplicative approximation of $\||\psi\rangle\|^2$ is a $\theta/9$-multiplicative approximation of $\frac{\mathrm{Tr}\left((I+A/2)e^{-H_+}\right)}{4n}$, assuming that $\varepsilon \leq \frac{\theta}{9} \cdot \frac{\mathrm{Tr}\left((I+A/2)e^{-H_+}\right)}{16n}$. We therefore choose $\varepsilon = \frac{\theta}{9} \cdot \frac{\mathrm{Tr}\left((I+A/2)e^{-H_+}\right)}{16n}$. Repeating the procedure with $A = 0$ and combining the resulting multiplicative approximations using Lemma 11.8 shows that an additive θ-approximation to $\mathrm{Tr}(A\rho) = \frac{\mathrm{Tr}\left(Ae^{-H}\right)}{\mathrm{Tr}(e^{-H})}$ can be obtained using $\mathcal{O}\left(\frac{\sqrt{n}}{\theta}\right)$ queries to A and H. □

11.2.3 An efficient 2-sparse oracle

Recall from the end of Section 11.2.1 that $\tilde{a}_j$ is an additive θ-approximation to $\mathrm{Tr}(A_j\rho)$, $\tilde{c}$ is a θ-approximation to $\mathrm{Tr}(C\rho)$ and $c' = \tilde{c} - r\theta - \theta$.

Our goal is to find a $y \in \tilde{\mathcal{P}}(\tilde{a}, c')$, i.e., a y such that

$$\begin{aligned} \|y\|_1 &\leq r \\ b^T y &\leq \alpha \\ \tilde{a}^T y &\geq c' \\ y &\geq 0 \end{aligned} \tag{11.7}$$

We first describe our quantum 2-sparse oracle assuming access to a unitary which acts as $|j\rangle|0\rangle|0\rangle \mapsto |j\rangle|\tilde{a}_j\rangle|\psi_j\rangle$, where $|\psi_j\rangle$ is some workspace state depending on j. We then briefly discuss how to modify the analysis when we are given an oracle which acts as $|j\rangle|0\rangle|0\rangle \mapsto |j\rangle \sum_i \beta_j^i |\tilde{a}_j^i\rangle|\psi_j^i\rangle$ (where each $\tilde{a}_j^i$ is an additive θ-approximation to $\mathrm{Tr}(A_j\rho)$), since this is the output of the trace-estimation procedure of the previous section.

If $\alpha \geq 0$ and $c' \leq 0$, then $y = 0$ is a solution to (11.7) and our oracle can return it. If not, then we may write $y = Nq$ with $N = \|y\|_1 > 0$ and hence $\|q\|_1 = 1$. So we are looking for an N and a q such that

$$\begin{aligned} b^T q &\leq \alpha/N \\ \tilde{a}^T q &\geq c'/N \\ \|q\|_1 &= 1 \\ q &\geq 0 \\ 0 < N &\leq r \end{aligned} \tag{11.8}$$

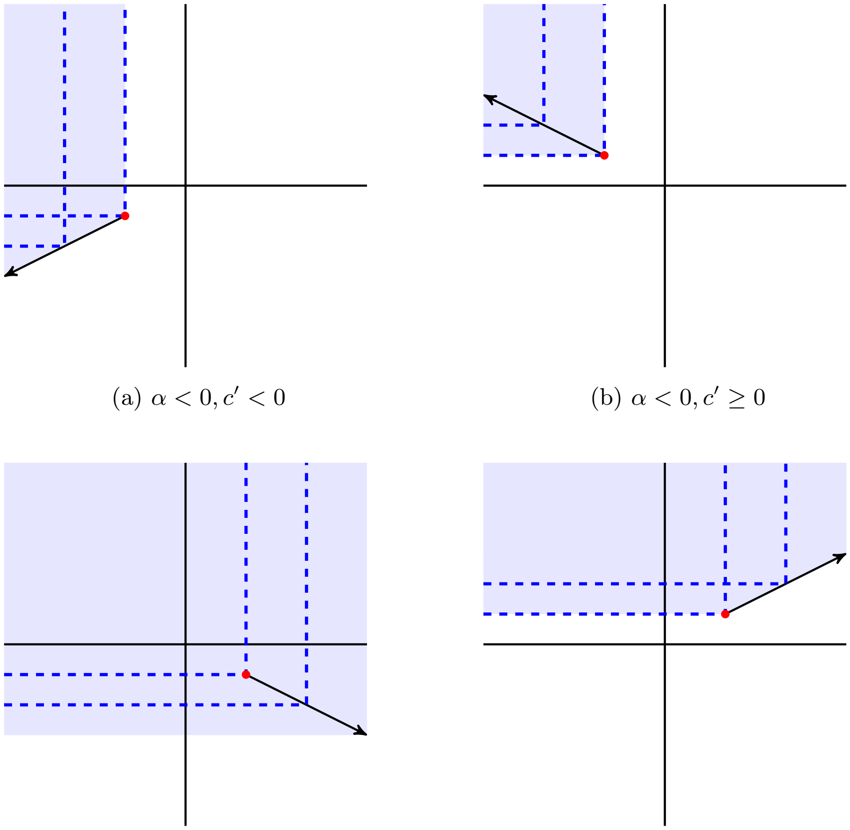
We can now view $q \in \mathbb{R}_+^m$ as the coefficients of a convex combination of the points $p_i = (b_i, \tilde{a}_i)$ in the plane. We want such a combination that lies to the upper left of $g_N = (\alpha/N, c'/N)$ for some $0 < N \leq r$. Let $\mathcal{G}_N$ denote the upper-left quadrant of the plane starting at g_N.

Lemma 11.14. *If there is a $y \in \tilde{\mathcal{P}}(\tilde{a}, c')$, then there is a 2-sparse $y' \in \tilde{\mathcal{P}}(\tilde{a}, c')$ such that $\|y\|_1 = \|y'\|_1$.*

Proof. Consider $p_i = (b_i, \tilde{a}_i)$ and $g = (\alpha/N, c'/N)$ as before, and write $y = Nq$ where $\sum_{j=1}^m q_j = 1$, $q \geq 0$. The vector q certifies that a convex combination of the points p_i lies in $\mathcal{G}_N$. But then there exist $j, k \in [m]$ such that the line segment $\overline{p_j p_k}$

intersects $\mathcal{G}_N$. All points on this line segment are convex combinations of p_j and p_k, hence there is a convex combination of p_j and p_k that lies in $\mathcal{G}_N$. This gives a 2-sparse q', and $y' = Nq' \in \tilde{\mathcal{P}}(\tilde{a}, c')$. □

Let $\mathcal{G} = \bigcup_{N\in(0,r]} \mathcal{G}_N$, see Figure 11.1 for the shape of $\mathcal{G}$. Then we want to find two points p_j, p_k that have a convex combination in $\mathcal{G}$, since this implies that a scaled version of their convex combination gives a $y \in \tilde{\mathcal{P}}(\tilde{a}, c')$ with $\|y\|_1 \leq r$ (this scaling can be computed efficiently given p_j and p_k).



(a) $\alpha < 0, c' < 0$

(b) $\alpha < 0, c' \geq 0$

(c) $\alpha \geq 0, c' < 0$

(d) $\alpha \geq 0, c' \geq 0$

Figure 11.1: The region $\mathcal{G}$ in light blue. The borders of two quadrants $\mathcal{G}_N$ have been drawn by thick dashed blue lines. The red dot at the beginning of the arrow is the point $(\alpha/r, c'/r)$.

Furthermore, regarding the possible (non-)emptiness of $\mathcal{G}$ we know the following by Lemma 11.6 and Lemma 11.14:

- If $\mathcal{P}_0(X) \cap \{y \in \mathbb{R}^m \colon \sum_j y_j \leq r\}$ is non-empty, then some convex combination of two of the p_j's lies in $\mathcal{G}$.

- If $\mathcal{P}_{4Rr\theta}(X) \cap \{y \in \mathbb{R}^m : \sum_j y_j \leq r\}$ is empty, then no convex combination of the p_j's lies in $\mathcal{G}$.

Lemma 11.15. *There is an algorithm that returns a 2-sparse vector q such that $\sum_{j=1}^m q_j p_j \in \mathcal{G}$, if one exists, using one search and two minimizations over the m points $p_j = (b_j, \tilde{a}_j)$. This gives a classical algorithm that uses $\mathcal{O}(m)$ calls to the subroutine that gives the entries of $\tilde{a}$, and $\mathcal{O}(m)$ other operations; and a quantum algorithm that (in order to solve the problem with high probability) uses $\mathcal{O}(\sqrt{m})$ calls to an (exact quantum) subroutine that gives the entries of $\tilde{a}$, and $\widetilde{\mathcal{O}}(\sqrt{m})$ other gates.*

Proof. The algorithm can be summarized as follows:

1. Check if $\alpha \geq 0$ and $c' \leq 0$. If so, then return $q = 0$.

2. Check if there is a $p_i \in \mathcal{G}$. If so, then return $q = e_i$

3. Find p_j, p_k so that the line segment $\overline{p_j p_k}$ goes through $\mathcal{G}$ and return the corresponding q.

4. If the first three steps did not return a vector q, then output 'Fail'.

The main realization is that in step 3 we can search separately for p_j and p_k. We explain this in more detail below, but first we will need a better understanding of the shape of $\mathcal{G}$ (see Figure 11.1 for illustration). The shape of $\mathcal{G}$ depends on the sign of α and c'.

(a) If $\alpha < 0$ and $c' < 0$. The corner point of $\mathcal{G}$ is $(\alpha/r, c'/r)$. One edge goes up vertically and an other follows the line segment $\lambda \cdot (\alpha, c')$ for $\lambda \in [1/r, \infty)$ starting at the corner.

(b) If $\alpha < 0$ and $c' \geq 0$. Here $\mathcal{G}_N \subseteq \mathcal{G}_r$ for $N \leq r$. So $\mathcal{G} = \mathcal{G}_r$. The corner point is again $(\alpha/r, c'/r)$, but now one edge goes up vertically and one goes to the left horizontally.

(c) If $\alpha \geq 0$ and $c' \leq 0$. This is the case where $y = 0$ is a solution, $\mathcal{G}$ is the whole plane and has no corner.

(d) If $\alpha \geq 0$ and $c' > 0$. The corner point of $\mathcal{G}$ is again $(\alpha/r, c'/r)$. From there one edge goes to the left horizontally and one edge follows the line segment $\lambda \cdot (\alpha, c')$ for $\lambda \in [1/r, \infty)$.

Since $\mathcal{G}$ is always an intersection of at most 2 halfspaces, steps 1-2 of the algorithm are easy to perform. In step 1 we handle case (c) by simply returning $y = 0$. For the other cases $(\alpha/r, c'/r)$ is the corner point of $\mathcal{G}$ and the two edges are simple lines. Hence in step 2 we can easily search through all the points to find out if there is one lying in $\mathcal{G}$; since $\mathcal{G}$ is a very simple region, this only amounts to checking on which side of the two lines a point lies.

Now, if we cannot find a single point in $\mathcal{G}$ in step 2, then we need a combination of two points in step 3. Let L_1, L_2 be the edges of $\mathcal{G}$ and let ℓ_j and ℓ_k be the

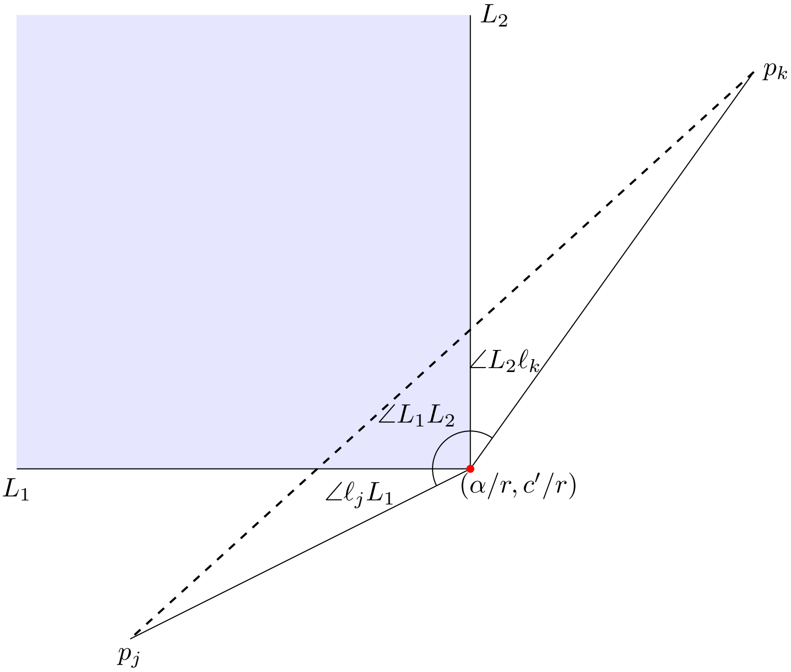


Figure 11.2: Illustration of $\mathcal{G}$ with the points p_j, p_k and the angles $\angle\ell_jL_1, \angle L_1L_2, \angle L_2\ell_k$ drawn in. Clearly the line $\overline{p_jp_k}$ only crosses $\mathcal{G}$ when the total angle is less than π.

line segments from $(\alpha/r, c'/r)$ to p_j and p_k, respectively. Then, as can be seen in Figure 11.2, the line segment $\overline{p_jp_k}$ goes through $\mathcal{G}$ if and only if (up to relabeling p_j and p_k) $\angle\ell_jL_1 + \angle L_1L_2 + \angle L_2\ell_k \leq \pi$. Since $\angle L_1L_2$ is fixed, we can simply look for a j such that $\angle\ell_jL_1$ is minimized and a k such that $\angle L_2\ell_k$ is minimized. If $\overline{p_jp_k}$ does not pass through $\mathcal{G}$ for this pair of points, then it does not for any of the pairs of points.

Notice that these minimizations can be done separately and hence can be done in the stated complexity. Given the minimizing points p_j and p_k, it is easy to check if they give a solution by calculating the angle between ℓ_j and ℓ_k. The coefficients of the convex combination q are then easy to compute. □

The analysis above applies if there are m points $p_j = (b_j, \tilde{a}_j)$, where $j \in [m]$, and we are given a unitary which acts as $|j\rangle|0\rangle|0\rangle \mapsto |j\rangle|\tilde{a}_j\rangle|\psi_j\rangle$. We now consider the more general case where we are given access to a unitary which for each j provides a superposition over different values $\tilde{a}_j$. That is, we assume that we are given an oracle that acts as $|j\rangle|0\rangle|0\rangle \mapsto |j\rangle\sum_i \beta_j^i|\tilde{a}_j^i\rangle|\psi_j^i\rangle$ where each $|\tilde{a}_j^i\rangle$ is an approximation of a_j and the amplitudes β_j^i are such that measuring the second register with high probability returns an $\tilde{a}_j^i$ which is θ-close to a_j. We do so because the trace estimation procedure of Corollary 11.11 provides an oracle of this form.

Since we can exponentially reduce the probability that we obtain an $\tilde{a}_j^i$ which is further than θ away from a_j, we will for simplicity assume that for all i, j we have $|\tilde{a}_j^i - a_j| \leq \theta$; the neglected exponentially small probabilities will only affect the analysis in negligible ways.

Let $p_j^i := (b_j, \tilde{a}_j^i)$. Our new goal will be to find a 2-sparse vector q and points p_j^i and $p_k^{i'}$ such that $q_j p_j^i + q_k p_k^{i'} \in \mathcal{G}$, or to conclude that for all $j, k \in [m]$ there exist i and i' such that no q exists for which $q_j p_j^i + q_k p_k^{i'} \in \mathcal{G}$.

Note that while we do not allow our quantum algorithm enough time to obtain classical descriptions of all $\tilde{a}_j$s (we aim for a runtime of $\widetilde{\mathcal{O}}(\sqrt{m})$), we do have enough time to compute $\tilde{c}$ once initially (after this measurement $\mathcal{G}$ is well-defined). Knowing $\tilde{c}$, we can compute the angles defined by the points $p_j^i = (b_j, \tilde{a}_j^i)$ with respect to the corner point of $(\alpha/r, (\tilde{c} - \theta)/r - \theta)$ and the lines L_1, L_2 (see Figure 11.2). We now apply our generalized minimum-finding algorithm with runtime $\widetilde{\mathcal{O}}(\sqrt{m})$ (see Theorem 9.4) starting with a uniform superposition over the js to find $k, \ell \in [m]$ and points p_k^i and $p_\ell^{i'}$ approximately minimizing the respective angles to lines L_1, L_2. Here 'approximately minimizing' means that there is no $j \in [m]$ such that for all i'' the angle of $p_j^{i''} = (b_j, \tilde{a}_j^{i''})$ with L_1 is smaller than that of p_k^i with L_1 (and similar for ℓ and L_2). From this point on we can simply consider the model in Lemma 11.15 since by the analysis above there exists an approximation $\tilde{a} \in \mathbb{R}^m$ with $\tilde{a}_k = \tilde{a}_k^i$ and $\tilde{a}_\ell = \tilde{a}_\ell^{i'}$ and where k and ℓ are the correct minimizers.

11.2.4 Total runtime

We are now ready to add our quantum implementations of the trace calculations and the oracle to the classical Arora-Kale framework.

Theorem 11.1. *Instantiating Meta-Algorithm 1 using the trace calculation algorithm from Section 11.2.2 and the oracle from Section 11.2.3 (with width-bound $w := r + 1$), and using this to do a binary search for* OPT $\in [-R, R]$ *(using different guesses α for* OPT*), gives a quantum algorithm for solving SDPs of the form* (11.1)*, which (with high probability) produces a feasible solution y to the dual program which is optimal up to an additive error ε, and uses*

$$\widetilde{\mathcal{O}}\left(\sqrt{nm}s^2\left(\frac{Rr}{\varepsilon}\right)^8\right)$$

queries to the input matrices and the same order of other gates.

Proof. Using our implementations of the different building blocks, it remains to calculate what the total complexity will be when they are used together.

Cost of the oracle for $H^{(t)}$. The first problem in each iteration is to obtain access to an oracle for $H^{(t)}$. In each iteration the oracle will produce a $y^{(t)}$ that is at most 2-sparse, and hence in the $(t+1)$th iteration, $H^{(t)}$ is a linear combination of $2t$ of the A_j matrices and the C matrix.

We can write down a sparse representation of the coefficients of the linear combination that gives $H^{(t)}$ in each iteration by adding the new terms coming from $y^{(t)}$. This will clearly not take longer than $\widetilde{\mathcal{O}}(T)$, since there are only a constant number of terms to add for our oracle. As we will see, this term will not dominate the complexity of the full algorithm.

Using such a sparse representation of the coefficients, one query to a sparse representation of $H^{(t)}$ will cost $\widetilde{\mathcal{O}}(st)$ queries to the input matrices and $\widetilde{\mathcal{O}}(st)$ other gates. For a detailed explanation and a matching lower bound for this part, see [vAGGdW17, App. D].

Cost of the oracle for $\mathrm{Tr}(A_j\rho)$**.** In each iteration $M^{(t)}$ is made to have operator norm at most 1. This means that

$$\left\|-\eta H^{(t)}\right\| \leq \eta \sum_{\tau=1}^{t} \left\|M^{(\tau)}\right\| \leq \eta t.$$

Furthermore we know that $H^{(t)}$ is at most $d := s(2t+1)$-sparse. Calculating $\mathrm{Tr}(A_j\rho)$ for one index j up to an additive error of $\theta := \varepsilon/(12Rr)$ can be done using the algorithm from Theorem 11.11. This will take

$$\widetilde{\mathcal{O}}\left(\sqrt{n}\frac{\|H\|d}{\theta}\right) = \widetilde{\mathcal{O}}\left(\sqrt{n}s\eta t^2\left(\frac{Rr}{\varepsilon}\right)\right)$$

queries to the oracle for $H^{(t)}$ and the same order of other gates. Since each query to $H^{(t)}$ takes $\widetilde{\mathcal{O}}(st)$ queries to the input matrices, this means that

$$\widetilde{\mathcal{O}}\left(\sqrt{n}\eta s^2 t^3\left(\frac{Rr}{\varepsilon}\right)\right)$$

queries to the input matrices will be made, and the same order of other gates, for each approximation of a $\mathrm{Tr}(A_j\rho)$ (and similarly for approximating $\mathrm{Tr}(C\rho)$).

Total cost of one iteration. Lemma 11.15 tells us that we will use $\widetilde{\mathcal{O}}(\sqrt{m})$ calculations of $\mathrm{Tr}(A_j\rho)$, and the same order of other gates, to calculate a classical description of a 2-sparse $y^{(t)}$. This brings the total cost of one iteration to

$$\widetilde{\mathcal{O}}\left(\sqrt{nm}\eta s^2 t^3\left(\frac{Rr}{\varepsilon}\right)\right)$$

queries to the input matrices, and the same order of other gates.

Total quantum runtime for SDPs. Since $w \leq r+1$ we can set $T = \widetilde{\mathcal{O}}\left(\frac{R^2r^2}{\varepsilon^2}\right)$. With $\eta = \sqrt{\frac{\ln(n)}{T}}$, summing over all iterations in one run of the algorithm gives a total cost of

$$\widetilde{\mathcal{O}}\left(\sum_{t=1}^{T}\sqrt{nm}\eta s^2 t^3\left(\frac{Rr}{\varepsilon}\right)\right) = \widetilde{\mathcal{O}}\left(\sqrt{nm}\eta s^2 T^4\left(\frac{Rr}{\varepsilon}\right)\right)$$
$$= \widetilde{\mathcal{O}}\left(\sqrt{nm}s^2\left(\frac{Rr}{\varepsilon}\right)^8\right)$$

queries to the input matrices and the same order of other gates. □

Total quantum runtime for LPs. The final complexity of our algorithm contains a factor $\widetilde{\mathcal{O}}(sT)$ that comes from the sparsity of the $H^{(t)}$ matrix. This assumes that when we add the input matrices together, the rows become less sparse. This need not happen for certain SDPs. For example, in the SDP relaxation of MAX-CUT, the $H^{(t)}$ will always be d-sparse, where d is the degree of the graph. A more important class of examples is that of linear programs: since LPs have diagonal A_j and C, their sparsity is $s = 1$, and even the sparsity of the $H^{(t)}$ is always 1. This, plus the fact that the traces can be computed without a factor $\|H\|$ in the complexity (as shown in Corollary 11.13 in Section 11.2.2), means that our algorithm solves LPs with

$$\widetilde{\mathcal{O}}\left(\sqrt{nm}\left(\frac{Rr}{\varepsilon}\right)^5\right)$$

queries to the input matrices and the same order of other gates.

11.3 Downside of this method: general oracles are restrictive

In this section we show some of the limitations of a method that uses sparse or general oracles, i.e., ones that are not optimized for the properties of specific SDPs. We will start by discussing sparse oracles in the next section. We will use a counting argument to show that sparse solutions cannot hold too much information about a problem's solution. In Section 11.3.2 we will show that width-bounds that do not depend on the specific structure of an SDP are for many problems not efficient.

11.3.1 Sparse oracles are restrictive

Lemma 11.16. *If, for some specific SDP of the form* (11.1)*, every ε-optimal dual-feasible vector has at least ℓ non-zero elements, then the width w of any k-sparse $\mathsf{Oracle}_{\varepsilon/3}$ for this SDP is such that $\frac{Rw}{\varepsilon} = \Omega\left(\sqrt{\frac{\ell}{k\ln(n)}}\right)$.*

Proof. The vector $\bar{y}$ returned by Meta-Algorithm 1 is, by construction, the average of T vectors $y^{(t)}$ that are all k-sparse, plus one extra 1-sparse term of $\frac{\varepsilon}{R}e_1$, and hence $\ell \leq kT + 1$. The stated bound on $\frac{Rw}{\varepsilon}$ then follows directly by combining this inequality with $T = \mathcal{O}(\frac{R^2w^2}{\varepsilon^2}\ln(n))$. □

The oracle presented in Section 11.2.3 always provides a 2-sparse vector y. This implies that if an SDP requires an ℓ-sparse dual solution, we must have $\frac{Rw}{\varepsilon} = \Omega(\sqrt{\ell/\ln(n)})$. This in turn means that the upper bound on the runtime of our algorithm will be of order $\ell^{7/2}\sqrt{nm}s^{3/2}$. This is clearly bad if ℓ is of the order n or m.

Of course it could be the case that almost every SDP of interest has a sparse approximate dual solution (or can easily be rewritten so that it does), and hence sparseness might be not a restriction at all. However, as we will see below, this is not the case. We will prove that for certain kinds of SDPs, no "useful" dual solution can be very sparse. Intuitively, a dual solution to an SDP is "useful" if it can be turned into a solution of the problem that the SDP is trying to solve. We make this more precise in the definition below.

Definition 11.17. *A problem is defined by a function f that, for every element p of the problem domain $\mathcal{D}$, gives a subset of the solution space $\mathcal{S}$, consisting of the solutions that are considered correct. We say a family of SDPs, $\{SDP^{(p)}\}_{p\in\mathcal{D}}$, solves the problem via the dual if there is an $\varepsilon \geq 0$ and a function g such that for every $p \in \mathcal{D}$ and every ε-optimal dual-feasible vector $y^{(p)}$ to $SDP^{(p)}$:*

$$g(y^{(p)}) \in f(p).$$

In other words, an ε-optimal dual solution can be converted into a correct solution of the original problem without more knowledge of p.

For these kinds of SDP families we will prove a lower bound on the sparsity of the dual solutions. The idea for this bound is as follows. If you have a lot of different instances that require different solutions, but the SDPs are equivalent up to permuting the constraints and the coordinates of $\mathbb{R}^n$, then a dual solution vector should have a lot of unique permutations and hence cannot be too sparse.

Theorem 11.18. *Consider a problem and a family of SDPs as in Definition 11.17. Let $\mathcal{T} \subseteq \mathcal{D}$ be such that for all $p, q \in \mathcal{T}$:*

- *$f(p) \cap f(q) = \emptyset$. That is, a solution to p is not a solution to q and vice versa.*
- *The number of constraints m and the primal variable size n are the same for $SDP^{(p)}$ and $SDP^{(q)}$.*
- *Let $A_j^{(p)}$ be the constraints of $SDP^{(p)}$ and $A_j^{(q)}$ those from $SDP^{(q)}$ (and define $C^{(p)}$, $C^{(q)}$, $b_j^{(p)}$, and $b_j^{(q)}$ in the same manner). Then there exist $\sigma \in S_n$, $\pi \in S_m$ s.t. $\sigma^{-1}A_{\pi(j)}^{(p)}\sigma = A_j^{(q)}$ (and $\sigma^{-1}C^{(p)}\sigma = C^{(q)}$). That is, the SDPs are the same up to permutations of the labels of the constraints and permutations of the coordinates of $\mathbb{R}^n$.*

If $y^{(p)}$ is an ε-optimal dual-feasible vector to $SDP^{(p)}$ for some $p \in \mathcal{T}$, then $y^{(p)}$ is at least $\frac{\log(|\mathcal{T}|)}{\log m}$-dense (i.e., has at least that many non-zero entries).

Proof. We first observe that, with $SDP^{(p)}$ and $SDP^{(q)}$ as in the lemma, if $y^{(p)}$ is an ε-optimal dual-feasible vector of $SDP^{(p)}$, then $y^{(q)}$ defined by

$$y_j^{(q)} := y_{\pi(j)}^{(p)} = \pi(y^{(p)})_j$$

is an ε-optimal dual vector for $SDP^{(q)}$. Here we use the fact that a permutation of the n coordinates in the primal does not affect the dual solutions. Since $f(p) \cap f(q) = \emptyset$ we know that $g(y^{(p)}) \neq g(y^{(q)})$ and so $y^{(p)} \neq y^{(q)}$. Since this is true for every q in $\mathcal{T}$, there should be at least $|\mathcal{T}|$ different vectors $y^{(q)} = \pi(y^{(p)})$.

A k-sparse vector can have k different non-zero entries and hence the number of possible unique permutations of that vector is at most

$$\binom{m}{k}k! = \frac{m!}{(m-k)!} = \prod_{t=m-k+1}^{m} t \leq m^k$$

so

$$\frac{\log|\mathcal{T}|}{\log m} \leq k. \qquad \square$$

Example. Consider the (s,t)-mincut problem, i.e., the dual of the (s,t)-maxflow. Specifically, consider a simple instance of this problem: the union of two complete graphs of size $z+1$, where s is in one subgraph and t in the other. Let the other vertices be labeled by $\{1,2,\ldots,2z\}$. Every assignment of the labels over the two halves gives a unique mincut, in terms of which labels fall on which side of the cut. There is exactly one partition of the vertices in two sets that cuts no edges (namely the partition consisting of the two complete graphs), and every other partition cuts at least z edges. Hence a $z/2$-approximate cut is a mincut. This means that there are $\binom{2z}{z}$ problems that require a different output. So for every family of SDPs that is symmetric under permutation of the vertices and for which a $z/2$-approximate dual solution gives an (s,t)-mincut, the sparsity of a $z/2$-approximate dual solution is at least[6]

$$\frac{\log\binom{2z}{z}}{\log m} \geq \frac{z}{\log m},$$

where we used that $\binom{2z}{z} \geq \frac{2^{2z}}{2\sqrt{z}}$.

11.3.2 General width-bounds are restrictive for certain SDPs

In this section we will show that width-bounds can be restrictive when they do not consider the specific structure of an SDP.

Definition 11.19. *An algorithm is called a* general oracle *if it implements an oracle for all SDPs, optimal value guesses α and error parameters ε. A function $w(n,m,s,r,R,\varepsilon)$ is called a* general width-bound *for a general oracle if it is a correct width-bound for that oracle, for every SDP with parameters n,m,s,r,R,ε. In particular, the function w may not depend on the structure of the input $A_1,\ldots,A_m$, C, b or on the value of α.*

We will show that general width-bounds need to scale with r^* (recall that r^* denotes the smallest ℓ_1-norm of an optimal solution to the dual). We then go on to show that if two SDPs in a class can be combined to get another element of that class in a natural manner, then, under some mild conditions, r^* will be of the order n and m for some instances of the class.

We start by showing, for specifically constructed LPs, a lower bound on the width of any oracle. Although these LPs will not solve any useful problem, every

[6] Here m is the number of constraints, not the number of edges in the graph.

general width-bound should also apply to these LPs. This gives a lower bound on general width-bounds.

Lemma 11.20. *For every $n \geq 3$, $m \geq 3$, $s \geq 1$, $R^* \geq 1$, $r^* > 0$, and $\varepsilon \leq 1/2$ there is an SDP with these parameters for which every oracle with precision ε has width at least $\frac{1}{2}r^*$.*

Proof. We will construct an LP for $n = m = 3$. This is enough to prove the lemma since LPs are a subclass of SDPs and we can increase n, m, and s by adding more dimensions and s-sparse SDP constraints that do not influence the analysis below. For some $k > 0$, consider the following LP

$$\begin{aligned} \max \quad & x_1 \\ \text{s.t.} \quad & \begin{bmatrix} 1 & 1 & 1 \\ 1/k & 1 & 0 \\ -1 & 0 & -1 \end{bmatrix} x \leq \begin{bmatrix} R \\ 0 \\ -R \end{bmatrix} \\ & x \geq 0 \end{aligned}$$

where the first row is the primal trace constraint. Notice that $x_1 = x_2 = 0$ due to the second constraint. This implies that OPT $= 0$ and, due to the last constraint, that $x_3 \geq R$. In fact, $(0, 0, R)$ is an optimal solution, so $R^* = R$.

To calculate r^*, look at the dual of the LP:

$$\begin{aligned} \min \quad & R(y_1 - y_3) \\ \text{s.t.} \quad & \begin{bmatrix} 1 & 1/k & -1 \\ 1 & 1 & 0 \\ 1 & 0 & -1 \end{bmatrix} y \geq \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ & y \geq 0, \end{aligned}$$

due to strong duality its optimal value is 0 as well. This implies $y_1 = y_3$, so the first constraint becomes $y_2 \geq k$. This in turn implies $r^* \geq k$, which is actually attained (by $y = (0, k, 0)$) so $r^* = k$.

Since the oracle and width-bound should work for every $x \in \mathbb{R}^3_+$ and every α, they should in particular work for $x = (R, 0, 0)$ and $\alpha = 0$. In this case the polytope for the oracle becomes

$$\begin{aligned} \mathcal{P}_\varepsilon(x) := \{y \in \mathbb{R}^m : \; & y_1 - y_3 \leq 0, \\ & (y_1 - y_3 + \frac{y_2}{k})R \geq R - \varepsilon, \\ & y \geq 0\}. \end{aligned}$$

This implies that for every $y \in \mathcal{P}_\varepsilon(x)$, we have $y_2 \geq k(1 - \frac{\varepsilon}{R}) \geq k/2 = r^*/2$.

Notice that the term

$$\left\| \sum_{j=1}^{m} y_j A_j - C \right\|$$

in the definition of width for an SDP becomes

$$\left\|A^T y - c\right\|_\infty$$

in the case of an LP. In our case, due to the second constraint in the dual, we know that

$$\left\|A^T y - c\right\|_\infty \geq y_1 + y_2 \geq \frac{r^*}{2}$$

for every vector y from $\mathcal{P}_\varepsilon(x)$. This shows that any oracle has width at least $r^*/2$ for this LP. □

Corollary 11.21. *For every general width-bound $w(n, m, s, r, R, \varepsilon)$, if $n, m \geq 3$, $s \geq 1$, $r > 0$, $R \geq 1$, and $\varepsilon \leq 1/2$, then*

$$w(n, m, s, r, R, \varepsilon) \geq \frac{r}{2}.$$

Note that this bound applies to both our algorithm and the one given by Brandão and Svore. It turns out that for many natural classes of SDPs, r^*, R^*, ε, n and m can grow linearly for some instances. In particular, this is the case if SDPs in a class combine in a natural manner. Take for example two SDP relaxations for the MAXCUT problem on two graphs $G^{(1)}$ and $G^{(2)}$ (on $n^{(1)}$ and $n^{(2)}$ vertices, respectively):

$$\begin{array}{ll} \max & \mathrm{Tr}\Big(L(G^{(1)})X^{(1)}\Big) \\ \text{s.t.} & \mathrm{Tr}\Big(X^{(1)}\Big) \leq n^{(1)} \\ & \mathrm{Tr}\Big(E_{jj}X^{(1)}\Big) \leq 1 \text{ for } j \in [n^{(1)}] \\ & X^{(1)} \succeq 0 \end{array} \qquad \begin{array}{ll} \max & \mathrm{Tr}\Big(L(G^{(2)})X^{(2)}\Big) \\ \text{s.t.} & \mathrm{Tr}\Big(X^{(2)}\Big) \leq n^{(2)} \\ & \mathrm{Tr}\Big(E_{jj}X^{(2)}\Big) \leq 1 \text{ for } j \in [n^{(2)}] \\ & X^{(2)} \succeq 0 \end{array}$$

Where $L(G)$ is the Laplacian of a graph. Note that this is not normalized to operator norm ≤ 1, but for simplicity we ignore this here. For the disjoint union of the two graphs, we have

$$L(G^{(1)} \cup G^{(2)}) = L(G^{(1)}) \oplus L(G^{(2)}).$$

This, plus the fact that the trace distributes over direct sums of matrices, means that the SDP relaxation for MAXCUT on $G^{(1)} \cup G^{(2)}$ is the same as a natural combination of the two separate maximizations:

$$\begin{array}{ll} \max & \mathrm{Tr}\Big(L(G^{(1)})X^{(1)}\Big) + \mathrm{Tr}\Big(L(G^{(2)})X^{(2)}\Big) \\ \text{s.t.} & \mathrm{Tr}\Big(X^{(1)}\Big) + \mathrm{Tr}\Big(X^{(2)}\Big) \leq n^{(1)} + n^{(2)} \\ & \mathrm{Tr}\Big(E_{jj}X^{(1)}\Big) \leq 1 \text{ for } j = 1, \ldots, n^{(1)} \\ & \mathrm{Tr}\Big(E_{jj}X^{(2)}\Big) \leq 1 \text{ for } j = 1, \ldots, n^{(2)} \\ & X^{(1)}, X^{(2)} \succeq 0. \end{array}$$

It is easy to see that the new value of n is $n^{(1)} + n^{(2)}$, the new value of m is $m^{(1)} + m^{(2)} - 1$ and the new value of R^* is $n^{(1)} + n^{(2)} = R^{*(1)} + R^{*(2)}$. Since it is natural for the MAXCUT relaxation that the additive errors also add, it remains to see what happens to r^*, and so, for general width-bounds, what happens to w. As we will see later in this section, under some mild conditions, these kind of combinations imply that there are MAXCUT-relaxation SDPs for which r^* also increases linearly, but this requires a bit more work.

Definition 11.22. *We say that a class of SDPs (each with an associated allowed approximation error) is* combinable *if there is a $k \geq 0$ so that for every two elements in this class, $(SDP^{(a)}, \varepsilon^{(a)})$ and $(SDP^{(b)}, \varepsilon^{(b)})$, there is an instance in the class, $(SDP^{(c)}, \varepsilon^{(c)})$, that is a combination of the two in the following sense:*

- $C^{(c)} = C^{(a)} \oplus C^{(b)}$.
- $A_j^{(c)} = A_j^{(a)} \oplus A_j^{(b)}$ *and* $b_j^{(c)} = b_j^{(a)} + b_j^{(b)}$ *for* $j \in [k]$.
- $A_j^{(c)} = A_j^{(a)} \oplus \mathbf{0}$ *and* $b_j^{(c)} = b_j^{(a)}$ *for* $j = k+1, \ldots, m^{(a)}$.
- $A_{m^{(a)}+j-k}^{(c)} = \mathbf{0} \oplus A_j^{(b)}$ *and* $b_{m^{(a)}+j-k}^{(c)} = b_j^{(b)}$ *for* $j = k+1, \ldots, m^{(b)}$.
- $\varepsilon^{(c)} \leq \varepsilon^{(a)} + \varepsilon^{(b)}$.

In other words, some fixed set of constraints are summed pairwise, and the remaining constraints get added separately.

Note that this is a natural generalization of the combining property of the MAXCUT relaxations (in that case $k = 1$ to account for the trace bound).

Theorem 11.23. *If a class of SDPs is combinable and there is a $\delta > 0$ and an element $SDP^{(1)}$ for which every optimal dual solution has the property that*

$$\sum_{j=k+1}^{m} y_j \geq \delta,$$

then there is a sequence $(SDP^{(t)})_{t\in\mathbb{N}}$ in the class such that $\frac{R^{*(t)} r^{*(t)}}{\varepsilon^{(t)}}$ increases linearly in $n^{(t)}$, $m^{(t)}$ and t.

Proof. The sequence we will consider is the t-fold combination of $SDP^{(1)}$ with itself. If $SDP^{(1)}$ is

$$\begin{aligned} \max \quad & \mathrm{Tr}(CX) \\ \text{s.t.} \quad & \mathrm{Tr}(A_j X) \leq b_j \quad \text{for } j \in [m^{(1)}], \\ & X \succeq 0 \end{aligned} \qquad\qquad \begin{aligned} \min \quad & \sum_{j=1}^{m^{(1)}} b_j y_j \\ \text{s.t.} \quad & \sum_{j=1}^{m^{(1)}} y_j A_j - C \succeq 0, \\ & y \geq 0 \end{aligned}$$

then $SDP^{(t)}$ is

$$\begin{array}{lll} \max & \sum_{i=1}^{t} \mathrm{Tr}(CX_i) & \\ \text{s.t.} & \sum_{i=1}^{t} \mathrm{Tr}(A_jX_i) \leq tb_j & \text{for } j \in [k], \\ & \mathrm{Tr}(A_jX_i) \leq b_j & \text{for } j = k+1, \dots, m^{(1)} \text{ and } i = 1, \dots, t \\ & X_i \succeq 0 & \text{for all } i = 1, \dots, t \end{array}$$

with dual

$$\begin{array}{ll} \min & \sum_{j=1}^{k} tb_jy_j + \sum_{i=1}^{t} \sum_{j=k+1}^{m^{(1)}} b_jy_j^i \\ \text{s.t.} & \sum_{j=1}^{k} y_jA_j + \sum_{j=k+1}^{m^{(1)}} y_j^iA_j \succeq C \text{ for } i = 1, \dots, t \\ & y, y^i \geq 0. \end{array}$$

First, let us consider the value of $\mathrm{OPT}^{(t)}$. Let $X^{(1)}$ be an optimal solution to $SDP^{(1)}$ and for all $i \in [t]$ let $X_i = X^{(1)}$. Since these X_i form a feasible solution to $SDP^{(t)}$, this shows that $\mathrm{OPT}^{(t)} \geq t \cdot \mathrm{OPT}^{(1)}$. Furthermore, let $y^{(1)}$ be an optimal dual solution of $SDP^{(1)}$, then $(y_1^{(1)}, \dots, y_k^{(1)}) \oplus \left(y_{k+1}^{(1)}, \cdots, y_{m^{(1)}}^{(1)}\right)^{\oplus t}$ is a feasible dual solution for $SDP^{(t)}$ with objective value $t \cdot \mathrm{OPT}^{(1)}$, so $\mathrm{OPT}^{(t)} = t \cdot \mathrm{OPT}^{(1)}$.

Next, let us consider the value of $r^{*(t)}$. Let $\tilde{y} \oplus y^1 \oplus \cdots \oplus y^t$ be an optimal dual solution for $SDP^{(t)}$, split into the parts of y that correspond to different parts of the combination. Then $\tilde{y} \oplus y^i$ is a feasible dual solution for $SDP^{(1)}$ and hence $b^T(\tilde{y} \oplus y^i) \geq \mathrm{OPT}^{(1)}$. On the other hand we have

$$t \cdot \mathrm{OPT}^{(1)} = \mathrm{OPT}^{(t)} = \sum_{i=1}^{t} b^T(\tilde{y} \oplus y^i),$$

this implies that each term in the sum is actually equal to $\mathrm{OPT}^{(1)}$. But if $(\tilde{y} \oplus y^i)$ is an optimal dual solution of $SDP^{(1)}$ then $\left\|(\tilde{y} \oplus y^i)\right\|_1 \geq r^{*(1)}$ by definition and $\left\|y^i\right\|_1 \geq \delta$. We conclude that $r^{*(t)} \geq r^{*(1)} - \delta + t\delta$.

Now we know the behavior of r^* under combinations, let us look at the primal to find a similar statement for $R^{*(t)}$. Define a new SDP, $\widehat{SDP}^{(t)}$, in which all the constraints are summed when combining, that is, in Definition 11.22 we take

$k = n^{(1)}$, however, contrary to that definition, we even sum the psd constraints:

$$\begin{aligned} \max \quad & \sum_{i=1}^{t} \mathrm{Tr}(CX_i) \\ \text{s.t.} \quad & \sum_{i=1}^{t} \mathrm{Tr}(A_j X_i) \leq t b_j \quad \text{for } j \in [m^{(1)}], \\ & \sum_{i=1}^{t} X_i \succeq 0. \end{aligned}$$

This SDP has the same objective function as $SDP^{(t)}$ but a larger feasible region: every feasible $X_1, \ldots, X_t$ for $SDP^{(t)}$ is also feasible for $\widehat{SDP}^{(t)}$. However, by a change of variables, $X := \sum_{i=1}^{t} X_i$, it is easy to see that $\widehat{SDP}^{(t)}$ is simply a scaled version of $SDP^{(1)}$. So, $\widehat{SDP}^{(t)}$ has optimal value $t \cdot \mathrm{OPT}^{(1)}$. Since optimal solutions to $\widehat{SDP}^{(t)}$ are scaled optimal solutions to $SDP^{(1)}$, we have $\hat{R}^{*(t)} = t \cdot R^{*(1)}$. Combining the above, it follows that every optimal solution to $SDP^{(t)}$ is optimal to $\widehat{SDP}^{(t)}$ as well, and hence has trace at least $t \cdot R^{*(1)}$, so $R^{*(t)} \geq t \cdot R^{*(1)}$.

We conclude that

$$\frac{R^{*(t)} r^{*(t)}}{\varepsilon^{(t)}} \geq \frac{t R^{*(1)} (r^{*(1)} + (t-1)\delta)}{t \varepsilon^{(1)}} = \Omega(t)$$

and $n^{(t)} = tn^{(1)}$, $m^{(t)} = t(m^{(1)} - k) + k$. □

This shows that for many natural SDP formulations for combinatorial problems, such as the MAXCUT relaxation, $R^* r^* / \varepsilon$ increases linearly in n and m for some instances. Hence, using $R^* \leq R$ and Lemma 11.20, Rw/ε grows at least linearly when a general width-bound is used.

11.4 Lower bounds on quantum query complexity

In this section we will show that every LP-solver (and hence every SDP-solver) that can distinguish (with high probability) between an optimal value being 0 or 1 needs

$$\Omega\left(\sqrt{\max\{n,m\}} (\min\{n,m\})^{3/2}\right)$$

quantum queries to the input in the worst case.

For the lower bound on LP-solving we will give a reduction from a composition of Majority and OR functions. Here the majority function on a bits is the function $MAJ_a : \{0,1\}^a \to \{0,1\}$ that maps $x \in \{0,1\}^a$ to 1 if $|x| > a/2$ and to 0 otherwise. We say that an input x to MAJ_a is a *boundary case* if $|x| = \frac{a}{2}$ or $|x| = \frac{a}{2} + 1$ (we assume a is even from now on). We have seen the OR function before in Chapter 9.

Definition 11.24. *Given input bits $Z_{ij\ell} \in \{0,1\}^{a\times b\times c}$ the problem of calculating*

$$\begin{aligned}
&MAJ_a(\\
&\qquad OR_b(MAJ_c(Z_{111},\dots,Z_{11c}),\dots,MAJ_c(Z_{1b1},\dots,Z_{1bc})), \\
&\qquad \dots, \\
&\qquad OR_b(MAJ_c(Z_{a11},\dots,Z_{a1c}),\dots,MAJ_c(Z_{ab1},\dots,Z_{abc})) \\
&\qquad)
\end{aligned}$$

with the promise that

- *Each inner* MAJ_c *is a boundary case, in other words* $\sum_{\ell=1}^{c} Z_{ij\ell} \in \{c/2, c/2{+}1\}$ *for all* i,j*.*
- *The outer* MAJ_a *is a boundary case, in other words, if* $\tilde{Z} \in \{0,1\}^a$ *is the bitstring that results from all the OR calculations, then* $|\tilde{Z}| \in \{a/2, a/2+1\}$*.*

is called the promise MAJ_a-OR_b-MAJ_c *problem.*

Lemma 11.25. *It takes at least* $\Omega(a\sqrt{b}\,c)$ *queries to the input to solve the promise* MAJ_a-OR_b-MAJ_c *problem.*

Proof. The promise version of MAJ_k is known to require $\Omega(k)$ quantum queries. Likewise, it is known that the OR_k function requires $\Omega(\sqrt{k})$ queries. Furthermore, it is known that the general adversary bound that we have mentioned in Chapter 10 is multiplicative under composition of functions; Kimmel [Kim13, Lemma A.3 (Lemma 6 in the arXiv version)] showed that this even holds for promise functions. Since the general adversary method characterizes quantum query complexity we have the same multiplicativity for quantum query complexity. Therefore, the quantum query complexity of MAJ_a-OR_b-MAJ_c is $\Omega(a\sqrt{b}\,c)$. □

Lemma 11.26. *Determining the value*

$$\sum_{i=1}^{a} \max\nolimits_{j\in[b]} \sum_{\ell=1}^{c} Z_{ij\ell}$$

for a Z from the promise MAJ_a-OR_b-MAJ_c *problem up to additive error* $\varepsilon = 1/3$*, solves the promise* MAJ_a-OR_b-MAJ_c *problem.*

Proof. Notice that due to the first promise, $\sum_{\ell=1}^{c} Z_{ij\ell} \in \{c/2, c/2+1\}$ for all $i \in [a], j \in [b]$. This implies that

- If the ith OR is 0, then all of its inner MAJ functions are 0 and hence
$$\max\nolimits_{j\in[b]} \sum_{\ell=1}^{c} Z_{ij\ell} = \frac{c}{2}$$
- If the ith OR is 1, then at least one of its inner MAJ functions is 1 and hence
$$\max\nolimits_{j\in[b]} \sum_{\ell=1}^{c} Z_{ij\ell} = \frac{c}{2} + 1$$

Now, if we denote the string of outcomes of the OR functions by $\tilde{Z} \in \{0,1\}^a$, then

$$\sum_{i=1}^{a} \max_{j\in[b]} \sum_{\ell=1}^{c} Z_{ij\ell} = a\frac{c}{2} + |\tilde{Z}|$$

Hence determining the left-hand side will determine $|\tilde{Z}|$; this Hamming weight is either $\frac{a}{2}$ if the full function evaluates to 0, or $\frac{a}{2}+1$ if it evaluates to 1. □

Lemma 11.27. *For an input $Z \in \{0,1\}^{a\times b\times c}$ there is an LP with $m = c + a$ and $n = c + ab$ for which the optimal value is*

$$\sum_{i=1}^{a} \max_{j\in[b]} \sum_{\ell=1}^{c} Z_{ij\ell}$$

Furthermore, a query to an entry of the input matrix or vector costs at most 1 *query to Z.*

Proof. Let $Z^{(i)}$ be the matrix one gets by fixing the first index of Z and putting the entries in a $c \times b$ matrix, so $Z^{(i)}_{\ell j} = Z_{ij\ell}$. We define the following LP:

$$\begin{aligned} \mathrm{OPT} = \max \quad & \sum_{k=1}^{c} w_k \\ \text{s.t.} \quad & \begin{bmatrix} I & -Z^1 & \cdots & -Z^a \\ 0 & \mathbf{1}^T & & \\ 0 & & \ddots & \\ 0 & & & \mathbf{1}^T \end{bmatrix} \begin{bmatrix} w \\ v^{(1)} \\ \vdots \\ v^{(a)} \end{bmatrix} \leq \begin{bmatrix} 0 \\ \mathbf{1} \\ \vdots \\ \mathbf{1} \end{bmatrix} \\ & v^1, \ldots, v^a \in \mathbb{R}^b_+, w \in \mathbb{R}^c_+ \end{aligned}$$

Notice every $Z^{(i)}$ is of size $c \times b$, so that indeed $m = c + a$ and $n = c + ab$.

For every $i \in [a]$ there is a constraint that says

$$\sum_{j=1}^{b} v_j^{(i)} \leq 1.$$

The constraints involving w say that for every $k \in [c]$

$$w_k \leq \sum_{i=1}^{a}\sum_{j=1}^{b} v_j^{(i)} Z_{kj}^{(i)} = \sum_{i=1}^{a} (Z^{(i)}v^{(i)})_k$$

where $(Z^{(i)}v^{(i)})_k$ is the kth entry of the matrix-vector product $Z^{(i)}v^{(i)}$. Clearly, for an optimal solution these constraints will be satisfied with equality, since in the objective function w_k has a positive weight. Summing over k on both sides, we get the equality

$$\begin{aligned}\mathrm{OPT} &= \sum_{k=1}^{c} w_k \\ &= \sum_{k=1}^{c}\sum_{i=1}^{a}(Z^{(i)}v^{(i)})_k \\ &= \sum_{i=1}^{a}\sum_{k=1}^{c}(Z^{(i)}v^{(i)})_k \\ &= \sum_{i=1}^{a}\left\|Z^{(i)}v^{(i)}\right\|_1\end{aligned}$$

so in the optimum $\left\|Z^{(i)}v^{(i)}\right\|_1$ will be maximized. Note that we can use the ℓ_1-norm as a shorthand for the sum over vector elements since all elements are positive. In particular, the value of $\left\|Z^{(i)}v^{(i)}\right\|_1$ is given by

$$\begin{aligned}\max \quad & \left\|Z^{(i)}v^{(i)}\right\|_1 \\ \text{s.t.} \quad & \left\|v^{(i)}\right\|_1 \leq 1 \\ & v^{(i)} \geq 0\end{aligned}$$

Now $\|Z^{(i)}v^{(i)}\|_1$ will be maximized by putting all weight in $v^{(i)}$ on the index that corresponds to the column of $Z^{(i)}$ that has the highest Hamming weight. In particular in the optimum $\|Z^{(i)}v^{(i)}\|_1 = \max_{j\in[b]}\sum_{\ell=1}^{c} Z^{(i)}_{\ell j}$. Putting everything together gives:

$$\mathrm{OPT} = \sum_{i=1}^{a}\left\|Z^{(i)}v^{(i)}\right\|_1 = \sum_{i=1}^{a}\max_{j\in[b]}\sum_{\ell=1}^{c} Z^{(i)}_{\ell j} = \sum_{i=1}^{a}\max_{j\in[b]}\sum_{\ell=1}^{c} Z_{ij\ell}$$ □

Theorem 11.28. *There is a family of LPs, with $m \leq n$ and two possible integer optimal values, that require at least $\Omega(\sqrt{n}m^{3/2})$ quantum queries to the input to distinguish those two values.*

Proof. Let $a = c = m/2$ and $b = \frac{n-c}{a} = \frac{2n}{m} - 1$, so that $n = c + ab$ and $m = c + a$. By Lemma 11.27 there exists an LP with $n = c + ab$ and $m = c + a$ that calculates

$$\sum_{i=1}^{a}\max_{j\in[b]}\sum_{\ell=1}^{c} Z_{ij\ell}$$

for an input Z to the promise MAJ_a-OR_b-MAJ_c problem. By Lemma 11.26, calculating this value will solve the promise MAJ_a-OR_b-MAJ_c problem. By Lemma 11.25 the promise MAJ_a-OR_b-MAJ_c problem takes $\Omega(a\sqrt{b}c)$ quantum queries in the worst case. This implies a lower bound of

$$\Omega\left(m^2\sqrt{\frac{n}{m}}\right) = \Omega(m^{3/2}\sqrt{n})$$

quantum queries on solving these LPs. □

Corollary 11.29. *Distinguishing two optimal values of an LP (and hence also of an SDP) with additive error $\varepsilon < 1/2$ requires*

$$\Omega\Big(\sqrt{\max\{n,m\}}(\min\{n,m\})^{3/2}\Big)$$

quantum queries to the input matrices in the worst case.

Proof. Since the roles of m and n are exchanged by passing to the dual, the result follows from Theorem 11.28. □

It is important to note that the parameters R and r from the Arora-Kale algorithm are not constant in this family of LPs ($R, r = \Theta(\min\{n,m\}^2)$ here), and hence this lower bound does not contradict the scaling with $\sqrt{mn}$ of the complexity of our SDP-solver or Brandão and Svore's. Since we show in the appendix that one can always rewrite the LP (or SDP) so that 2 of the parameters R, r, ε are constant, the lower bound implies that any algorithm with a sub-linear dependence on m or n has to depend at least polynomially on Rr/ε. For example, the above family of LPs shows that an algorithm with a $\sqrt{mn}$ dependence has to have an $(Rr/\varepsilon)^\kappa$ factor in its complexity with $\kappa \geq 1/4$.

11.5 Discussion and related work

In this chapter we have seen better algorithms and lower bounds for quantum SDP-solvers. Below we briefly point to related work, but first we mention some directions for future work:

- **Applications of our algorithm.** As mentioned, both our and Brandão-Svore's quantum SDP-solvers only improve upon the best classical algorithms for a specific regime of parameters, namely where $mn \gg Rr/\varepsilon$. Unfortunately, we don't know particularly interesting problems in combinatorial optimization in this regime. As shown in Section 11.3, many natural SDP formulations will not fall into this regime. Therefore, it would be interesting to find useful SDPs for which our algorithm gives a significant speed-up.

  In subsequent work [vAG18a] van Apeldoorn and Gilyen have described two possible applications where a speed-up in terms of some parameter is possible. (Below we state the complexity bound that they achieve using their improved quantum SDP-solver.)

  – They show how the *quantum state discrimination problem* can be solved up to additive error ε using $\widetilde{\mathcal{O}}(\frac{\sqrt{k}}{\varepsilon^5}\mathrm{poly}(d))$ queries to the input. Here the quantum state discrimination problem can be described as follows: find a POVM $\{M^{(i)}\}_{i\in[k]}$ that best discriminates a given set of states $\rho^{(1)}, \ldots, \rho^{(k)} \in \mathrm{S}^d_+$. Here 'best discriminates' is measured according to the objective $\max \sum_i \mathrm{Tr}(M^{(i)}\rho^{(i)})$.

  – They show how to determine the 'optimal' weighting of k experiments with which you can learn a hidden d-dimensional state $\theta \in \mathbb{R}^d$. Here the

ith experiment produces a sample from a normal distribution centered at $\langle \theta, u^{(i)} \rangle$. We refer to their paper for the definition of 'optimal'. A quantum SDP-solver can find an approximately optimal weighting in time that scales as $\sqrt{k}$.

- **New algorithms.** As in the work by Arora and Kale, it might be more promising to look at oracles (now quantum) that are designed for specific SDPs. Such oracles could build on the techniques developed here, or develop totally new techniques. It might also be possible to speed up other classical SDP-solvers, for example those based on interior-point methods; we mention a first step in this direction below.

Subsequent work. Following the first version of [vAGGdW17], improvements in the running time were obtained in [BKL$^+$17, vAG18a], the latter providing a runtime of $\widetilde{\mathcal{O}}\Big((\sqrt{m} + \sqrt{n}\frac{Rr}{\varepsilon})s\big(\frac{Rr}{\varepsilon}\big)^4\Big)$. In addition to the sparse input model, these works also consider different input models. Most notably, in light of the presentation of the results in this chapter, they also consider a model where we are given access to the input matrices via block-encodings.

More recently, a quantum interior point method for solving SDPs and LPs was obtained by Kerenidis and Prakash [KP18]. It is hard to compare the latter algorithm to the other SDP-solvers for two reasons. First, the output of their algorithm consists only of almost-feasible solutions to the primal and dual (their algorithm has a polynomial dependence on the distance to feasibility). It is therefore not clear what their output means for the optimal value of the SDPs. Secondly, the runtime of their algorithm depends polynomially on the condition number of the matrices that the interior point method encounters, and no explicit bounds for these condition numbers are given.

The recent work on quantum SDP-solvers is a step towards solving optimization problems faster using quantum computers. Another such step is taken recently in the direction of black-box convex optimization, where one optimizes over a general convex set K, and the access to K is via membership and/or separation oracles [vAGGdW18, CCLW18]. We will discuss this in much more detail in the next chapter.

Chapter 12

Quantum algorithms for convex optimization

This chapter is based on the paper "Convex optimization using quantum oracles", by J. van Apeldoorn, A. Gilyén, S. Gribling, R. de Wolf [vAGGdW18].

12.1 Introduction

One of the most successful optimization paradigms is *convex* optimization, which optimizes a convex function over a convex set that is given explicitly (by a set of constraints) or implicitly (by an oracle). See for instance the classical work of Grötschel, Lovász, and Schrijver [GLS88] or the recent survey of Bubeck [Bub15]. Recent experimental progress on building quantum computers created a surge of interest in the following question: can we solve optimization problems more efficiently by exploiting quantum effects such as superposition, interference, and entanglement? In the previous chapter we have studied that question in the setting of semidefinite optimization problems, where the constraints are given explicitly. In this chapter we study investigate to what extent quantum computers can help solve convex optimization problems when the constraints are given implicitly, the 'black-box' model.

To be more concrete, let us formally state the problem (see also Section 1.2). The general *convex optimization problem* is to maximize a linear function $c^T x$ over points $x \in K \subseteq \mathbb{R}^n$, where K is a closed convex set and c is a unit vector $\|c\|_2 = 1$.

$$\max \quad c^T x \quad \text{s.t. } x \in K. \tag{12.1}$$

Unless explicitly stated otherwise, we assume that a point $x_0 \in \mathbb{R}^n$ and radii $r, R > 0$ are known that satisfy $B(x_0, r) \subseteq K \subseteq B(x_0, R)$. Here $B(x_0, r)$ is the Euclidean ball of radius r centered at x_0.

Previously we have seen an important special class of convex optimization problems: semidefinite programs (SDPs). In an SDP the convex set K is the intersection of an affine subspace with the cone of positive semidefinite matrices. In Chapter 11 we have seen a quantum SDP-solver. This quantum SDP-solver made explicit use of the structure of the cone of positive semidefinite matrices; the structure was crucial to the algorithm. In this chapter we study to what extent quantum computers are useful for solving general convex optimization problems. But, unlike the previous chapter, we do so in the setting where access to the convex set is given only in a black-box manner, through an oracle. We consider the usual membership, separation, optimization, violation, and validity oracles (see Section 12.2 for the definitions). We examine the efficiency of reductions between the different oracles in terms of the underlying dimension n. That is, given an oracle O for one of these problems, how many applications of O do we need to implement an oracle for any of the other problems? It is known since the 1980s that all these oracles are polynomial-time equivalent on classical computers [GLS88]. Subsequent work made these polynomial-time reductions more efficient, reducing the degree of the polynomials. We now study this problem from a quantum perspective: given quantum query access to an oracle O for one of these problems, how many *quantum* queries to O do we need to implement a *classical* oracle for any of the other problems?

Let us highlight a recent work in the classical setting that formed the starting point for the work in this chapter. Recently Lee, Sidford, and Vempala [LSV18] showed that with[1] $\widetilde{\mathcal{O}}\big(n^2\big)$ calls to a membership oracle (and $\widetilde{\mathcal{O}}\big(n^3\big)$ other elementary arithmetic operations) one can implement an optimization oracle, offering a significant improvement over the original reduction of Grötschel, Lovász, and Schrijver [GLS88].[2] They did so by showing that $\widetilde{\mathcal{O}}(n)$ calls to a membership oracle suffice to do separation, and then composing this with the known fact [LSW15] (see also [LSV18, Theorem 15]) that $\widetilde{\mathcal{O}}(n)$ calls to a separation oracle suffice for optimization.

Our main algorithmic result (Section 12.4) shows that on a quantum computer $\widetilde{\mathcal{O}}(1)$ calls to a membership oracle suffice to implement a separation oracle, and hence, using the classical reduction, $\widetilde{\mathcal{O}}(n)$ quantum queries to a membership oracle suffice for optimization (the best known classical upper bound on the number of membership queries is quadratic).[3] Besides this algorithmic result, we also prove several lower bounds on the efficiency of (quantum) reductions between the five oracles.

In the remainder of this section we first give an overview of related work on quantum optimization algorithms, and then we give an overview of our results. The latter will also serve as a roadmap to the rest of this chapter.

[1] Here, and in the rest of this chapter, the notation $\widetilde{\mathcal{O}}(\cdot)$ is used to hide polylogarithmic factors in n, r, R and the desired additive error ε.

[2] The original reduction of [GLS88] uses the ellipsoid method (twice) and appears to use $\Omega(n^{10})$ membership queries in order to implement an optimization oracle.

[3] Although not stated explicitly in our results, we also use $\widetilde{\mathcal{O}}\big(n^3\big)$ additional operations for optimization using membership, like [LSV18]. This is because our quantum algorithm for separation uses only $\widetilde{\mathcal{O}}(n)$ gates in addition to the $\widetilde{\mathcal{O}}(1)$ membership queries, and we use the same reduction from optimization to separation as [LSV18].

12.1.1 Related work

Quantum optimization. Quantum algorithms for solving convex optimization problems have been considered before. In 2008, Jordan [Jor08] described a faster quantum algorithm for minimizing quadratic functions. Recently, for an important class of convex optimization problems (semidefinite optimization) quantum speed-ups were achieved using algorithms whose runtime scales polynomially with the desired precision and some geometric parameters [BS17, vAGGdW17, BKL$^+$17, vAG18a] (see also Chapter 11). However, many convex optimization problems can be solved classically using algorithms whose runtime scales *logarithmically* with the desired precision and the relevant geometric parameters. We are aware of only one quantum speed-up which is partially in this regime, namely the very recent quantum interior point method of Kerenidis and Prakash [KP18]. In this chapter we look at general convex optimization problems, considering algorithms that have such favorable logarithmic scaling with the precision.

Related independent work. In independent simultaneous work, Chakrabarti, Childs, Li, and Wu [CCLW18] discovered a similar upper bound as ours: combining the recent classical work of Lee et al. [LSV18] with a quantum algorithm for computing gradients, they show how to implement an optimization oracle via $\widetilde{\mathcal{O}}(n)$ quantum queries to a membership oracle and to an oracle for the objective function. Their proof stays quite close to [LSV18] while ours first simplifies some of the technical lemmas of [LSV18], giving us a slightly simpler presentation and a better error-dependence of the resulting algorithm.

12.1.2 Our results

Recall that our main algorithmic result is that, on a quantum computer, $\widetilde{\mathcal{O}}(1)$ calls to a membership oracle suffice to implement a separation oracle, and hence (by the known classical reduction from optimization to separation) $\widetilde{\mathcal{O}}(n)$ calls to a membership oracle suffice for optimization. The proof of this result is inspired by the work of Lee et al. [LSV18]. They used a geometric idea to reduce separation to finding an approximate subgradient of a convex Lipschitz function. They then showed that $\widetilde{\mathcal{O}}(n)$ evaluations of a convex Lipschitz function suffice to get an approximate subgradient. We use the same geometric idea, but we provide a simpler way to compute an approximate subgradient of a convex Lipschitz function (Section 12.3). We point out that this new algorithm is purely classical. But, besides being simpler, the main advantage of our algorithm is that it is suitable for a quantum speed-up using known quantum algorithms for computing approximate (sub)gradients [Jor05, GAW], which we show in Section 12.4.

As a second set of results, in Section 12.5 we provide lower bounds on the number of membership or separation queries needed to implement several other oracles. We show that our quantum reduction from separation to membership indeed improves over the best possible classical reduction: $\Omega(n)$ classical membership queries are needed to do separation.[4] We only have partial results regarding the optimality of

[4] We are not aware of an existing proof of this classical lower bound, but it may well be some-

the reduction from optimization to separation. In the setting where we are not given an interior point of the set K, we can prove an essentially optimal $\Omega(n)$ lower bound on the number of quantum queries to a separation oracle needed to do optimization. However, for the case of quantum algorithms that *do* know an interior point, we are only able to prove an $\Omega(\sqrt{n})$ lower bound. In the classical setting, regardless of whether or not we know an interior point, the reduction uses $\widetilde{\Theta}(n)$ queries. This raises the interesting question of whether knowing an interior point can lead to a better quantum algorithm. We therefore view closing the gap between upper and lower bound as an important direction for future work.

Finally, we briefly mention (Section 12.6) how to obtain upper and lower bounds for some of the other oracle reductions, using a convex polarity argument. As we show, in the setting where we are given an interior point, the relation between membership and separation is analogous to the relation between validity and optimization. In particular, our better quantum algorithm for separation using membership queries implies that on a quantum computer $\widetilde{\mathcal{O}}(1)$ queries to a validity oracle suffice to implement an optimization oracle. That is, on a quantum computer, finding the optimal value is equivalent to finding an optimizer. Also, the same polarity argument shows that algorithms for optimization using separation are essentially equivalent to algorithms for separation using optimization. In particular, this turns our lower bound on the number of separation queries needed to implement an optimization oracle into a lower bound on the reverse direction.

We have summarized the current state of the art (informally) in Figure 12.1, the bold-face entries indicate our results; the (change in) accuracy is ignored here for simplicity. The above-mentioned polarity manifests itself in the central symmetry of the figure.

Classical:

$$\begin{tikzcd}
\boxed{\mathrm{MEM}(K)} \arrow[r, shift right=1.5, "\widetilde{\Theta}(n)"'] & \boxed{\mathrm{SEP}(K)} \arrow[l, shift right=1.5, "\Theta(1)"'] \arrow[r, shift right=1.5, "\widetilde{\Theta}(n)"'] & \boxed{\mathrm{OPT}(K)} \arrow[l, shift right=1.5, "\widetilde{\Theta}(n)"'] \arrow[r, shift right=1.5, "\Theta(1)"'] & \boxed{\mathrm{VAL}(K)} \arrow[l, shift right=1.5, "\widetilde{\Theta}(n)"']
\end{tikzcd}$$

Quantum:

$$\begin{tikzcd}
\boxed{\mathrm{MEM}(K)} \arrow[r, shift right=1.5, "\boldsymbol{\widetilde{\Theta}(1)}"'] & \boxed{\mathrm{SEP}(K)} \arrow[l, shift right=1.5, "\Theta(1)"'] \arrow[r, shift right=1.5, "\substack{\widetilde{\mathcal{O}}(n) \\ \boldsymbol{\Omega(n)^*}}"'] \arrow[r, phantom, "\boldsymbol{\Omega(\sqrt{n})}"] & \boxed{\mathrm{OPT}(K)} \arrow[l, shift right=1.5, "\widetilde{\mathcal{O}}(n)"'] \arrow[r, shift right=1.5, "\Theta(1)"'] & \boxed{\mathrm{VAL}(K)} \arrow[l, shift right=1.5, "\boldsymbol{\widetilde{\Theta}(1)}"']
\end{tikzcd}$$

Figure 12.1: The top and bottom diagram illustrate the relations between the basic (weak) oracles for respectively classical and quantum queries, with boldface entries marking our new results. All upper and lower bounds hold in the setting where we know an interior point of the convex set K, except the $*$-marked $\Omega(n)$ lower bound on the number of separation queries needed for optimization.

where in the vast literature on convex optimization.

12.2 Preliminaries

For $p \geq 1$, $\varepsilon \geq 0$, and a set $C \subseteq \mathbb{R}^n$ we let

$$B_p(C, \varepsilon) = \{x \in \mathbb{R}^n : \exists y \in C \text{ such that } ||x - y||_p \leq \varepsilon\}$$

be the set of points of distance at most ε from C in the ℓ_p-norm. When $C = \{x\}$ is a singleton set we abuse notation and write $B_p(x, \varepsilon)$. We overload notation by setting

$$B_p(C, -\varepsilon) = \{x \in \mathbb{R}^n : B_p(x, \varepsilon) \subseteq C\}.$$

Whenever p is omitted it is assumed that $p = 2$.

Recall that a function $f : C \to \mathbb{R}$ is *Lipschitz* if there exists a constant $L > 0$ such that

$$|f(y') - f(y)| \leq L\|y' - y\|_2 \text{ for all } y, y' \in C.$$

We write that f is *L-Lipschitz*. The inner product between vectors $v, w \in \mathbb{R}^n$ is $\langle v, w \rangle = v^T w$.

Definition 12.1 (Subgradient)**.** *Let $C \subseteq \mathbb{R}^n$ be convex and let x be an element of the interior of C. For a convex function $f : C \to \mathbb{R}$ we denote by $\underline{\partial} f(x)$ the set of subgradients of f at x, i.e., those vectors g satisfying*

$$f(y) \geq f(x) + \langle g, y - x \rangle \text{ for all } y \in C.$$

Note that in the above definition $\underline{\partial} f(x) \neq \emptyset$ due to convexity.

If $f : C \to \mathbb{R}$ is L-Lipschitz, then for any x in the interior of C and any $g \in \underline{\partial} f(x)$ we have $\|g\| \leq L$, as follows. Consider a $y \in C$ such that $y - x = \alpha g$ for some $\alpha > 0$. Then since g is a subgradient of f at x we have

$$\alpha \|g\|^2 = \langle g, y - x \rangle \leq f(y) - f(x) \leq L\|y - x\| = \alpha L \|g\|, \tag{12.2}$$

and therefore $\|g\| \leq L$.

We will assume familiarity with quantum computing [NC00]. In particular, a standard quantum oracle corresponds to a unitary transformation that acts on two registers, where the first register contains the query and the answer is added to the second register. For example, for $X \subseteq \mathbb{Q}^n$, $Y \subseteq \mathbb{Q}$, a function evaluation oracle for $f\colon X \to Y$ would map $|x, 0\rangle$ to $|x, f(x)\rangle$, where $|x\rangle$ and $|f(x)\rangle$ are basis states corresponding to binary representations of x and $f(x)$ respectively. Unlike classical algorithms, quantum computers can apply such an oracle to a *superposition* of different y's. They are also allowed to apply the inverse of a unitary oracle.

The standard quantum oracle described above models problems where there is a single correct answer to a query. When there are multiple good answers (for instance, different good approximations to the correct value) and the oracle is only required to give a correct answer with high probability, then we will work with the more liberal notion of *relational* quantum oracles.

Definition 12.2 (Relational quantum oracle)**.** *Let $\mathcal{F}\colon X \to \mathcal{P}(Y)$ be a function, such that for each $x \in X$ the subset $\mathcal{F}(x) \subseteq Y$ is the set of valid answers to an*

x query. A relational quantum oracle for $\mathcal{F}$ which answers queries with success probability $\geq 1-\rho$, is a unitary that for all $x \in X$ maps

$$U\colon |x,0,0\rangle \mapsto \sum_{y\in Y} \alpha_{x,y}|x,y,\psi_{x,y}\rangle,$$

where $|\psi_{x,y}\rangle$ denotes some normalized quantum state and $\sum_{y\in\mathcal{F}(x)} |\alpha_{x,y}|^2 \geq 1-\rho$. Thus measuring the second register of $U|x,0,0\rangle$ gives a valid answer to the x query with probability at least $1-\rho$.

This definition is very natural for cases where the oracle is implemented by a quantum algorithm that produces a valid answer with probability $\geq 1-\rho$.

12.2.1 Oracles for convex sets

We largely follow the seminal work [GLS88] in the different types of access to a convex set K that we consider. The main difference is that we allow each oracle an error probability: with probability at most ρ the output of the oracle may be incorrect (and we have no way to detect an incorrect answer). By choosing to allow an error probability we follow the work of Lee, Sidford, and Vempala [LSV18].

Definition 12.3 (Membership oracle $\mathrm{MEM}_{\varepsilon,\rho}(K)$)**.** *Queried with a vector $y \in \mathbb{Q}^n$, the oracle, with success probability $\geq 1-\rho$, correctly asserts one of the following*

- *$y \in B(K,\varepsilon)$, or*
- *$y \notin B(K,-\varepsilon)$.*

Definition 12.4 (Separation oracle $\mathrm{SEP}_{\varepsilon,\rho}(K)$)**.** *Queried with a vector $y \in \mathbb{Q}^n$, the oracle, with success probability at least $\geq 1-\rho$, correctly asserts one of the following*

- *$y \in B(K,\varepsilon)$, or*
- *$y \notin B(K,-\varepsilon)$,*

and in the second case it returns a unit[5] vector $g \in \mathbb{Q}^n$ such that $\langle g,x\rangle \leq \langle g,y\rangle + \varepsilon$ for all $x \in B(K,-\varepsilon)$.

Definition 12.5 (Optimization oracle $\mathrm{OPT}_{\varepsilon,\rho}(K)$)**.** *Queried with a unit vector $c \in \mathbb{Q}^n$, the oracle, with probability $\geq 1-\rho$, does one of the following:*

- *it returns a vector $y \in \mathbb{Q}^n$ such that $y \in B(K,\varepsilon)$ and $\langle c,x\rangle \leq \langle c,y\rangle + \varepsilon$ for all $x \in B(K,-\varepsilon)$,*
- *or it correctly asserts that $B(K,-\varepsilon)$ is empty.*

Note that the above optimization oracle corresponds to *maximizing* a linear function over a convex set; we could equally well state it for minimization.

[5] In [GLS88], the vector $g \in \mathbb{Q}^n$ is required to be a unit vector in the ∞-norm, to avoid having to normalize by an irrational number. We choose to work with the 2-norm, which means that 'unit' should be interpreted as 2-norm very close to 1, say $\|g\|_2 \in [0.99, 1.01]$.

Definition 12.6 (Violation oracle $\mathrm{VIOL}_{\varepsilon,\rho}(K)$)**.** *Queried with a unit vector $c \in \mathbb{Q}^n$ and a real number γ, the oracle, with probability $\geq 1-\rho$, does one of the following:*

- *it asserts that $\langle c, x\rangle \leq \gamma + \varepsilon$ for all $x \in B(K, -\varepsilon)$,*
- *or it finds a rational vector $y \in B(K, \varepsilon)$ such that $\langle c, y\rangle \geq \gamma - \varepsilon$.*

Definition 12.7 (Validity oracle $\mathrm{VAL}_{\varepsilon,\rho}(K)$)**.** *Queried with a unit vector $c \in \mathbb{Q}^n$ and a rational number γ, the oracle, with probability $\geq 1-\rho$, does one of the following:*

- *it asserts that $\langle c, x\rangle \leq \gamma + \varepsilon$ for all $x \in B(K, -\varepsilon)$,*
- *or it asserts that $\langle c, y\rangle \geq \gamma - \varepsilon$ for some $y \in B(K, \varepsilon)$.*

If in the above definitions both ε and ρ are equal to 0, then we call the oracle *strong*. If either is non-zero then we sometimes call it *weak*. If $\rho = 0$, then we recover the weak oracles defined by Grötschel, Lovász, and Schrijver in [GLS88].

When we discuss membership queries, we will always assume that we are given a small ball which lies inside the convex set. It is easy to see that without such a small ball one cannot obtain an optimization oracle using only poly(n) classical queries to a membership oracle (see, e.g., [GLS88, Sec. 4.1] or the example below). As the following example shows, the same holds for quantum queries. We will use a reduction from a version of the well-studied *search* problem:

Given $z \in \{0,1\}^N$ such that $|z| := \|z\|_1 = 1$, find $b \in [N]$ such that $z_b = 1$.

It is not hard to see that if the access to z is given via classical queries $i \mapsto z_i$, then $\Omega(N)$ queries are needed. It is well known [BBBV97] that if we allow quantum queries, i.e., applications of the unitary $|i\rangle|b\rangle \mapsto |i\rangle|z_i \oplus b\rangle$, then $\Omega(\sqrt{N})$ queries are needed. As we have seen in Chapter 9, the Grover search algorithm shows that $\mathcal{O}(\sqrt{N})$ queries are also sufficient. Now suppose that we have an algorithm that turns a membership oracle for any convex set $K \subseteq B(0, \sqrt{n})$ into an optimization oracle. We will show that this algorithm needs to make $2^{\Omega(n)}$ queries to the membership oracle. Let $N = 2^n$ and consider an input $z \in \{0,1\}^N$ to the search problem. Let $b \in \{0,1\}^n$ be the index such that $z_b = 1$. Consider maximizing the linear function $\langle e, z\rangle$ (where e is the all-1 vector) over the set $K_z = \prod_{i=1}^n [b_i - 1/2, b_i]$. Clearly the optimal solution to this convex optimization problem, even with a small constant additive error in the answer, gives the solution to the search problem. Also, a membership query is essentially equivalent to querying a bit of z. Therefore, $\Omega(\sqrt{N}) = \Omega(2^{n/2})$ quantum queries to the membership oracle are needed to implement an optimization oracle.

12.3 Computing approximate subgradients of convex Lipschitz functions

Here we show how to compute an approximate subgradient (at 0) of a convex Lipschitz function. That is, given a convex set C such that $0 \in \mathrm{int}(C)$ and a convex function $f : C \to \mathbb{R}$, we show how to compute a vector $\tilde{g} \in \mathbb{R}^n$ such that $f(y) \geq f(0) + \langle \tilde{g}, y\rangle - a\|y\| - b$ for some real numbers $a, b > 0$ that will be defined later (see Lemma 12.12 and Lemma 12.19). The idea of the classical algorithm given in the next section is to pick a point $z \in B_\infty(0, r_1)$ uniformly at random and use the finite difference $\nabla^{(r_2)} f(z)$ (defined below) as an approximate subgradient of f at 0; the radii r_1 and r_2 need to be chosen small to make the approximation good. This results in a slightly simplified version of the algorithm of Lee et al. [LSV18]. In Section 12.3.2 we show how to speed up this classical algorithm on a quantum computer.

12.3.1 Classical approach

Definition 12.8 (Finite difference gradient approximation)**.** *For a function $f : C \to \mathbb{R}$, and a point $x \in \mathbb{R}^n$ such that $B_1(x, r) \subseteq C$, and $i \in [n]$, we define $\nabla_i^{(r)} f(x) := \frac{f(x+re_i)-f(x-re_i)}{2r}$, where $e_i \in \{0,1\}^n$ is the vector that has a 1 only in its ith coordinate. Similarly we define*

$$\nabla^{(r)} f(x) := \left(\nabla_1^{(r)} f(x), \nabla_2^{(r)} f(x), \ldots, \nabla_n^{(r)} f(x)\right) \in \mathbb{R}^n.$$

For a differentiable function f we have that $\nabla^{(r)} f(x) \to \nabla f(x)$ as $r \to 0$.

Definition 12.9 (Finite difference Laplace approximation)**.** *For a function $f : C \to \mathbb{R}$, and a point $x \in \mathbb{R}^n$ such that $B_1(x, r) \subseteq C$, and $i \in [n]$, we define $\Delta_i^{(r)} f(x) := \frac{f(x+re_i)-2f(x)+f(x-re_i)}{r^2}$. Similarly*

$$\Delta^{(r)} f(x) := \sum_{i=1}^{n} \Delta_i^{(r)} f(x) \in \mathbb{R}.$$

Note that for a convex function we have $\Delta_i^{(r)} f(x) \geq 0$ for all x for which the inclusion $B_1(x, r) \subseteq C$ holds.

The next two lemmas will be needed in the proof of the main result of this section, Lemma 12.12. In Lemma 12.10 we give an upper bound on the deviation $\left\|g - \nabla^{(r_2)} f(z)\right\|_1$ of a finite difference gradient approximation $\nabla^{(r_2)} f(z)$ from an actual subgradient g at the point z, in terms of the finite difference Laplace approximation $\Delta^{(r_2)} f(z)$. Then, in Lemma 12.11 we show that in expectation, the finite difference Laplace approximation is small. Together with Markov's inequality this gives us good control over the quality of a finite difference gradient approximation.

Lemma 12.10. *If $r_2 > 0$, $z \in \mathbb{R}^n$, and $f : B_1(z, r_2) \to \mathbb{R}$ is convex, then*

$$\sup_{g \in \underline{\partial} f(z)} \left\| g - \nabla^{(r_2)} f(z) \right\|_1 \leq \frac{r_2 \Delta^{(r_2)} f(z)}{2}.$$

Proof. Fix a $g \in \underline{\partial} f(z)$. For every $i \in [n]$, we have

$$f(z + r_2 e_i) \geq f(z) + \langle g, r_2 e_i\rangle = f(z) + r_2 g_i,$$

and, similarly, $f(z - r_2e_i) \geq f(z) - r_2g_i$. Rearranging gives

$$\underbrace{\frac{f(z) - f(z - r_2e_i)}{r_2}}_{:=A} \leq g_i \leq \underbrace{\frac{f(z + r_2e_i) - f(z)}{r_2}}_{:=B}.$$

Note that $|g_i - \frac{A+B}{2}| \leq \frac{B-A}{2}$ for any three real numbers $A \leq g_i \leq B$. Moreover, $\frac{A+B}{2} = \nabla_i^{(r_2)} f(z)$ and $B - A = r_2\Delta_i^{(r_2)} f(z)$, thus $\left|g_i - \nabla_i^{(r_2)} f(z)\right| \leq \frac{r_2\Delta_i^{(r_2)} f(z)}{2}$. Now we can finish the proof by summing this inequality over all $i \in [n]$. □

Lemma 12.11. *If $0 < r_2 \leq r_1$, and $f : B_\infty(x, r_1 + r_2) \to \mathbb{R}$ is convex and L-Lipschitz, then*

$$\mathop{\mathbb{E}}_{z\in B_\infty(x,r_1)} \Delta^{(r_2)} f(z) \leq \frac{nL}{r_1},$$

where the expectation is taken with respect to the uniform distribution on $B_\infty(x, r_1)$.

Proof. Below we show that $\mathop{\mathbb{E}}_{z\in B_\infty(x,r_1)} \Delta_i^{(r_2)} f(z) \leq \frac{L}{r_1}$ for all $i \in [n]$, and then sum over i.

$$\begin{aligned}
&\mathop{\mathbb{E}}_{z\in B_\infty(x,r_1)} \Delta_i^{(r_2)} f(z) \\
&\quad = \frac{1}{(2r_1)^n} \int_{z\in B_\infty(x,r_1)} \frac{f(z + r_2e_i) - 2f(z) + f(z - r_2e_i)}{r_2^2}\, dz \\
&\quad = \frac{1}{(2r_1)^n} \int_{\substack{z_j\in[x_j-r_1,x_j+r_1],\\ j\in[n],j\neq i}} \int_{z_i\in[x_i-r_1,x_i+r_1]} \frac{f(z + r_2e_i) - 2f(z) + f(z - r_2e_i)}{r_2^2}\, dz \\
&\quad = \frac{1}{(2r_1)^n} \int_{\substack{z_j\in[x_j-r_1,x_j+r_1],\\ j\in[n],j\neq i}} \Big(\int_{z_i\in[x_i-r_1,x_i-r_1+r_2]} \frac{f(z + r_2e_i) - f(z)}{r_2^2}\, dz \\
&\qquad\qquad + \int_{z_i\in[x_i+r_1-r_2,x_i+r_1]} \frac{-f(z) + f(z - r_2e_i)}{r_2^2}\, dz \Big) \\
&\quad \leq \frac{1}{(2r_1)^n} \int_{\substack{z_j\in[x_j-r_1,x_j+r_1],\\ j\in[n],j\neq i}} 2L\, dz \quad = \frac{L}{r_1}.
\end{aligned}$$

□

Note that the above lemma is stated and proved for continuous random variables, but the same proof holds if we have a uniform hypergrid over the same hypercube, providing a discrete version of the above result. In the discrete case, in order to get the same cancellations we need to assume that both r_1 and r_2 are integer multiples of the grid spacing.

We are now ready to prove the main result of this section. Informally, the next lemma proves that an approximate subgradient of a convex Lipschitz function f at 0 can be obtained by an algorithm that outputs $\nabla^{(r_2)} \tilde{f}(z)$ for a random z close enough to 0, where $\tilde{f}$ is an approximate version of f. In other words, this lemma gives us a classical algorithm to compute an approximate subgradient of f using $2n$ classical queries to an approximate version of f.

Lemma 12.12. *Let $r_1 > 0$, $L > 0$, $\rho \in (0, 1/3]$, and $\delta \in (0, r_1\sqrt{n}L/\rho]$. Then $r_2 := \sqrt{\frac{\delta r_1 \rho}{\sqrt{n}L}} \leq r_1$. Suppose $f : C \to \mathbb{R}$ is a convex function that is L-Lipschitz on $B_\infty(0, 2r_1)$, and $\tilde{f} : B_\infty(0, 2r_1) \to \mathbb{R}$ is such that $\left\|\tilde{f} - f\right\|_\infty \leq \delta$. Then for a uniformly random $z \in B_\infty(0, r_1)$, with probability at least $1 - \rho$*

$$f(y) \geq f(0) + \left\langle \nabla^{(r_2)} \tilde{f}(z), y \right\rangle - \frac{3n^{\frac{3}{4}}}{2}\sqrt{\frac{\delta L}{\rho r_1}}\|y\| - 2L\sqrt{n}r_1 \qquad \text{for all } y \in C.$$

Proof. Let $z \in B_\infty(0, r_1)$ and $g \in \underline{\partial} f(z)$. Recall $\|g\| \leq L$ by Equation (12.2). Then for all $y \in C$

$$\begin{aligned}
f(y) &\geq f(z) + \langle g, y - z\rangle \\
&= f(z) + \langle g, y - z\rangle + \left(\left\langle \nabla^{(r_2)} f(z), y \right\rangle - \left\langle \nabla^{(r_2)} f(z), y \right\rangle\right) + (f(0) - f(0)) \\
&= f(0) + \left\langle \nabla^{(r_2)} f(z), y \right\rangle + \langle g - \nabla^{(r_2)} f(z), y\rangle + (f(z) - f(0)) + \langle g, -z\rangle \\
&\geq f(0) + \left\langle \nabla^{(r_2)} f(z), y \right\rangle - \left\|g - \nabla^{(r_2)} f(z)\right\|_1 \|y\|_\infty - L\|z\| - \|g\|\|z\| \\
&\geq f(0) + \left\langle \nabla^{(r_2)} f(z), y \right\rangle - \left\|g - \nabla^{(r_2)} f(z)\right\|_1 \|y\|_\infty - L\sqrt{n}r_1 - L\sqrt{n}r_1 \\
&\geq f(0) + \left\langle \nabla^{(r_2)} \tilde{f}(z), y \right\rangle - \frac{\delta\sqrt{n}}{r_2}\|y\| - \left\|g - \nabla^{(r_2)} f(z)\right\|_1 \|y\|_\infty - 2L\sqrt{n}r_1.
\end{aligned}$$

Note that in the last line we switched from f to $\tilde{f}$, using that $\nabla^{(r_2)} f(z)$ and $\nabla^{(r_2)} \tilde{f}(z)$ differ by at most δ/r_2 in each coordinate. Our choice of r_2 gives $\frac{\delta\sqrt{n}}{r_2} = n^{\frac{3}{4}}\sqrt{\frac{\delta L}{\rho r_1}}$ and by Lemma 12.10–12.11 we have

$$\mathop{\mathbb{E}}_{z \in B_\infty(x, r_1)} \left\|g - \nabla^{(r_2)} f(z)\right\|_1 \leq \frac{nLr_2}{2r_1} = \frac{n^{\frac{3}{4}}}{2}\sqrt{\frac{\delta L \rho}{r_1}}.$$

By Markov's inequality we get that $\left\|g - \nabla^{(r_2)} f(z)\right\|_1 \leq \frac{n^{\frac{3}{4}}}{2}\sqrt{\frac{\delta L}{\rho r_1}}$ with probability at least $1 - \rho$ over the choice of z. Plugging this bound on $\left\|g - \nabla^{(r_2)} f(z)\right\|_1$ into the above lower bound on $f(y)$ concludes the proof of the lemma. □

12.3.2 Quantum improvements

In this section we show how to improve subgradient computation of convex functions via Jordan's quantum algorithm for gradient computation [Jor05]. We use the formulation given by Gilyén et al. [GAW, Lemma 20], for which we first introduce the following definition.

Definition 12.13 (Hyper-grid)**.** *For $k \in \mathbb{N}$ we define the following discretization of the interval* $(-1/2, 1/2)$*:*

$$G_k := \left\{ \frac{j}{2^k} - \frac{1}{2} + 2^{-k-1} : j \in \{0, \ldots, 2^k - 1\} \right\} \subset (-1/2, 1/2).$$

Similarly we define the n-dimensional hyper-grid $G_k^n := G_k \times \ldots \times G_k$ (n times).

Note that an element of G_k^n can be represented using $n \times k$ (qu)bits. Basically, Jordan's algorithm just sets up a uniform superposition over all grid points, applies a "phase query" to f, and then a quantum Fourier transform over each coordinate.

Lemma 12.14. (Jordan's algorithm [GAW, Lemma 20])
Let $m \in \mathbb{N}$, $c \in \mathbb{R}$ and $g \in \mathbb{R}^n$ such that $\|g\|_\infty \leq 1/3$. If $f : G_m^n \to \mathbb{R}$ is such that

$$|f(x) - \langle g, x\rangle - c| \leq \frac{2^{-m}}{42\pi}, \tag{12.3}$$

for 99.9% of the points $x \in G_m^n$, then using a single query to a phase oracle $\mathrm{O}\colon |x\rangle \mapsto e^{2\pi i 2^m f(x)}|x\rangle$ Jordan's gradient computation algorithm outputs a vector $v \in \mathbb{R}^n$ such that:

$$\Pr\big[|v_i - g_i| > 2^{2-m}\big] \leq 1/3 \quad \textit{for every } i \in [n].$$

We now show that the above algorithm allows us to compute an approximate subgradient of a function f, even if we are only given standard oracle access to a function $\tilde{f}$ which is sufficiently close to f. In particular, we will assume we are given access to a standard unitary oracle of a function $\tilde{f}\colon G_m^n \to \mathbb{R}$ which satisfies $|\tilde{f}(x) - f(x)| \leq \delta$ for all $x \in G_m^n$. That is, we assume we are given access to a unitary U acting as

$$U : |x\rangle|0\rangle \mapsto |x\rangle|\tilde{f}(x)\rangle \tag{12.4}$$

Note that if we can classically efficiently evaluate $\tilde{f}$, then it is well known that we can construct such a unitary as a small quantum circuit (see [NC00, Sec. 1.4.1]).

The main idea is that, using one application of U, a phase gate corresponding to the output register, and another application of U^* to uncompute the function value, we can implement a phase oracle for $\tilde{f}$. Moreover, Equation (12.5) below will also hold for $\tilde{f}$, with a slightly worse right-hand side, since f is close to $\tilde{f}$. A version of the following is proven in [GAW, Theorem 21], for completeness we sketch a proof.

Corollary 12.15 (Gradient computation using approximate function evaluation)**.** *Let $\delta, B, r, c \in \mathbb{R}$, and let $\rho \in (0, 1/3]$. Let $x_0, g \in \mathbb{R}^n$ with $\|g\|_\infty \leq \frac{B}{r}$. Let $m := \lceil \log_2\big(\frac{B}{28\pi\delta}\big)\rceil$ and suppose $f : (x_0 + rG_m^n) \to \mathbb{R}$ is such that*

$$|f(x_0 + rx) - \langle g, rx\rangle - c| \leq \delta \tag{12.5}$$

for 99.9% of the points $x \in G_m^n$. Assume we have access to a standard unitary oracle U, providing $\mathcal{O}\big(\log\big(\frac{B}{\delta}\big)\big)$-bit binary approximations $\tilde{f}(z)$ such that $|\tilde{f}(z) - f(z)| \leq \delta$ for all $z \in (x_0 + rG_m^n)$. Then we can compute a vector $\tilde{g} \in \mathbb{R}^n$ such that

$$\Pr\left[\|\tilde{g} - g\|_\infty > \frac{8 \cdot 42\pi\delta}{r}\right] \leq \rho,$$

with $\mathcal{O}\Big(\log\big(\frac{n}{\rho}\big)\Big)$ queries to U and U^* and with gate complexity

$$\mathcal{O}\left(n \log\left(\frac{n}{\rho}\right)\log\left(\frac{B}{\delta}\right)\mathrm{loglog}\left(\frac{n}{\rho}\right)\mathrm{loglog}\left(\frac{B}{\delta}\right)\right).$$

Proof. As described above the corollary, we first implement a phase oracle for $\tilde{f}$ and then we apply Jordan's gradient computation algorithm (Lemma 12.14).

With a single query to U and its inverse we can implement a phase oracle O that acts as $\mathrm{O} : |x\rangle \mapsto e^{2\pi i \frac{M}{3B}\tilde{f}(x_0+rx)}|x\rangle$, where $M := \frac{3B}{84\pi\delta}$, and[6] $m := \log_2(M)$. Let $h(x) := \frac{\tilde{f}(x_0+rx)}{3B}$, then by Equation (12.5) 99.9% of the points $x \in G_m^n$ satisfy

$$\left|h(x) - \left\langle \frac{r}{3B}g, x\right\rangle - \frac{c}{3B}\right| \leq \frac{2\delta}{3B} = \frac{1}{42\pi M}.$$

Since $\left\|\frac{r}{3B}g\right\|_\infty \leq \frac{1}{3}$, by Lemma 12.14 we can compute a vector $v \in \mathbb{R}^n$ which is a coordinatewise $\frac{4}{M}$-approximator of $\frac{r}{3B}g$: for each $i \in [n]$ we have $\left|g_i - \frac{3B}{r}v_i\right| \leq \frac{12B}{rM} = \frac{8\cdot 42\pi\delta}{r}$ with probability at least $\frac{2}{3}$.

Note that the above success probability is per coordinate of g. However, repeating the whole procedure $\mathcal{O}(\log(\frac{n}{\rho}))$ times and taking the median of the resulting vectors coordinatewise gives a gradient approximator $\tilde{g}$ with the desired approximation quality with probability at least $1-\rho$. For the proof of the gate complexity we refer to [GAW, Theorem 21] where the complexity of Jordan's algorithm is analyzed in detail.[7] □

Remark 12.16. *With essentially the same approach, the above corollary of Jordan's quantum gradient computation algorithm can also be proven in the setting where our access to an approximation of f is not given by a standard quantum oracle but by a* relational *quantum oracle as in Definition 12.2, see Appendix A of [vAGGdW18] for both the definition of this type of approximation to f and a proof of this corollary.*

In terms of applications, we want to point out that if the membership oracle used in Section 12.4 comes from a deterministic algorithm, then we get a standard quantum oracle. Only when the membership oracle itself is relational (for example, when it is itself computed by a bounded-error quantum algorithm) do we need the more general setting of Appendix A of [vAGGdW18].

In order to apply the above corollary, we need to find some function which is sufficiently close to linear. Fortunately, convex Lipschitz functions can be very well approximated by linear functions over most small-enough regions. Similarly to the classical case (Lemma 12.12) we make this claim quantitative using Lemma 12.11. In order to apply the more efficient quantum gradient computation of Corollary 12.15 we also need the following two lemmas to ensure that Equation (12.5) holds.

Lemma 12.17. *Let $S \subseteq \mathbb{R}^n$ be such that $S = -S$, and let* $\mathrm{conv}(S)$ *denote the convex hull of S. If $f : \mathrm{conv}(S) \to \mathbb{R}$ is a convex function, $f(0) = 0$, and $|f(s)| \leq \delta$ for all $s \in S$, then*

$$|f(s')| \leq \delta \textit{ for all } s' \in \mathrm{conv}(S).$$

Proof. Since f is convex and $f(s) \leq \delta$ for all $s \in S$ we immediately get that $f(s') \leq \delta$ for all $s' \in \mathrm{conv}(S)$. Because $f(0) = 0$ and $S = -S$, due to convexity we get that $f(s') \geq -f(-s') \geq -\delta$. □

[6] We can assume w.l.o.g. that the upper bound B is such that M is a power of two.

[7] The correspondence with the parametrization of [GAW] is $\varepsilon \leftrightarrow \frac{8\cdot 42\pi\delta}{r}$, $M \leftrightarrow \frac{B}{r}$.

Lemma 12.18. *If $r_2 > 0$, $z \in \mathbb{R}^n$ and $f : B_1(z, r_2) \to \mathbb{R}$ is convex, then*

$$\sup_{y\in B_1(0,r_2)}\left|f(z+y) - f(z) - \left\langle y, \nabla^{(r_2)} f(z)\right\rangle\right| \le \frac{r_2^2 \Delta^{(r_2)} f(z)}{2}.$$

Proof. Let $d(y) := f(z+y) - f(z) - \langle y, \nabla^{(r_2)} f(z)\rangle$ be the difference between $f(z+y)$ and its linear approximator. Let $S := \{\pm r_2 e_i : i \in [n]\}$. It is easy to see that $d(0) = 0$, $S = -S$, and $\mathrm{conv}(S) = B_1(0, r_2)$. Also, for all $s \in S$ we have that $|d(s)| \le r_2^2 \Delta^{(r_2)} f(z)/2$:

$$\begin{aligned}
d(\pm r_2 e_i) &= f(z \pm r_2 e_i) - f(z) - \left\langle \pm r_2 e_i, \nabla^{(r_2)} f(z)\right\rangle \\
&= f(z \pm r_2 e_i) - f(z) \mp r_2 \nabla_i^{(r_2)} f(z) \\
&= f(z \pm r_2 e_i) - f(z) \mp \frac{f(z + r_2 e_i) - f(z - r_2 e_i)}{2} \\
&= \frac{f(z + r_2 e_i) - 2f(z) + f(z - r_2 e_i)}{2} \\
&= r_2^2 \Delta_i^{(r_2)} f(z)/2 \\
&\le r_2^2 \Delta^{(r_2)} f(z)/2.
\end{aligned}$$

Therefore Lemma 12.17 implies that $\sup_{y\in B_1(0,r_2)}|d(y)| \le r_2^2 \Delta^{(r_2)} f(z)/2$. □

We can now state the main result of this section, the quantum analogue of Lemma 12.12.

Lemma 12.19. *Let $r_1 > 0$, $L > 0$, $\rho \in (0, 1/3]$, and suppose $\delta \in (0, r_1 n L/\rho]$. Then $r_2 := \sqrt{\frac{\delta r_1 \rho}{nL}} \le r_1$. Suppose $f : C \to \mathbb{R}$ is a convex function that is L-Lipschitz on $B_\infty(0, 2r_1)$, and we have quantum query access[8] to $\tilde{f}$, which is a δ-approximate version of f, via a unitary U over a (fine-enough) hypergrid of $B_\infty(0, 2r_1)$. Then we can compute a $\tilde{g} \in \mathbb{R}^n$ using $\mathcal{O}(\log(n/\rho))$ queries to U, such that with probability $\ge 1 - \rho$, we have*

$$f(y) \ge f(0) + \langle \tilde{g}, y\rangle - (23n)^2\sqrt{\frac{\delta L}{\rho r_1}}\|y\| - 2L\sqrt{n} r_1 \qquad \textit{for all } y \in C.$$

Proof. The quantum algorithm works roughly as follows. It first picks a uniformly[9] random $z \in B_\infty(0, r_1)$. Then it uses Jordan's quantum algorithm to compute an approximate gradient at z by approximately evaluating f in superposition over a discrete hypergrid of $B_\infty(z, r_2/n)$. This then yields an approximate subgradient of f at 0.

[8] Using [vAGGdW18, Cor. 29] instead of Corollary 12.15 shows that a relational quantum oracle also suffices as input.

[9] A discrete quantum computer strictly speaking cannot do this, but (as noted after Lemma 12.11) a uniformly random point from a fine enough hypergrid suffices.

We now work out this rough idea. Since $B_\infty(z, r_2/n) \subseteq B_1(z, r_2)$, Lemma 12.18 implies

$$\sup_{y\in B_\infty(0,r_2/n)}\left|f(z+y)-f(z)-\left\langle y, \nabla^{(r_2)}f(z)\right\rangle\right| \leq \frac{r_2^2\Delta^{(r_2)}f(z)}{2}. \tag{12.6}$$

Also as shown by Lemma 12.11 and Markov's inequality we have

$$\Delta^{(r_2)}f(z) \leq \frac{2nL}{\rho r_1} \tag{12.7}$$

with probability $\geq 1-\rho/2$ over the choice of z. If z is such that Equation (12.7) holds, then we get

$$\sup_{y\in B_\infty(0,r_2/n)}\left|f(z+y)-f(z)-\left\langle y, \nabla^{(r_2)}f(z)\right\rangle\right| \leq \frac{nLr_2^2}{\rho r_1} = \delta.$$

Now we apply the quantum algorithm of Corollary 12.15 with $r = 2r_2/n$, $c = f(z)$, $g = \nabla^{(r_2)}f(z)$, and $B = Lr$. This uses $\mathcal{O}(\log(n/\rho))$ queries to U, and with probability $\geq 1-\rho/2$ computes an approximate gradient $\tilde{g}$ such that

$$\left\|\nabla^{(r_2)}f(z)-\tilde{g}\right\|_\infty \leq \frac{8\cdot 42\pi n}{2r_2}\cdot\delta = 4\cdot 42\cdot\pi\sqrt{\frac{\delta n^3 L}{\rho r_1}}. \tag{12.8}$$

Also, if z is such that Equation (12.7) holds, then by Lemma 12.10 we get that

$$\sup_{g\in\underline{\partial} f(z)}\left\|\nabla^{(r_2)}f(z)-g\right\|_1 \leq \frac{r_2\Delta^{(r_2)}f(z)}{2} \leq \frac{nLr_2}{\rho r_1} = \sqrt{\frac{\delta nL}{\rho r_1}},$$

and therefore by the triangle inequality and Equation (12.8) we get that

$$\begin{aligned}\sup_{g\in\underline{\partial} f(z)}\|g-\tilde{g}\|_\infty &\leq \sup_{g\in\underline{\partial} f(z)}\left\|g-\nabla^{(r_2)}f(z)\right\|_\infty + \left\|\nabla^{(r_2)}f(z)-\tilde{g}\right\|_\infty\\ &\leq \sup_{g\in\underline{\partial} f(z)}\left\|g-\nabla^{(r_2)}f(z)\right\|_1 + \left\|\nabla^{(r_2)}f(z)-\tilde{g}\right\|_\infty\\ &\leq \sqrt{\frac{\delta nL}{\rho r_1}} + 4\cdot 42\cdot\pi\sqrt{\frac{\delta n^3L}{\rho r_1}} \quad < 23^2\sqrt{\frac{\delta n^3 L}{\rho r_1}}.\end{aligned}$$

Thus with probability at least $1-\rho$, for all $y\in C$ and for all $g\in\underline{\partial} f(z)$ we have that

$$\begin{aligned}f(y) &\geq f(z)+\langle g, y-z\rangle\\ &= f(0)+\langle\tilde{g},y\rangle+\langle g-\tilde{g},y\rangle+(f(z)-f(0))+\langle g,-z\rangle\\ &\geq f(0)+\langle\tilde{g},y\rangle-|\langle g-\tilde{g},y\rangle|-L\|z\|-\|g\|\|z\|\\ &\geq f(0)+\langle\tilde{g},y\rangle-\|g-\tilde{g}\|_\infty\|y\|_1-L\sqrt{n}r_1-L\sqrt{n}r_1 \qquad \text{(by (12.2))}\\ &\geq f(0)+\langle\tilde{g},y\rangle-23^2\sqrt{\frac{\delta n^3L}{\rho r_1}}\|y\|_1-2L\sqrt{n}r_1\\ &\geq f(0)+\langle\tilde{g},y\rangle-(23n)^2\sqrt{\frac{\delta L}{\rho r_1}}\|y\|-2L\sqrt{n}r_1. \qquad \square\end{aligned}$$

12.4 Algorithms for separation using membership queries

Let $K \subseteq \mathbb{R}^n$ be a convex set such that $B(0,r) \subseteq K \subseteq B(0,R)$. Given a membership oracle[10] $\mathrm{MEM}_{\varepsilon,0}(K)$ as in Definition 12.3, we construct a separation oracle $\mathrm{SEP}_{\eta,\rho}(K)$ as in Definition 12.4. Let x be the point we want to separate from K. We first make a membership query to x itself, receiving answer $x \in B(K,\varepsilon)$ or $x \notin B(K,-\varepsilon)$. Suppose $x \notin B(K,-\varepsilon)$, then we need to find a hyperplane that approximately separates x from K. Due to the rotational symmetry of the separation problem, for ease of notation we assume that $x = -\|x\|e_n$.[11] For this x define $h : \mathbb{R}^{n-1} \to \mathbb{R} \cup \{\infty\}$ as

$$h(y) := \inf\nolimits_{(y,y_n)\in K} y_n. \tag{12.9}$$

Our h is a bit different from the one used in [LSV18], but we can show that it has many of the same properties. Since K is a convex set, h is a convex function over $\mathbb{R}^{n-1}$. As we show below, the function h is also Lipschitz (Lemma 12.20) and we can approximately compute its value using binary search with $\widetilde{\mathcal{O}}(1)$ classical queries to a membership oracle (Lemma 12.21). Furthermore, an approximate subgradient of h at 0 allows to construct a hyperplane approximately separating x from K (Lemma 12.22). Combined with the results of Section 12.3 this leads to the main results of this section, Theorems 12.23 and 12.24, which show how to efficiently construct a separation oracle using classical (resp. quantum) queries to a membership oracle.

Analogously to [LSV18, Lemma 12] we first show that our h is Lipschitz.

Lemma 12.20. *For every $\delta \in (0,r)$, h is $\frac{R}{r-\delta}$-Lipschitz on $B(0,\delta) \subseteq \mathbb{R}^{n-1}$, that is, we have*

$$|h(y') - h(y)| \leq \frac{R}{r-\delta}\|y'-y\| \quad \textit{for all } y, y' \in B(0,\delta).$$

Proof. Observe that for all $y \in B(0,r)$ we have $-R \leq h(y) \leq 0$, because of the inclusions $B(0,r) \subseteq K \subseteq B(0,R)$. To show that h is Lipschitz, let $y, y' \in B(0,\delta)$ be arbitrary, and let $z = \frac{y'-y}{\|y'-y\|}$. Observe that

$$y + (\|y'-y\| + (r-\delta))z = y' + (r-\delta)z \in B(0,r),$$

[10] For simplicity we assume throughout this section that the membership oracle succeeds with certainty (i.e., its error probability is 0). This is easy to justify: suppose we have a classical T-query algorithm, which uses $\mathrm{MEM}_{\varepsilon,0}(K)$ queries and succeeds with probability at least $1-\rho$. If we are given access to a $\mathrm{MEM}_{\varepsilon,\frac{1}{3}}(K)$ oracle instead, then we can create a $\mathrm{MEM}_{\varepsilon,\frac{\rho}{T}}(K)$ oracle by $\mathcal{O}(\log(T/\rho))$ queries to $\mathrm{MEM}_{\varepsilon,\frac{1}{3}}(K)$ and taking the majority of the answers. Then running the original algorithm with $\mathrm{MEM}_{\varepsilon,\frac{\rho}{T}}(K)$ will fail with probability at most 2ρ. Therefore the assumption of a membership oracle with error probability 0 can be removed at the expense of only a small logarithmic overhead in the number of queries. A similar argument works for the quantum case.

[11] For the query complexity this is without loss of generality, since we can always apply a rotation to all the points such that this holds. If we instead consider the computational cost of our algorithm, then we have to take into account the cost of this rotation and its inverse. Note, however, that this rotation can always be written as the product of n rotations on only 2 coordinates, and hence can be applied in $\widetilde{\mathcal{O}}(n)$ additional steps.

and that

$$y' = \frac{\|y'-y\|}{\|y'-y\| + (r-\delta)}(y' + (r-\delta)z) + \frac{r-\delta}{\|y'-y\| + (r-\delta)}y.$$

Therefore, due to the convexity of h, we have

$$h(y') - h(y) \leq [h(y' + (r-\delta)z) - h(y)]\frac{\|y'-y\|}{\|y'-y\| + (r-\delta)} \leq \frac{R}{r-\delta}\|y'-y\|. \quad \square$$

Now we show how to compute the value of h using membership queries to K.

Lemma 12.21. *For all $y \in B\big(0, \frac{r}{2}\big) \subset \mathbb{R}^{n-1}$ we can compute a δ-approximation of $h(y)$ with $\mathcal{O}\big(\log\big(\frac{R}{\delta}\big)\big)$ queries to a $\mathrm{MEM}_{\varepsilon,0}(K)$ oracle, where $\varepsilon \leq \frac{r}{3R}\delta$.*

Proof. Let $y \in B(0, \frac{r}{2})$, then $(y, h(y))$ is a boundary point of K by the definition of h. Note that $h(y) \in [-R, -r/2]$, our goal is to perform binary search over this interval to find a good approximation of $h(y)$. Suppose $y_n \leq -\frac{r}{2}$ is our current guess for $h(y)$. We first show that

(a) if $(y, y_n) \in B(K, \varepsilon)$, then $y_n \geq h(y) - \delta$, and

(b) if $(y, y_n) \notin B(K, -\varepsilon)$, then $y_n \leq h(y) + \frac{2}{3}\delta$.

For the proof of (a) consider a $g \in \underline{\partial} h(y)$. Since g is a subgradient we have that $h(z) \geq h(y) + \langle g, z - y\rangle$ for all $z \in \mathbb{R}^{n-1}$. Hence, for all $z \in \mathbb{R}^{n-1}$ and z_n such that $(z, z_n) \in K$ we have

$$\left\langle \begin{pmatrix} -g \\ 1 \end{pmatrix}, \begin{pmatrix} y \\ h(y) \end{pmatrix} \right\rangle \leq \left\langle \begin{pmatrix} -g \\ 1 \end{pmatrix}, \begin{pmatrix} z \\ h(z) \end{pmatrix} \right\rangle \leq \left\langle \begin{pmatrix} -g \\ 1 \end{pmatrix}, \begin{pmatrix} z \\ z_n \end{pmatrix} \right\rangle$$

where the first inequality is a rewriting of the subgradient inequality and the second inequality uses that $z_n \geq h(z)$ since $(z, z_n) \in K$. Since $(y, y_n) \in B(K, \varepsilon)$ it follows from the above inequality that

$$\left\langle \begin{pmatrix} -g \\ 1 \end{pmatrix}, \begin{pmatrix} y \\ y_n \end{pmatrix} \right\rangle \geq \left\langle \begin{pmatrix} -g \\ 1 \end{pmatrix}, \begin{pmatrix} y \\ h(y) \end{pmatrix} \right\rangle - \varepsilon \left\| \begin{pmatrix} -g \\ 1 \end{pmatrix} \right\| \geq \left\langle \begin{pmatrix} -g \\ 1 \end{pmatrix}, \begin{pmatrix} y \\ h(y) \end{pmatrix} \right\rangle - \varepsilon(\|g\| + 1).$$

Lemma 12.20 together with the argument of Equation (12.2) implies that $\|g\| \leq \frac{2R}{r}$. Since

$$\varepsilon(\|g\| + 1) \leq \varepsilon\left(\frac{2R}{r} + 1\right) \leq \varepsilon\frac{3R}{r} \leq \delta,$$

we obtain the inequality of (a).

For (b), consider the convex set C which is the convex hull of $B((y,0), r/2)$ and $(y, h(y))$. Note that $B(C, -\varepsilon)$ is the convex hull of $B((y,0), r/2 - \varepsilon)$ and $\big(y, h(y)\big(1 - \frac{2\varepsilon}{r}\big)\big)$. Since $C \subseteq K$, we have $B(C, -\varepsilon) \subseteq B(K, -\varepsilon)$. Therefore $(y, y_n) \notin B(K, -\varepsilon)$ implies $(y, y_n) \notin B(C, -\varepsilon)$, and

$$y_n \leq h(y)\left(1 - \frac{2\varepsilon}{r}\right) = h(y) - \varepsilon\frac{2h(y)}{r} \leq h(y) + \varepsilon\frac{2R}{r} \leq h(y) + \frac{2}{3}\delta.$$

Now we can analyze the binary search algorithm. By making $\mathcal{O}\big(\log\big(\frac{R}{\delta}\big)\big)$ queries to $\mathrm{MEM}_{\varepsilon,0}(K)$ with points of the form (y,z), we can find a value $y_n \in [-R, -\frac{r}{2}]$ such that $(y, y_n) \in B(K,\varepsilon)$ but $(y, y_n - \frac{\delta}{3}) \notin B(K,-\varepsilon)$. By (a)-(b) we get that $|h(y) - y_n| \leq \delta$. □

The following lemma shows how to convert an approximate subgradient of h to a hyperplane that approximately separates x from K.

Lemma 12.22. *Suppose $-\|x\|e_n = x \notin B(K,-\varepsilon)$, and $\tilde{g} \in \mathbb{R}^{n-1}$ is an approximate subgradient of the function h of Equation* (12.9) *at* 0*, meaning that for some $a, b \in \mathbb{R}$ and for all $y \in \mathbb{R}^{n-1}$*

$$h(y) \geq h(0) + \langle \tilde{g}, y\rangle - a\|y\| - b,$$

then $s := \frac{(-\tilde{g},1)}{\|(-\tilde{g},1)\|}$ satisfies $\langle s, z\rangle \geq \langle s, x\rangle - \frac{aR+b}{\|(-\tilde{g},1)\|} - \frac{2R}{r}\frac{\varepsilon}{\|(-\tilde{g},1)\|}$ for all $z \in K$.

Proof. Let us introduce the notation $z = (y, z_n)$ and $s' := (-\tilde{g}, 1) = \|(-\tilde{g},1)\|s$, then

$$\begin{aligned}
\langle s', z\rangle &= z_n - \langle \tilde{g}, y\rangle \\
&\geq h(y) - \langle \tilde{g}, y\rangle \\
&\geq h(0) - a\|y\| - b \\
&\geq -\|x\| - \frac{2R}{r}\varepsilon - aR - b \\
&= \langle s', x\rangle - aR - b - \frac{2R}{r}\varepsilon,
\end{aligned}$$

where the last inequality used claim (b) from the proof of Lemma 12.21. □

We now construct a separation oracle using $\widetilde{\mathcal{O}}(n)$ classical queries to a membership oracle. In particular, for an η-precise separation oracle, we require an ε-precise membership oracle with

$$\varepsilon = \frac{\eta}{676} n^{-2} \Big(\frac{r}{R}\Big)^3 \Big(\frac{\eta}{R}\Big)^2 \rho$$

The analogous result in [LSV18, Theorem 14] uses the stronger assumption[12]

$$\varepsilon \approx \frac{\eta}{8 \cdot 10^6} n^{-\frac{7}{2}} \Big(\frac{r}{R}\Big)^6 \Big(\frac{\eta}{R}\Big)^2 \rho^3.$$

Compared to this, our result scales better in terms of n, $\frac{r}{R}$ and ρ.

Theorem 12.23. *Let K be a convex set satisfying $B(0,r) \subseteq K \subseteq B(0,R)$. For any $\eta \in (0,R]$ and $\rho \in (0,1/3]$, we can implement the oracle $\mathrm{SEP}_{\eta,\rho}(K)$ using $\mathcal{O}\Big(n\log\Big(\frac{n}{\rho}\frac{R}{\eta}\frac{R}{r}\Big)\Big)$ classical queries to a $\mathrm{MEM}_{\varepsilon,0}(K)$ oracle, assuming that ε is at most $\eta(26n)^{-2}\big(\frac{r}{R}\big)^3\big(\frac{\eta}{R}\big)^2\rho$.*

[12] It seems that Lee et al. [LSV18, Algorithm 1] did not take into account the change in precision analogous to our Lemma 12.21, therefore one would probably need to worsen their exponent of $\frac{r}{R}$ from 6 to 7.

Proof. Let $x \notin B(K, -\varepsilon)$ be the point we want to separate from K. Let $\delta := \eta\frac{n^{-2}}{9\cdot 24}\left(\frac{r}{R}\cdot\frac{\eta}{R}\right)^2\rho$,
then $\varepsilon \leq \frac{r}{3R}\delta$. By Lemma 12.21 we can evaluate h to within error δ using $\mathcal{O}\left(\log\left(\frac{R}{\delta}\right)\right)$ queries to a $\mathrm{MEM}_{\varepsilon,0}(K)$ oracle. By Lemma 12.20 we know that h is $\frac{2R}{r}$-Lipschitz on $B(0, r/2)$. Let us choose $r_1 := \frac{r}{12\sqrt{n}}\frac{\eta}{R}$, then $r_1\sqrt{n} \leq \frac{r}{4}$ and therefore we have the inclusion $B_\infty(0, 2r_1) \subseteq B(0, r/2)$. Also note that $\delta \leq \frac{\eta}{6\rho} = \frac{2r_1\sqrt{n}R}{\rho r}$. Hence by Lemma 12.12, using $\mathcal{O}\left(n\log\left(\frac{R}{\delta}\right)\right)$ queries to a $\mathrm{MEM}_{\varepsilon,0}(K)$ oracle, we can compute an approximate subgradient $\tilde{g}$ such that with probability at least $1 - \rho$ we have

$$h(y) \geq h(0) + \langle\tilde{g}, y\rangle - \frac{3n^{\frac{3}{4}}}{2}\sqrt{\frac{\delta 2R}{\rho r_1 r}}\|y\| - \frac{4R}{r}\sqrt{n}r_1 \qquad \text{for all } y \in \mathbb{R}^{n-1}.$$

Substituting the value of r_1 and δ we get $h(y) \geq h(0) + \langle\tilde{g}, y\rangle - \frac{\eta}{2R}\|y\| - \frac{\eta}{3}$, which by Lemma 12.22 gives an s such that $\langle s, z\rangle \geq \langle s, x\rangle - \frac{5}{6}\eta - \frac{2R}{r}\varepsilon \geq \langle s, x\rangle - \eta$ for all $z \in K$ □

Finally, we give a proof of our main result: we construct a separation oracle using $\widetilde{\mathcal{O}}(1)$ quantum queries to a membership oracle.

Theorem 12.24. *Let K be a convex set satisfying $B(0, r) \subseteq K \subseteq B(0, R)$. For any $\eta \in (0, R]$ and $\rho \in (0, 1/3]$, we can implement the oracle $\mathrm{SEP}_{\eta,\rho}(K)$ using $\mathcal{O}\left(\log\left(\frac{n}{\rho}\right)\log\left(\frac{n}{\rho}\frac{R}{\eta}\frac{R}{r}\right)\right)$ quantum queries to a $\mathrm{MEM}_{\varepsilon,0}(K)$ oracle, where $\varepsilon \leq \eta(58n)^{-\frac{9}{2}}\left(\frac{r}{R}\right)^3\left(\frac{\eta}{R}\right)^2\rho$.*

Proof. Let $x \notin B(K, -\varepsilon)$ be the point we want to separate from the convex set K. Let $\delta := \eta\frac{23^{-4}}{4\cdot 24}n^{-\frac{9}{2}}\left(\frac{r}{R}\cdot\frac{\eta}{R}\right)^2\rho$, then $\varepsilon \leq \frac{r}{3R}\delta$. By Lemma 12.21 we can evaluate h to within error δ using $\mathcal{O}\left(\log\left(\frac{R}{\delta}\right)\right)$ queries to a $\mathrm{MEM}_{\varepsilon,0}(K)$ oracle. By Lemma 12.20 we know that h is $\frac{2R}{r}$-Lipschitz on $B(0, r/2)$. Let us choose $r_1 := \frac{r}{12\sqrt{n}}\frac{\eta}{R}$, then $r_1\sqrt{n} \leq \frac{r}{4}$, and therefore $B_\infty(0, 2r_1) \subseteq B(0, r/2)$. Also note that $\delta \leq \frac{\eta}{6\rho} = \frac{2r_1 nR}{\rho r}$. Hence by Lemma 12.19, using $\mathcal{O}\left(\log\left(\frac{n}{\rho}\right)\log\left(\frac{R}{\delta}\right)\right)$ queries to a $\mathrm{MEM}_{\varepsilon,0}(K)$ oracle, we can compute an approximate subgradient $\tilde{g}$ such that with probability at least $1 - \rho$ we have

$$h(y) \geq h(0) + \langle\tilde{g}, y\rangle - (23n)^2\sqrt{\frac{2\delta R}{\rho r_1 r}}\|y\| - \frac{4R}{r}\sqrt{n}r_1 \qquad \text{for all } y \in \mathbb{R}^{n-1}.$$

Substituting the value of r_1 and δ we get $h(y) \geq h(0) + \langle\tilde{g}, y\rangle - \frac{\eta}{2R}\|y\| - \frac{\eta}{3}$, which by Lemma 12.22 gives an s such that $\langle s, z\rangle \geq \langle s, x\rangle - \frac{5}{6}\eta - \frac{2R}{r}\varepsilon \geq \langle s, x\rangle - \eta$ for all $z \in K$. □

12.5 Lower bounds

For a convex set K satisfying $B(0, r) \subseteq K \subseteq B(0, R)$, we have shown in Theorem 12.24 that one can implement a $\mathrm{SEP}(K)$ oracle with $\widetilde{\mathcal{O}}(1)$ quantum queries to a MEM(K) oracle if the membership oracle is sufficiently precise. In this section we first show that this is exponentially better than what can be achieved using classical access to a membership oracle. We also investigate how many queries to a membership/separation oracle are needed in order to implement an optimization oracle. Our results are as follows.

- We show that $\Omega(n)$ classical queries to a membership oracle are needed to implement a separation oracle.
- We show that $\Omega(n)$ classical (resp. $\Omega(\sqrt{n})$ quantum) queries to a separation oracle are needed to implement an optimization oracle; even when we *know an interior point* in the set.
- We show an $\Omega(n)$ lower bound on the number of classical and/or quantum queries to a separation oracle needed to optimize over the set when we *do not know an interior point.*

In this section we will always assume that the input oracle is a strong oracle but the output oracle is allowed to be a weak oracle with error ε. Furthermore, we will make sure that R, $1/r$, and $1/\varepsilon$ are all upper bounded by a polynomial in n. This guarantees that the lower bound is based on the dimension of the problem, not the required precision.

12.5.1 Classical lower bound on the number of MEM queries needed for SEP

Here we show that a separation query can provide $\Omega(n)$ bits of information about the underlying convex set K; since a classical membership query returns a 0 or a 1 and hence can give at most 1 bit of information[13], this theorem immediately implies a lower bound of $\Omega(n)$ on the number of classical membership queries needed to implement one separation query.

Theorem 12.25. *Let $\varepsilon \leq \frac{39}{1600}$. There exist a set of $m = 2^{\Omega(n)}$ convex sets $K_1, \ldots, K_m$ and points $y, x_0 \in \mathbb{R}^n$ such that $B(x_0, 1/3) \subseteq K_i \subseteq B(x_0, 2\sqrt{n})$ for all $i \in [m]$, and such that the result of a classical query to* $\mathrm{SEP}_{\varepsilon,0}(K_i)$ *with the point y correctly identifies i.*

Proof. Let $h_1, \ldots, h_m \in \mathbb{R}^n$ be a set of $m = 2^{\Omega(n)}$ entrywise non-negative unit vectors such that $\langle h_i, h_j \rangle \leq 0.51$ for all distinct $i, j \in [m]$. Such a set of m vectors can for instance be constructed from a good error-correcting code that encodes $\Omega(n)$-bit words into n-bit codewords with pairwise Hamming distance close to $n/2$.

Now pick an $i \in [m]$ and define $\hat{K}_i := \{x : \langle h_i, x \rangle \leq 0\} \cap B(0, \sqrt{n})$ and $K_i := B(\hat{K}_i, \varepsilon)$. Then $\hat{K}_i = B(K_i, -\varepsilon)$. We claim that a query to $\mathrm{SEP}_{\varepsilon,0}(K_i)$ with the point $y = 3\varepsilon e \in \mathbb{R}^n$ will identify h_i. First note that $y \notin B(K_i, \varepsilon)$, since $\hat{K}_i$ does not contain any entrywise positive vectors and y has distance at least 3ε from all

[13] This is not true for *quantum* membership queries!

vectors that have at least one non-positive entry. Hence a separation query with y will return a unit vector g such that for all $x \in \hat{K}_i$

$$\langle g, x\rangle \leq \langle g, y\rangle + \varepsilon \leq \|g\| \cdot \|y\| + \varepsilon \leq (3\sqrt{n}+1)\varepsilon \leq 4\sqrt{n}\varepsilon. \tag{12.10}$$

Now consider the specific point x that is the projection of g onto $h_i^{\perp}$ (the hyperplane orthogonal to h_i) scaled by a factor $\sqrt{n}$, i.e., $x = \sqrt{n}(g - \langle g, h_i\rangle h_i)$. Since $\langle h_i, x\rangle = 0$ and $\|x\| \leq \sqrt{n}$, we have $x \in \hat{K}_i$. Therefore (12.10) gives the following inequality

$$\sqrt{n}(1 - \langle g, h_i\rangle^2) = \langle g, x\rangle \leq 4\sqrt{n}\varepsilon.$$

Hence $|\langle g, h_i\rangle| \geq \sqrt{1-4\varepsilon} \geq \frac{19}{20}$. This implies that $g - h_i$ or $g + h_i$ has length at most $\sqrt{2 - 2|\langle g, h_i\rangle|} \leq \sqrt{\frac{1}{10}}$; assume the former for simplicity. Now for all $j \neq i$ we have

$$|\langle g, h_j\rangle| \leq |\langle g - h_i, h_j\rangle| + |\langle h_i, h_j\rangle| \leq \sqrt{\frac{1}{10}} + 0.51 < \frac{9}{10}.$$

Hence g uniquely identifies h_i. Finally, for $x_0 = -e/3$ we have the inclusions $B(x_0, 1/3) \subseteq K_i \subseteq B(x_0, 2\sqrt{n})$. □

12.5.2 Lower bound on number of SEP queries for OPT (given an interior point)

We now consider lower bounding the number of quantum queries to a separation oracle needed to do optimization. In fact, we prove a lower bound on the number of separation queries needed for validity, which implies the same bound on optimization. We will use a reduction from a version of the well-studied *search* problem:

Given $z \in \{0,1\}^n$ such that either $|z| = 0$ or $|z| = 1$, decide which of the two holds.

This is a slightly different version from the one used in Section 12.2.1, but again $\Theta(n)$ classical queries and $\Theta(\sqrt{n})$ quantum queries are necessary and sufficient. We use this problem to show that there exist convex sets for which it is hard to construct a weak validity oracle, given a strong separation oracle. Since a separation oracle can be used as a membership oracle, this gives the same hardness result for constructing a weak validity oracle from a strong membership oracle.

Theorem 12.26. *Let $0 < \rho \leq 1/3$. Let $\mathcal{A}$ be an algorithm that can implement a $\mathrm{VAL}_{(4n)^{-1},\rho}(K)$ oracle for every convex set K (with $B(x_0, r) \subseteq K \subseteq B(x_0, R)$) using only queries to a $\mathrm{SEP}_{0,0}(K)$ oracle, and unitaries that are independent of K. Then the following statements are true, even when we restrict to convex sets K with $r = 1/3$ and $R = 2\sqrt{n}$:*

- *if the queries to $\mathrm{SEP}_{0,0}(K)$ are classical, then the algorithm uses $\Omega(n)$ queries.*
- *if the queries to $\mathrm{SEP}_{0,0}(K)$ are quantum, then the algorithm uses $\Omega(\sqrt{n})$ queries.*

Proof. Let $z \in \{0,1\}^n$ have Hamming weight $|z| = 0$ or $|z| = 1$. We construct a set K_z in such a way that solving the weak validity problem solves the search problem for z, while separation queries for K_z can be answered using a single query to z. The known classical and quantum lower bounds on the search problem then imply the two claims of the theorem, respectively.

Define $K_z := \times_{i=1}^{n}[-1, z_i]$. We first show how to implement a strong separation oracle using a single query to z. Suppose the input is the point y. The strong separation oracle works as follows:

1. If $y \in [-1,0]^n$, then return the statement that $y \in B(K_z, 0) = K_z$.
2. If $y \notin [-1,1]^n$, then return a hyperplane that separates y from $[-1,1]^n$ (and hence from K_z).
3. Let i be such that $y_i > 0$. Query z_i.
   (a) If $z_i = 1$ and i is the only index such that $y_i > 0$, then return that $y \in B(K_z, 0) = K_z$.
   (b) If $z_i = 1$ and there is a $j \neq i$ such that $y_j > 0$, return separating hyperplane $x_j \leq y_j$.
   (c) If $z_i = 0$, then return the separating hyperplane $x_i \leq y_i$.

It remains to show that a query to a weak validity oracle with accuracy $\varepsilon = \frac{1}{4n}$ can solve the search problem on z. We show that a validity query over K_z with the direction $c = \frac{1}{\sqrt{n}}(1, \dots, 1) \in \mathbb{R}^n$ and value $\gamma = \frac{1}{2\sqrt{n}}$ solves the search problem:

- If $|z| = 0$, then we claim validity will return that $\langle c, x\rangle \leq \gamma + \varepsilon$ holds for all $x \in B(K_0, -\varepsilon)$.

  Indeed, we show there is no $x \in B(K_0, \varepsilon)$ with $\langle c, x\rangle \geq \gamma - \varepsilon$. For all points $x \in K_0$ we have $\langle c, x\rangle \leq 0$. Thus, for all points $x \in B(K_0, \varepsilon)$ we have $\langle c, x\rangle \leq \varepsilon < \gamma - \varepsilon$.

- If $|z| = 1$, then we claim validity will return that $\langle c, x\rangle \geq \gamma - \varepsilon$ holds for some $x \in B(K_z, \varepsilon)$.

  Indeed, we show there is an $x \in B(K_z, -\varepsilon)$ for which $\langle c, x\rangle > \gamma + \varepsilon$. The point $z \in K_z$ satisfies $\langle z, c\rangle = \frac{1}{\sqrt{n}}$ and therefore $x = z - \varepsilon e \in B(K_z, -\varepsilon)$ satisfies $\langle c, x\rangle = \frac{1}{\sqrt{n}} - \sqrt{n}\varepsilon > \gamma + \varepsilon$.

Finally, we observe that if we set $x_0 = (-1/2, \dots, -1/2)$, then we have the inclusions $B(x_0, \frac{1}{3}) \subseteq K_z \subseteq B(x_0, 2\sqrt{n})$. □

12.5.3 Lower bound on number of SEP queries for OPT (without interior point)

We now lower bound the number of quantum queries to a separation oracle needed to solve the optimization problem, if our algorithm does not already know an interior point of K. In fact we prove a lower bound on finding a point in K using separation queries, which implies the lower bound on the number of separation queries needed for optimization.

We prove our lower bound by a reduction to the problem of learning z with *first-difference queries*. Here one needs to find an initially unknown n-bit binary string z via a guessing game. For a given guess $g \in \{0,1\}^n$ a query returns the first index in $[n]$ for which the binary strings z and g differ (or it returns $n+1$ if $z = g$). The goal is to recover z with as few guesses as possible. First we prove an $\Omega(n)$ quantum query lower bound for this problem.[14]

Theorem 12.27 (Quantum lower bound for learning z with first-difference queries). *Let $z \in \{0,1\}^n$ be an unknown string accessible by an oracle acting as $O_z|g,b\rangle = |g, b \oplus f(g,z)\rangle$, where $f(g,z)$ is the first index for which z and g differ, more precisely*

$$f(g,z) = \begin{cases} \min\{i \in [n] : g_i \neq z_i\} & \text{if } g \neq z \\ f(g,z) = n+1 & \text{otherwise.} \end{cases}$$

Then every quantum algorithm that outputs z with high probability uses at least $\Omega(n)$ queries to O_z.

Proof. We will use the general adversary bound [HLŠ07]. For this problem, we call $\Gamma \in \mathbb{R}^{2^n \times 2^n}$ an *adversary matrix* if it is a non-zero matrix with zero diagonal whose rows and columns are indexed by all $z \in \{0,1\}^n$. For $g \in \{0,1\}^n$ let us define $\Delta_g \in \{0,1\}^{2^n \times 2^n}$ such that the $[z,z']$ entry of Δ_g is 0 if and only if $f(g,z) = f(g,z')$. The general adversary bound tells us that for any adversary matrix Γ, the quantum query complexity of our problem is

$$\Omega\left(\frac{\|\Gamma\|}{\max_{g \in \{0,1\}^n} \|\Gamma \circ \Delta_g\|}\right), \tag{12.11}$$

where "$\circ$" denotes the Hadamard product and $\|\cdot\|$ the operator norm.

We claim that Equation (12.11) gives a lower bound of $\Omega(n)$ for the adversary matrix Γ defined as

$$\Gamma[z,z'] = \begin{cases} 2^{f(z,z')} & \text{if } z \neq z' \\ 0 & \text{if } z = z' \end{cases}$$

It is easy to see that Γ is indeed an adversary matrix since it is zero on the diagonal and non-zero everywhere else. Furthermore, the all-one vector e is an eigenvector of Γ with eigenvalue $n2^n$:

$$(\Gamma e)_z = \sum_{z' \in \{0,1\}^n} \Gamma[z,z'] = \sum_{d=1}^{n} 2^d \cdot |\{z' \in \{0,1\}^n : f(z,z') = d\}| = \sum_{d=1}^{n} 2^d 2^{n-d} = n2^n.$$

So $\Gamma e = n2^n e$ and hence $\|\Gamma\| \geq n2^n$.

[14] Note that this is a strengthening of the $\Omega(n)$ quantum query lower bound for binary search on a space of size 2^n by Ambainis [Amb99], since first-difference queries are at least as strong as the queries one makes in binary search.

From the definition of Δ_g it follows that

$$(\Gamma \circ \Delta_g)[z, z'] = 2^{f(z,z')} \chi_{[f(g,z) \neq f(g,z')]},$$

where $\chi_{[f(g,z) \neq f(g,z')]}$ stands for the indicator function of the condition $f(g,z) \neq f(g,z')$. Let $\Gamma_g := \Gamma \circ \Delta_g$. We will show an upper bound on $\|\Gamma_g\|$. We decompose Γ_g in an "upper-triangular" and a "lower-triangular" part:

$$\begin{aligned} \Gamma_g^U[z, z'] &:= 2^{f(z,z')} \chi_{[f(g,z) < f(g,z')]} = 2^{f(g,z)} \chi_{[f(g,z) < f(g,z')]}, \\ \Gamma_g^L[z, z'] &:= 2^{f(z,z')} \chi_{[f(g,z') < f(g,z)]} = 2^{f(g,z')} \chi_{[f(g,z') < f(g,z)]}. \end{aligned} \tag{12.12}$$

So $\Gamma_g = \Gamma_g^U + \Gamma_g^L$ and $\Gamma_g^U = (\Gamma_g^L)^T$. Hence by the triangle inequality we have

$$\|\Gamma_g\| \leq \left\|\Gamma_g^U\right\| + \left\|\Gamma_g^L\right\| = 2\left\|\Gamma_g^U\right\|. \tag{12.13}$$

It thus suffices to upper bound $\left\|\Gamma_g^U\right\|$. Notice that as (12.12) shows, $\Gamma_g^U[z, z']$ only depends on the values $f(g,z)$, $f(g,z')$. Since the range of $f(g, \cdot\,)$ is $[n+1]$, we can think of Γ_g^U as an $(n+1) \times (n+1)$ block-matrix, where the blocks are determined by the values of $f(g,z)$ and $f(g,z')$, and within a block all matrix elements are the same. Also observe that for all $k \in [n]$ there are 2^{n-k} bitstrings $y \in \{0,1\}^n$ such that $f(g,y) = k$, which tells us the sizes of the blocks. Motivated by these observations we define an orthonormal set of vectors in $\mathbb{R}^{2^n}$ by $v_{n+1} := e_g$, and for all $k \in [n]$

$$v_k := \sum_{y: f(g,y)=k} \frac{e_y}{\sqrt{2^{n-k}}}.$$

Since the row and column spaces of Γ_g^U are spanned by $\{v_k : k \in [n+1]\}$, we can reduce Γ_g^U to a $(n+1) \times (n+1)$-dimensional matrix G:

$$\begin{aligned} \Gamma_g^U &= \left(\sum_{k=1}^{n+1} v_k v_k^T\right) \Gamma_g^U \left(\sum_{\ell=1}^{n+1} v_\ell v_\ell^T\right) \\ &= \left(\sum_{k=1}^{n+1} v_k e_k^T\right) \underbrace{\left(\sum_{k=1}^{n+1} e_k v_k^T\right) \Gamma_g^U \left(\sum_{\ell=1}^{n+1} v_\ell e_\ell^T\right)}_{G:=} \left(\sum_{\ell=1}^{n+1} e_\ell v_\ell^T\right). \end{aligned}$$

It follows from the above identity, together with the orthonormality of the vectors $\{v_1, \ldots, v_n, v_{n+1}\}$, that

$$\left\|\Gamma_g^U\right\| = \left\|\left(\sum_{k=1}^{n+1} e_k v_k^T\right) \Gamma_g^U \left(\sum_{\ell=1}^{n+1} v_\ell e_\ell^T\right)\right\| = \|G\|. \tag{12.14}$$

The matrix $G \in \mathbb{R}^{(n+1)\times(n+1)}$ is strictly upper-triangular, with the following entries for $k, \ell \in [n]$:

$$\begin{aligned}
G[k,\ell] &= v_k^T \Gamma_g^U v_\ell \\
&= \left(\sum_{z:f(g,z)=k} \frac{e_z^T}{\sqrt{2^{n-k}}}\right)\Gamma_g^U\left(\sum_{z':f(g,z')=\ell} \frac{e_{z'}}{\sqrt{2^{n-\ell}}}\right) \\
&= \frac{2^{\frac{k+\ell}{2}}}{2^n}\left(\sum_{z:f(g,z)=k} e_z^T\right)\Gamma_g^U\left(\sum_{z':f(g,z')=\ell} e_{z'}\right) \\
&= \frac{2^{\frac{k+\ell}{2}}}{2^n} \sum_{z:f(g,z)=k}\ \sum_{z':f(g,z')=\ell} \Gamma_g^U[z,z'] \\
&= \frac{2^{\frac{k+\ell}{2}}}{2^n} \sum_{z:f(g,z)=k}\ \sum_{z':f(g,z')=\ell} 2^k \chi_{[k<\ell]} && \text{(by (12.12))} \\
&= \frac{2^{\frac{k+\ell}{2}}}{2^n} 2^{n-k} 2^{n-\ell} 2^k \chi_{[k<\ell]} \\
&= 2^{n-\frac{\ell-k}{2}} \chi_{[k<\ell]}.
\end{aligned}$$

Similarly for $\ell = n+1$ we get that $G[k,\ell] = \sqrt{2}\, 2^{n-\frac{\ell-k}{2}} \chi_{[k<\ell]}$ for all $k \in [n+1]$. For each $d \in [n]$ define $G_d \in \mathbb{R}^{(n+1)\times(n+1)}$ such that $G_d[k,\ell] = G[k,\ell]\chi_{[d=\ell-k]}$. This G_d is only non-zero on a non-main diagonal (namely the (k,ℓ)-entries where $d = \ell - k$), and its non-zero entries are all upper bounded by $\sqrt{2}\, 2^n 2^{-\frac{d}{2}}$. We have $G = \sum_{d=1}^n G_d$ and therefore

$$\|G\| \le \sum_{d=1}^{n} \|G_d\| = \sum_{d=1}^{n} \sqrt{2}\, 2^n 2^{-\frac{d}{2}} = 2^n \sum_{d=0}^{n-1} (\sqrt{2})^{-d} \le \frac{2^n}{1 - 1/\sqrt{2}} \le 2^{n+2}. \tag{12.15}$$

Inequalities (12.13)-(12.15) give that $\|\Gamma_g\| \le 2^{n+3}$ and hence (12.11) yields a lower bound of $\Omega\left(\frac{n2^n}{2^{n+3}}\right) = \Omega(n)$ on the number of quantum queries to O_z needed to learn z. □

Theorem 12.28. *Finding a point in $B_\infty(K, 1/7)$ for an unknown convex set K such that $K \subseteq B_\infty(0,2) \subseteq \mathbb{R}^n$ requires $\Omega(n)$ quantum queries to a separation oracle* $\mathrm{SEP}_{0,0}(K)$*, even if we are promised there exists some unknown $x \in \mathbb{R}^n$ such that $B_\infty(x, 1/3) \subseteq K$.*

Proof. We will prove an $\Omega(n)$ quantum query lower bound for this problem by a reduction from learning with first-difference queries. Let $z \in \{0,1\}^n$ be an unknown binary string, and let us define $K_z := B_\infty(z, 1/3) \subset \mathbb{R}^n$ as a small box around the corner of the hypercube corresponding to z. Then clearly $K_z \subset B_\infty(0,2)$, and finding a point close enough to K_z is enough to recover z.

We can also easily reduce a separation oracle query to a first-difference query to z, as follows. Suppose y is the vector we query:

1. If y is outside $[-1/3, 4/3]^n$, then output a hyperplane separating y from $[-1/3, 4/3]^n$.
2. If y is in $[-1/3, 4/3]^n$, then let g be the nearest corner of the hypercube.
3. Let i be the result of a first-difference query to z with g.
   (a) If $z = g$, then we know K_z exactly, so we can find a separating hyperplane or conclude that $y \in K_z$.
   (b) If $z \neq g$, then return e_i if $g_i = 1$, and $-e_i$ if $g_i = 0$.

Hence our $\Omega(n)$ quantum lower bound on learning z with first-difference queries implies an $\Omega(n)$ lower bound on the number of quantum queries to a separation oracle needed for finding a point in a convex set. □

Since optimization over a set K gives a point in the set K, this also implies a lower bound on the number of separation queries needed for optimization. This theorem is tight up to logarithmic factors, since it is known that $\widetilde{\mathcal{O}}(n)$ classical separation queries suffice for optimization, even without knowing a point in the convex set. Finally we remark that, due to our improved algorithm for optimization using validity queries, this also gives an $\widetilde{\Omega}(n)$ lower bound on the number of separation queries needed to implement validity.

12.6 Consequences of convex polarity

Here we justify the central symmetry of Figure 12.1 using the results of Grötschel, Lovász, and Schrijver [GLS88, Section 4.4]. We first need to recall the definition and some basic properties of the polar K^* of a set $K \subseteq \mathbb{R}^n$. This is the closed convex set defined as follows:

$$K^* = \{y \in \mathbb{R}^n : \langle y, x \rangle \leq 1 \text{ for all } x \in K\}.$$

It is straightforward to verify that if $B(0,r) \subseteq K \subseteq B(0,R)$, then $B(0,1/R) \subseteq K^* \subseteq B(0,1/r)$, moreover $(K^*)^* = K$ for closed convex sets.[15] For the remainder of this section we assume that K is a closed convex set such that $B(0,r) \subseteq K \subseteq B(0,R)$.

We will observe that for the polar K^* of a set K the following holds:

$$\text{MEM}(K^*) \leftrightarrow \text{VAL}(K), \qquad \text{SEP}(K^*) \leftrightarrow \text{VIOL}(K), \tag{12.16}$$

where $\text{MEM}(K^*) \leftrightarrow \text{VAL}(K)$ means we can implement a weak validity oracle for K using a single query to a weak membership oracle for K^*, and vice versa. Since $\text{VIOL}(K)$ and $\text{OPT}(K)$ are equivalent up to $\widetilde{\Theta}(1)$ reductions (via binary search), this justifies the central symmetry of Figure 12.1, because it shows that algorithms that implement $\text{VIOL}(K)$ given $\text{VAL}(K)$ are equivalent to algorithms that implement $\text{SEP}(K^*)$ given $\text{MEM}(K^*)$, and similarly algorithms that implement

[15] Note that K^* is a dual representation of the convex set K. Each point in K^* corresponds to a (normalized) valid inequality for K. This duality is not to be confused with Lagrangian duality.

SEP(K) given VIOL(K) are equivalent to algorithms that implement VIOL(K^*) given SEP(K^*).

Grötschel, Lovász, and Schrijver [GLS88, Section 4.4] showed that the weak membership problem for K^* can be solved using a single query to a weak validity oracle for K, and that the weak separation problem for K^* can be solved using a single query to a weak violation oracle for K. Using similar arguments one can show the reverse directions as well, which justifies (12.16). Here we only motivate the equivalences between the above-mentioned weak oracles by showing the equivalence of the strong oracles (i.e., where ρ and ε are 0).

Strong membership on K^* is equivalent to strong validity on K. First, for a given vector $c \in \mathbb{R}^n$ and a $\gamma > 0$ observe the following:

$$\frac{c}{\gamma} \notin \mathrm{int}(K^*) \iff \exists y \in K \text{ s.t. } \langle c/\gamma, y\rangle \geq 1 \iff \exists y \in K \text{ s.t. } \langle c, y\rangle \geq \gamma.$$

Hence, a strong membership query to K^* with a point c can be implemented by querying a strong validity oracle for K with the vector c and the value 1. Likewise, a strong validity query to K with a point c and value[16] $\gamma > 0$ can be implemented using a strong membership query to K^* with c/γ.

Strong separation on K^* is equivalent to strong violation on K. To implement a strong separation query on K^* for a vector $y \in \mathbb{R}^n$ we do the following. Query the strong violation oracle for K with y and the value 1. If the answer is that $\langle y, x\rangle \leq 1$ for all $x \in K$, then $y \in K^*$. If instead we are given a vector $x \in K$ with $\langle y, x\rangle \geq 1$, then x separates y from K^* (indeed, for all $z \in K^*$, we have $\langle z, x\rangle \leq 1 \leq \langle y, x\rangle$).

For the reverse direction, to implement a strong violation oracle for K on the vector c and value[16] $\gamma > 0$ we do the following. Query the strong separation oracle for K^* with the point c/γ. If the answer is that $c/\gamma \in K^*$ then $\langle c, x\rangle \leq \gamma$ for all $x \in K$. If instead we are given a non-zero vector $y \in \mathbb{R}^n$ that satisfies $\langle c/\gamma, y\rangle \geq \langle z, y\rangle$ for all $z \in K^*$, then $\tilde{y} = y/\langle c/\gamma, y\rangle$ will be a valid answer for the strong violation oracle for K. Indeed, we have $\tilde{y} \in K$ because $\langle z, \tilde{y}\rangle \leq 1$ for all $z \in K^*$ and $K = (K^*)^*$, and by construction $\langle c, \tilde{y}\rangle = \gamma$.

12.7 Future work

We mention several open problems for future work:

- Can we improve our $\Omega(\sqrt{n})$ lower bound on the number of separation queries needed to implement an optimization oracle when our algorithm knows a point in K? We conjecture that the correct bound is $\tilde{\Theta}(n)$, in which case knowing a point in K does not help a quantum algorithm.
- Can we improve on the *time complexity* of algorithms that implement an optimization oracle using (quantum) queries to a separation oracle?

[16] Observe that queries with value $\gamma \leq 0$ can be answered trivially, since $0 \in K$.

- Are there interesting convex optimization problems where separation is much harder than membership for classical computers? That is, problems for which deciding membership costs $\Theta(n^\alpha)$ while doing separation costs $\Theta(n^\beta)$, where $\beta - \alpha > 0$. Such problems would be good candidates for quantum speed-up in optimization in the real, non-oracle setting. It is known that given a deterministic algorithm for a function, an algorithm with roughly the same complexity can be constructed to compute the gradient of that function [GW08], so for deterministic oracles separation is not much harder than membership queries. This, however, still leaves randomized and quantum membership oracles to be considered.

- The algorithms that give an $\widetilde{\mathcal{O}}(n)$ upper bound on the number of separation queries for optimization (for example [LSW15, Theorem 42]) give the best theoretical results for many convex optimization problems. However, due to the large constants in these algorithms they are rarely used in a practical setting. A natural question is whether the algorithms used in practice lend themselves to quantum speed-ups as well. Very recent work by Kerenidis and Prakash [KP18] on quantum interior point methods is a first step in this direction.

Bibliography

[AA15] S. Aaronson and A. Ambainis. Forrelation: A problem that optimally separates quantum from classical computing. In *Proceedings of the 47th ACM Symposium on Theory of Computing (STOC)*, pages 307–316, 2015. arXiv: `1411.5729` 6, 178

[AAI$^+$16] S. Aaronson, A. Ambainis, J. Iraids, M. Kokainis, and J. Smotrovs. Polynomials, quantum query complexity and Grothendieck's inequality. In *31st Conference on Computational Complexity*, pages 25:1–25:9, 2016. arXiv: `1511.08682` 176, 178

[ABDK16] S. Aaronson, S. Ben-David, and R. Kothari. Separations in query complexity using cheat sheets. In *Proceedings of the 48th ACM Symposium on Theory of Computing (STOC)*, pages 863–876, 2016. arXiv: `1511.01937` 169, 176

[ABP19] S. Arunachalam, J. Briët, and C. Palazuelos. Quantum query algorithms are completely bounded forms. *SIAM Journal on Computing*, 48(3):903–925, 2019. 6, 169, 170, 171, 176, 177, 178, 179, 180

[AGKM16] S. Arora, R. Ge, R. Kannan, and A. Moitra. Computing a nonnegative matrix factorization—provably. *SIAM Journal on Computing*, 45(4):1582–1611, 2016. 20

[AHK12] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(6):121–164, 2012. 12, 189, 195

[AHKS06] D. Avis, J. Hasegawa, Y. Kikuchi, and Y. Sasaki. A quantum protocol to win the graph coloring game on all Hadamard graphs. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E89-A(5):1378–1381, 2006. 136, 152

[AK16] S. Arora and S. Kale. A combinatorial, primal-dual approach to semidefinite programs. *Journal of the ACM*, 63(2):12, 2016. Earlier version in STOC'07. 6, 189, 191, 195, 196, 197, 198

[AL12] M. F. Anjos and J. B. Lasserre. *Handbook on Semidefinite, Conic and Polynomial Optimization.* International Series in Operations Research & Management Science Series. Springer, 2012. 9

[Amb99] A. Ambainis. A better lower bound for quantum algorithms searching an ordered list. In *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 352–357, 1999. quant-ph/9902053. 246

[Amb02] A. Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64(4):750–767, 2002. Earlier version in STOC'00. quant-ph/0002066. 177

[AMR$^+$19] A. Atserias, L. Mančinska, D. E. Roberson, R. Šámal, S. Severini, and A. Varvitsiotis. Quantum and non-signalling graph isomorphisms. *Journal of Combinatorial Theory, Series B*, 136:289 – 328, 2019. 19

[ApS17] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 8.0.0.81*, 2017. 101

[Bar02] A. Barvinok. *A course in convexity*, volume 54 of *Graduate Studies in Mathematics*. American Mathematical Society, 2002. 10

[BB03] F. Barioli and A. Berman. The maximal cp-rank of rank k completely positive matrices. *Linear Algebra and its Applications*, 363:17–33, 2003. 19

[BBBV97] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997. quant-ph/9701001. 231

[BBC$^+$01] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. Earlier version in FOCS'98. 6, 177, 178

[BBCH$^+$17] G. Braun, J. Brown-Cohen, A. Huq, S. Pokutta, P. Raghavendra, A. Roy, B. Weitz, and D. Zink. The matching problem has no small symmetric sdp. *Mathematical Programming*, 165(2):643–662, 2017. 17

[BBHT98] M. Boyer, G. Brassard, P. Høyer, and A. Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4–5):493–505, 1998. Earlier version in Physcomp'96. quant-ph/9605034. 159, 160

[BCK15] D. W. Berry, A. M. Childs, and R. Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *Proceedings of the 56th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 792–809, 2015. arXiv: `1501.01715` 165

[BCKP13] S. Burgdorf, K. Cafuta, I. Klep, and J. Povh. The tracial moment problem and trace-optimization of polynomials. *Mathematical Programming*, 137(1):557–578, 2013. 58

[BCT99] G. Brassard, R. Cleve, and A. Tapp. The cost of exactly simulating quantum entanglement with classical communication. *Physical Review Letters*, 83(9):1874–1877, 1999. quant-ph/9901035. 152

[BCW98] H. Buhrman, R. Cleve, and A. Wigderson. Quantum vs. classical communication and computation. In *Proceedings of the 30th ACM Symposium on Theory of Computing (STOC)*, pages 63–68, 1998. quant-ph/9802040. 152

[BEK78] G. P. Barker, L. Q. Eifler, and T. P. Kezlan. A non-commutative spectral theorem. *Linear Algebra and its Applications*, 20(2):95–100, 1978. 48

[Bel64] J. S. Bell. On the Einstein Podolsky Rosen paradox. *Physics*, 1(3):195–200, 1964. 29, 31

[BFPS15] G. Braun, S. Fiorini, S. Pokutta, and D. Steurer. Approximation limits of linear programs (beyond hierarchies). *Mathematics of Operations Research*, 40(3):756–772, 2015. Earlier version in FOCS'12. 96

[BFS16] M. Berta, O. Fawzi, and V. B. Scholz. Quantum bilinear optimization. *SIAM Journal on Optimization*, 26(3):1529–1564, 2016. 79

[BHMT02] G. Brassard, P. Høyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Quantum Information: A Millennium Volume*, volume 305 of *AMS Contemporary Mathematics Series*, pages 53–74. 2002. quant-ph/0005055. 160, 161

[BK12] S. Burgdorf and I. Klep. The truncated tracial moment problem. *Journal of Operator Theory*, 68(1):141–163, 2012. 48, 51, 52, 54

[BKL$^+$17] F.G.S.L. Brandão, A. Kalev, T. Li, C. Yen-Yu Lin, K.M. Svore, and X. Wu. Quantum SDP solvers: Large speed-ups, optimality, and applications to quantum learning. arXiv: `1710.02581`, 2017. 187, 190, 224, 227

[BKP16] S. Burgdorf, I. Klep, and J. Povh. *Optimization of Polynomials in Non-Commutative Variables.* Springer Briefs in Mathematics. Springer, 2016. 50, 51, 59

[BKT18] M. Bun, R. Kothari, and J. Thaler. The polynomial method strikes back: tight quantum query bounds via dual polynomials. In *Proceedings of the 50th ACM Symposium on Theory of Computing (STOC)*, pages 297–310, 2018. arXiv:1710.09079. 184

[Bla06] B. Blackadar. *Operator Algebras: Theory of C^*-Algebras and Von Neumann Algebras.* Encyclopaedia of Mathematical Sciences. Springer, 2006. 47, 48, 55

[BLP17] S. Burgdorf, M. Laurent, and T. Piovesan. On the closure of the completely positive semidefinite cone and linear approximations to quantum colorings. *Electronic Journal of Linear Algebra*, 32:15–40, 2017. 18, 19, 101

[BPA$^+$08] N. Brunner, S. Pironio, A. Acin, N. Gisin, A. A. Méthot, and V. Scarani. Testing the dimension of Hilbert spaces. *Physical Review Letters*, 100:210503, 2008. 123, 124, 127

[BR87] O. Bratteli and D.W. Robinson. *Operator Algebras and Quantum Statistical Mechanics 1.* Theoretical and Mathematical Physics. Springer-Verlag Berlin Heidelberg, 2nd edition, 1987. 49

[BR06] A. Berman and U. G. Rothblum. A note on the computation of the cp-rank. *Linear Algebra and its Applications*, 419:1–7, 2006. 20

[BS17] F.G.S.L. Brandão and K.M. Svore. Quantum speed-ups for solving semidefinite programs. In *Proceedings of the 58th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 415–426, 2017. arXiv: `1609.05537` 187, 227

[BSM03] A. Berman and N. Shaked-Monderer. *Completely Positive Matrices.* World Scienfic, 2003. 18, 19, 20, 90

[BSS03] H. Barnum, M. E. Saks, and M. Szegedy. Quantum query complexity and semi-definite programming. In *18th Annual IEEE Conference on Computational Complexity*, pages 179–193, 2003. 171, 176, 178, 180

[BSU14] I. M. Bomze, W. Schachinger, and R. Ullrich. From seven to eleven: Completely positive matrices with high cp-rank. *Linear Algebra and its Applications*, 459:208 – 221, 2014. 19, 21, 23, 91, 92

[BSU15] I.M. Bomze, W. Schachinger, and R. Ullrich. New lower bounds and asymptotics for the cp-rank. *SIAM Journal on Matrix Analysis and Applications*, 36(1):20–37, 2015. 21, 91

[BT06] C. Bayer and J. Teichmann. The proof of Tchakaloff's theorem. *Proceedings of the American Mathematical Society*, 134:3035–3040, 2006. 57

[BT13] M. Bun and J. Thaler. Dual lower bounds for approximate degree and markov-bernstein inequalities. In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 303–314, 2013. 184

[BTN01] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization.* Society for Industrial and Applied Mathematics, 2001. 9, 16, 188

[Bub15] S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3–4):231–357, 2015. arXiv: `1405.4980` 225

[Bur09] S. Burer. On the copositive representation of binary and continuous nonconvex quadratic programs. *Mathematical Programming*, 120(2):479–495, 2009. 19

[BV04] S.P. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge Univ Press, 2004. 9

[BW02] H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: A survey. *Theoretical Computer Science*, 288(1):21–43, 2002. 169, 176

[BWHKN18] A. Bene Watts, A. W. Harrow, G. Kanwar, and A. Natarajan. Algorithms, bounds, and strategies for entanglex XOR games. In *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, volume 124, pages 10:1–10:18, 2018. arXiv: `1801.00821` 62

[CCLW18] S. Chakrabarti, A. M. Childs, T. Li, and X. Wu. Quantum algorithms and lower bounds for convex optimization. arXiv: `1809.01731`, 2018. 224, 227

[CCZ10] M. Conforti, G. Cornuéjols, and G. Zambelli. Extended formulations in combinatorial optimization. *4OR*, 8:1–48, 2010. 16

[CF96] R. E. Curto and L. A. Fialkow. *Solution of the Truncated Complex Moment Problem for Flat Data.* Memoirs of the American Mathematical Society. American Mathematical Society, 1996. 52, 57

[CG17] R. Connelly and S.J. Gortler. Universal rigidity of complete bipartite graphs. *Discrete & Computational Geometry*, 57(2):281–304, 2017. 109, 111

[CHSH69] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt. Proposed experiment to test local hidden-variable theories. *Physical Review Letters*, 23(15):880–884, 1969. 33, 35, 36

[CKP12] K. Cafuta, I. Klep, and J. Povh. Constrained polynomial optimization problems with noncommuting variables. *SIAM Journal on Optimization*, 22(2):363–383, 2012. 62

[CMN$^+$07] P.J. Cameron, A. Montanaro, M.W. Newman, S. Severini, and A. Winter. On the quantum chromatic number of a graph. *The Electronic Journal of Combinatorics*, 14(1), 2007. 136, 140, 145

[Con76] A. Connes. Classification of Injective Factors Cases II_1, II_∞, III_λ, $\lambda \neq 1$. *Annals of Mathematics*, 104(1):73–115, 1976. 63

[CS87] E. Christensen and A.M. Sinclair. Representations of completely bounded multilinear operators. *Journal of Functional Analysis*, 72(1):151 – 181, 1987. 180

[DD12] P. Dickinson and M. Dür. Linear-time complete positivity detection and decomposition of sparse matrices. *SIAM Journal on Matrix Analysis and Applications*, 33(3):701–720, 2012. 20

[DG14] P. J. C. Dickinson and L. Gijben. On the computational complexity of membership problems for the completely positive cone and its dual. *Computational Optimization and Applications*, 57(2):403–415, 2014. 21

[DH96] C. Dürr and P. Høyer. A quantum algorithm for finding the minimum. quant-ph/9607014, 18 Jul 1996. 161, 205

[DJL94] J. H. Drew, C. R. Johnson, and R. Loewy. Completely positive matrices associated with M-matrices. *Linear and Multilinear Algebra*, 37(4):303–310, 1994. 21, 91

[dKP02] E. de Klerk and D. V. Pasechnik. Approximation of the stability number of a graph via copositive programming. *SIAM Journal on Optimization*, 12(4):875–892, 2002. 2, 19

[dKV16] E. de Klerk and F. Vallentin. On the turing model complexity of interior point methods for semidefinite programming. *SIAM Journal on Optimization*, 26(3):1944–1961, 2016. 12

[DLTW08] A.C. Doherty, Y.-C. Liang, B. Toner, and S. Wehner. The quantum moment problem and bounds on entangled multiprover games. In *Proceedings of 23rd Annual IEEE Conference on Computational Complexity*, 2008. 32

[DP16] K. J. Dykema and V. I. Paulsen. Synchronous correlation matrices and Connes' embedding conjecture. *Journal of Mathematical Physics*, 57(1), 2016. 38, 138, 142

[DPP19] K. J. Dykema, V. I. Paulsen, and J. Prakash. Non-closure of the set of quantum correlations via graphs. *Communications in Mathematical Physics*, 365:1125–1142, 2019. 20, 32, 126, 153

[dW11] R. de Wolf. Quantum computing: Lecture notes. *http://homepages.cwi.nl/~rdewolf/qcnotes.pdf*, 2011. Lecture notes. 25, 157

[Edm65] J. Edmonds. Maximum matching and a polyhedron with $0, 1$ vertices. *Journal of Research of the National Bureau of Standards*, 69 B:125–130, 1965. 17

[EGH$^+$11] P. Etingof, O. Golberg, S. Hensel, T. Liu, A. Schwendner, D. Vaintrob, and E. Yudovina. Introduction to representation theory. *Lecture notes*, 2011. 119

[ENLV14] M. E.-Nagy, M. Laurent, and A. Varvitsiotis. Forbidden minor characterizations for low-rank optimal solutions to semidefinite programs over the elliptope. *Journal of Combinatorial Theory B*, 108:40–80, 2014. 106

[EPR35] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Physical Review*, 47:777–780, 1935. 27

[FFGT15] Y. Faenza, S. Fiorini, R. Grappe, and H.R. Tiwari. Extended formulations, non-negative factorizations and randomized communication protocols. *Mathematical Programming*, 153(1):75–94, 2015. 17

[FGP$^+$15] H. Fawzi, J. Gouveia, P. A. Parrilo, R. Z. Robinson, and R. R. Thomas. Positive semidefinite rank. *Mathematical Programming*, 153(1):133–177, 2015. 16, 81, 99

[FKPT13] S. Fiorini, V. Kaibel, K. Pashkovich, and D.O. Theis. Combinatorial bounds on nonnegative rank and extended formulations. *Discrete Mathematics*, 313(1):67–83, 2013. 94

[FMP$^+$15] S. Fiorini, S. Massar, S. Pokutta, H. R. Tiwary, and R. de Wolf. Exponential lower bounds for polytopes in combinatorial optimization. *Journal of the ACM*, 62(2):17:1–17:23, 2015. Earlier version in STOC'12. 17, 94

[FP15] H. Fawzi and P. A. Parrilo. Lower bounds on nonnegative rank via nonnegative nuclear norms. *Mathematical Programming*, 153(1):41–66, 2015. 70, 72, 94, 95

[FP16] H. Fawzi and P. A. Parrilo. Self-scaled bounds for atomic cone ranks: applications to nonnegative rank and cp-rank. *Mathematical Programming*, 158(1):417–465, 2016. 70, 71, 72, 83, 85, 90, 93, 94, 95, 96, 97

[FR87] P. Frankl and V. Rödl. Forbidden intersections. *Transactions of the American Mathematical Society*, 300(1):259–286, 1987. 153

[Fri12] T. Fritz. Tsirelson's problem and Kirchberg's conjecture. *Reviews in Mathematical Physics*, 24(05), 2012. 32

[FW14] P. E. Frenkel and M. Weiner. On vector configurations that can be realized in the cone of positive matrices. *Linear Algebra and its Applications*, 459:465 – 474, 2014. 19

[GAW] A. Gilyén, S. Arunachalam, and N. Wiebe. Optimizing quantum optimization algorithms via faster quantum gradient computation. In *Proceedings of the 30th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1425–1444. 227, 234, 235, 236

[GB14] M. Grant and S. Boyd. CVX: Matlab Software for Disciplined Convex Programming, version 2.1. `http://cvxr.com/cvx`, 2014. 101

[GD18] P. Groetzner and M. Dür. A factorization method for completely positive matrices. *Preprint*, 2018. 20

[GdLL17] S. Gribling, D. de Laat, and M. Laurent. Matrices with high completely positive semidefinite rank. *Linear Algebra and its Applications*, 513:122–148, 2017. 8, 15, 22, 23, 81, 103, 121

[GdLL18] S. Gribling, D. de Laat, and M. Laurent. Bounds on entanglement dimensions and quantum graph parameters via noncommutative polynomial optimization. *Mathematical Programming*, 170:5–42, 2018. 8, 38, 123, 135

[GdLL19] S. Gribling, D. de Laat, and M. Laurent. Lower bounds on matrix factorization ranks via noncommutative polynomial optimization. *Foundations of Computational Mathematics*, Jan 2019. 8, 43, 59, 67, 79, 80, 101

[GG12] N. Gillis and F. Glineur. On the geometric interpretation of the nonnegative rank. *Linear Algebra and its Applications*, 437(11):2685 – 2712, 2012. 96

[GGS17] A. P. Goucha, J. Gouveia, and P. M. Silva. On ranks of regular polygons. *SIAM Journal on Discrete Mathematics*, 31(4):2612–2625, 2017. 101

[Gil17] N. Gillis. Introduction to nonnegative matrix factorization. *SIAG/OPT Views and News*, 25(1):7–16, 2017. 20

[GJW18] M. Göös, R. Jain, and T. Watson. Extension complexity of independent set polytopes. *SIAM Journal on Computing*, 47(1):241–269, 2018. Earlier version in FOCS'16. 17

[GL08a] N. Gvozdenović and M. Laurent. Computing semidefinite programming lower bounds for the (fractional) chromatic number via block-diagonalization. *SIAM Journal on Optimization*, 19(2):592–615, 2008. 149

[GL08b] N. Gvozdenović and M. Laurent. The operator ψ for the chromatic number of a graph. *SIAM Journal on Optimization*, 19(2):572–591, 2008. 19, 145, 147, 149, 151

[GL19] S. Gribling and M. Laurent. Semidefinite programming formulations for the completely bounded norm of a tensor. arXiv: `1901.04921`, 2019. 8, 169

[GLS81] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981. 11, 12, 188

[GLS88] M. Grötschel, L. Lovász, and S. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer, 1988. 13, 225, 226, 230, 231, 249, 250

[Goe15] M. Goemans. Smallest compact formulation for the permutahedron. *Mathematical Programming*, 153(1):5–11, 2015. 16

[GP90] R. Grone and S. Pierce. Extremal bipartite matrices. *Linear Algebra and its Applications*, 131:39–50, 1990. 111

[GP92] N. Gisin and A. Peres. Maximal violation of Bell's inequality for arbitrarily large spin. *Physics Letters A*, 166:15–17, 1992. 30

[GPT13] J. Gouveia, P. A. Parrilo, and R. R. Thomas. Lifts of convex sets and cone factorizations. *Mathematics of Operations Research*, 38(2):248–264, 2013. 15, 17, 96

[Gro96] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th ACM Symposium on Theory of Computing (STOC)*, pages 212–219, 1996. quant-ph/9605043. 5, 159

[GRT13] J. Gouveia, R. Z. Robinson, and R. R. Thomas. Polytopes of minimum positive semidefinite rank. *Discrete & Computational Geometry*, 50(3):679–699, 2013. 100

[GSLW18] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. arXiv: `1806.01838`, 2018. 164, 165, 166, 187

[GW08] A. Griewank and A. Walther. *Evaluating derivatives - principles and techniques of algorithmic differentiation.* SIAM, 2nd edition, 2008. 251

[Hav36] E. K. Haviland. On the Momentum Problem for Distribution Functions in More Than One Dimension. II. *American Journal of Mathematics*, 58(1):164–168, 1936. 56

[Hea15] B. Hensen et al. Loophole-free Bell inequality violation using electron spins separated by 1.3 kilometres. *Nature*, 526:682–686, 2015. 36

[HK18] G.A. Hanasusanto and D. Kuhn. Conic programming reformulations of two-stage distributionally robust linear programs over wasserstein balls. *Operations Research*, 66(3):849–869, 2018. 19

[HL83] J. Hanna and T.J. Laffey. Nonnegative factorization of completely positive matrices. *Linear Algebra and its Applications*, 55:1–9, 1983. 19

[HLŠ07] P. Høyer, T. Lee, and R. Špalek. Negative weights make adversaries stronger. In *Proceedings of the 39th ACM Symposium on Theory of Computing (STOC)*, pages 526–535, 2007. quant-ph/0611054. 171, 177, 180, 246

[HM04] J. W. Helton and S. A. McCullough. A positivstellensatz for noncommutative polynomials. *Transactions of the American Mathematical Society*, 356(9):3721–3737, 2004. 63

[Ji13] Z. Ji. Binary constraint system games and locally commutative reductions. arXiv: `1310:3794`, 2013. 104, 121, 137

[JNP$^+$11] M. Junge, M. Navascues, C. Palazuelos, D. Perez-Garcia, V.B. Scholtz, and R.F. Werner. Connes' embedding problem and Tsirelson's problem. *J. Math. Physics*, 52(012102), 2011. 32

[Jor05] S. P. Jordan. Fast quantum algorithm for numerical gradient estimation. *Physical Review Letters*, 95(5):050501, 2005. quant-ph/0405146. 227, 234

[Jor08] S. P. Jordan. *Quantum Computation Beyond the Circuit Model.* PhD thesis, Massachusetts Institute of Technology, 2008. arXiv: `0809.2307` 227

[JSWZ13] R. Jain, Y. Shi, Z. Wei, and S. Zhang. Efficient protocols for generating bipartite classical distributions and quantum states. *IEEE Transactions on Information Theory*, 59(8):5171–5178, 2013. 17

[Kar84] N. Karmakar. A new polynomial time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984. 16

[Kas95] C. Kassel. *Quantum Groups.* Graduate texts in mathematics. Springer-Verlag, 1995. 119

[Kim13] S. Kimmel. Quantum adversary (upper) bound. *Chicago Journal of Theoretical Computer Science*, 2013. 220

[KP16] I. Klep and J. Povh. Constrained trace-optimization of polynomials in freely noncommuting variables. *Journal of Global Optimization*, 64(2):325–348, 2016. 51, 54, 58, 59

[KP18] I. Kerenidis and A. Prakash. A quantum interior point method for LPs and SDPs. arXiv: `1808.09266`, 2018. 224, 227, 251

[KR97] R.V. Kadison and J.R. Ringrose. *Fundamentals of the Theory of Operator Algebras. Volume 1: Elementary Theory.* Graduate studies in Mathematics. American Mathematical Society, 1997. 49

[KS08] I. Klep and M. Schweighofer. Connes' embedding conjecture and sums of Hermitian squares. *Advances in Mathematics*, 217(4):1816–1837, 2008. 32, 59, 62, 63

[Las01] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001. 58, 143

[Las09] J. B. Lasserre. *Moments, Positive Polynomials and Their Applications.* Imperial College Press, 2009. 58

[Lau03] M. Laurent. A comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre relaxations for 0-1 programming. *Math. Oper. Res.*, 28(3):470–496, 2003. 143

[Lau09] M. Laurent. Sums of squares, moment matrices and optimization over polynomials. In M. Putinar and S. Sullivant, editors, *Emerging Applications of Algebraic Geometry*, pages 157–270. Springer, 2009. 58, 61

[LMR$^+$11] T. Lee, R. Mittal, B. Reichardt, R. Špalek, and M. Szegedy. Quantum query complexity of state conversion. In *Proceedings of the 52nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 344–353, 2011. arXiv: `1011.3020` 177

[Lov78] L. Lovász. Kneser's conjecture, chromatic number, and homotopy. *Journal of Combinatorial Theory, Series A*, 25(3):319 – 324, 1978. 146

[Lov79] L. Lovász. On the Shannon Capacity of a Graph. *IEEE Transactions on Information Theory*, 25(1):1–7, 1979. 2, 146

[Lov03] L. Lovász. Semidefinite programs and combinatorial optimization. In *Recent advances in algorithms and combinatorics*, volume 11 of *CMS Books Math./Ouvrages Math. SMC*, pages 137–194. Springer, New York, 2003. 9, 11

[LP15] M. Laurent and T. Piovesan. Conic approach to quantum graph parameters using linear optimization over the completely positive semidefinite cone. *SIAM Journal on Optimization*, 25(4):2461–2493, 2015. 3, 18, 19, 104

[LRS15] J. R. Lee, P. Raghavendra, and D. Steurer. Lower bounds on the size of semidefinite programming relaxations. In *Proceedings of the 47th ACM Symposium on Theory of Computing (STOC)*, pages 567–576, 2015. arXiv: `1411.6317` 17

[LSV18] Y. T. Lee, A. Sidford, and S. S. Vempala. Efficient convex optimization with membership oracles. In *Proceedings of the 31st Conference On Learning Theory (COLT)*, pages 1292–1294, 2018. arXiv: `1706.07357` 226, 227, 230, 232, 239, 241

[LSW15] Y. T. Lee, A. Sidford, and S. C. Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *Proceedings of the 56th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1049–1065, 2015. arXiv: `1508.04874` 12, 226, 251

[LT95] C.-K. Li and B.S. Tam. A note on extreme correlation matrices. *SIAM Journal on Matrix Analysis and Applications*, 15(3):903–908, 1995. 108

[LV14] M. Laurent and A. Varvitsiotis. Positive semidefinite matrix completion, universal rigidity and the strong arnold property. *Linear Algebra and its Applications*, 452(1):292–317, 2014. 109

[LWdW17] T. Lee, Z. Wei, and R. de Wolf. Some upper and lower bounds on psd-rank. *Mathematical Programming*, 162(1):495–521, 2017. 72, 98, 99, 101

[Mar91] R. K. Martin. Using separation algorithms to generate mixed integer model reformulations. *Operations Research Letters*, 10(3):119–128, 1991. 16

[Mer90] N. David Mermin. Simple unified form for the major no-hidden-variables theorems. *Phys. Rev. Lett.*, 65:3373–3376, Dec 1990. 33

[MGS81] L. Lovász M. Grötschel and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981. 16

[Moi16] A. Moitra. An almost optimal algorithm for computing nonnegative rank. *SIAM Journal on Computing*, 45(1):156–173, 2016. 20

[Moi18] A. Moitra. *Algorithmic Aspects of Machine Learning.* Cambridge University Press, 2018. 17

[MP68] M. Minsky and S. Papert. *Perceptrons.* MIT Press, Cambridge, MA, 1968. Second, expanded edition 1988. 6

[MR14] L. Mančinska and D. E. Roberson. Note on the correspondence between quantum correlations and the completely positive semidefinite cone. *Available at quantuminfo. quantumlah. org/memberpages/ laura/ corr. pdf*, 2014. 19, 37

[MR16a] L. Mančinska and D. E. Roberson. Oddities of quantum colorings. *Baltic Journal on Modern Computing*, 4:846–859, 2016. 153

[MR16b] L. Mančinska and D. E. Roberson. Quantum homomorphisms. *Journal of Combinatorial Theory, Series B*, 118:228 – 267, 2016. 38, 136, 137, 138, 140, 145, 146, 147, 149, 150, 152, 153

[MRR03] S. T. McCormick, M.R. Rao, and G. Rinaldi. Easy and difficult objective functions for max cut. *Mathematical Programming*, 94(2):459–466, 2003. 35

[MSS13] L. Mančinska, G. Scarpa, and S. Severini. New separations in zero-error channel capacity through projective Kochen-Specker sets and quantum coloring. *IEEE Transactions on Information Theory*, 59(6):4025–4032, 2013. 152

[MSvS03] D. Mond, J. Smith, and D. van Straten. Stochastic factorizations, sandwiched simplices and the topology of the space of explanations. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 459(2039):2821–2845, 2003. 96

[NC00] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000. 25, 32, 157, 158, 159, 163, 229, 235

[Nie14a] J. Nie. The $\mathcal{A}$-truncated K-moment problem. *Foundations of Computational Mathematics*, 14(6):1243–1276, 2014. 101

[Nie14b] Jiawang Nie. Optimality conditions and finite convergence of lasserre's hierarchy. *Mathematical Programming*, 146(1):97–121, 2014. 61

[Nie17] J. Nie. Symmetric tensor nuclear norms. *SIAM Journal on Applied Algebra and Geometry*, 1(1):599–625, 2017. 68, 95

[NN94] Y. Nesterov and A. Nemirovski. *Interior-point polynomial algorithms in convex programming*, volume 13 of *SIAM Studies in Applied Mathematics.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994. 12, 188

[NPA08] M. Navascués, S. Pironio, and A. Acín. A convergent hierarchy of semidefinite programs characterizing the set of quantum correlations. *New Journal of Physics*, 10(7):073013, 2008. 62, 142

[NPA12] M. Navascués, S. Pironio, and A. Acín. SDP relaxations for non-commutative polynomial optimization. In M. F. Anjos and J. B. Lasserre, editors, *Handbook on Semidefinite, Conic and Polynomial Optimization*, pages 601–634. Springer, 2012. 50, 58, 132

[NS94] N. Nisan and M. Szegedy. On the degree of Boolean functions as real polynomials. *Computational Complexity*, 4(4):301–313, 1994. Earlier version in STOC'92. 6

[NTZ11] K. Natarajan, C.P. Teo, and Z. Zheng. Mixed 0-1 linear programs under objective uncertainty: A completely positive representation. *Operations Research*, 59(3):713–728, 2011. 19

[O'D14] R. O'Donnell. *Analysis of Boolean Functions.* Cambridge University Press, 2014. 182

[OP16] C. M. Ortiz and V. I. Paulsen. Quantum graph homomorphisms via operator systems. *Linear Algebra and its Applications*, 497:23 – 43, 2016. 140

[Oza13] N. Ozawa. About the Connes' embedding problem–algebraic approaches. *Japanese Journal of Mathematics*, 8(1):147–183, 2013. 32

[Par00] P. A. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization.* PhD thesis, Caltech, 2000. 58, 95

[Per90] A. Peres. Incompatible results of quantum measurements. *Physics Letters A*, 151(3):107 – 108, 1990. 33

[PK97] L. Porkolab and L. Khachiyan. On the complexity of semidefinite programs. *Journal of Global Optimization*, 10(4):351–365, June 1997. 12

[PNA10] S. Pironio, M. Navascués, and A. Acín. Convergent relaxations of polynomial optimization problems with noncommuting variables. *SIAM Journal on Optimization*, 20(5):2157–2180, 2010. 43, 52, 58, 62, 142

[PR92] S. Popescu and D. Rohrlich. Generic quantum nonlocality. *Physics Letters A*, 166:293–297, 1992. 30

[Pro07] C. Procesi. *Lie groups - An approach through Invariants and Representations.* Springer, 2007. 115

[PSS$^+$16] V. I. Paulsen, S. Severini, D. Stahlke, I. G. Todorov, and A. Winter. Estimating quantum chromatic numbers. *Journal of Functional Analysis*, 270(6):2188 – 2222, 2016. 38, 138, 139, 142, 145, 146, 148

[PSVW18] A. Prakash, J. Sikora, A. Varvitsiotis, and Z. Wei. Completely positive semidefinite rank. *Mathematical Programming*, 171:397–431, 2018. 21, 71, 81, 82, 121

[Put93] M. Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42:969–984, 1993. 55, 56, 63

[PV08] K F. Pál and T. Vértesi. Efficiency of higher-dimensional Hilbert spaces for the violation of Bell inequalities. *Physical Review A*, 77:042105, 2008. 123

[PV16] C. Palazuelos and T. Vidick. Survey on nonlocal games and operator space theory. *Journal of Mathematical Physics*, 57(1):015220, 2016. 28, 170

[PV18] A. Prakash and A. Varvitsiotis. Correlation matrices, clifford algebras, and completely positive semidefinite rank. *Linear and Multilinear Algebra*, 0(0):1–18, 2018. 121

[PW09] D. Poulin and P. Wocjan. Preparing ground states of quantum many-body systems on a quantum computer. *Physical Review Letters*, 102:130503, 2009. arXiv: `0809.2705` 163

[Ram97] M.V. Ramana. An exact duality theory for semidefinite programming and its complexity implications. *Mathematical Programming*, 77(1):129–162, 1997. 12

[Rei09] B. Reichardt. Span programs and quantum query complexity. In *Proceedings of the 50th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 544–551, 2009. arXiv: `0904.2759` 177

[Rei11] B. Reichardt. Reflections for quantum query algorithms. In *Proceedings of the 22nd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 560–569, 2011. arXiv: `1005.1601` 177

[Ren88] J. Renegar. A polynomial-time algorithm, based on newton's method, for linear programming. *Mathematical Programming*, 40(1):59–93, 1988. 16

[Ren92] J. 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. 12, 20

[Ren01] J. Renegar. *A Mathematical View of Interior-Point Methods in Convex Optimization.* Society for Industrial and Applied Mathematics, 2001. 12

[Ren16] J. Renegar. "efficient subgradient methods for general convex optimization. *SIAM Journal on Optimization*, 26(4):2649–2676, 2016. 189

[Ren19] James Renegar. Accelerated first-order methods for hyperbolic programming. *Mathematical Programming*, 173(1):1–35, 2019. 189

[Rob13] D. E. Roberson. *Variations on a Theme: Graph Homomorphisms.* PhD thesis, University of Waterloo, 2013. 136, 138, 143, 151

[Rot17] T. Rothvoß. The matching polytope has exponential extension complexity. *Journal of the ACM*, 64(6), 2017. Earlier version in STOC'14. 17, 71, 94

[Rud87] W. Rudin. *Real and complex analysis.* Mathematics series. McGraw-Hill, 1987. 55

[She13] A. Sherstov. Approximating the AND-OR Tree. *Theory of Computing*, 9(20):653–663, 2013. 184

[Shi16] Y. Shitov. A universality theorem for nonnegative matrix factorizations. *arXiv:1606.09068v2*, 2016. 21

[Shi17] Y. Shitov. The complexity of positive semidefinite matrix factorization. *SIAM Journal on Optimization*, 27(3):1898–1909, 2017. 21

[Shi18] Y. Shitov. Matrices of bounded psd rank are easy to detect. *SIAM Journal on Optimization*, 28(3):2067–2072, 2018. 20

[Sho97] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. Earlier version in FOCS'94. quant-ph/9508027. 5

[Slo11] W. Slofstra. Lower bounds on the entanglement needed to play xor non-local games. *Journal of Mathematical Physics*, 52:102202, 2011. 104, 111, 112, 113, 114

[Slo16] W. Slofstra. Tsirelson's problem and an embedding theorem for groups arising from non-local games. arXiv: `1606.03140`, 2016. 33, 62

[Slo19] W. Slofstra. The set of quantum correlations is not closed. *Forum of Mathematics, Pi*, 7, 2019. 20, 32, 33, 103, 104, 126

[SMBB$^+$15] N. Shaked-Monderer, A. Berman, I. M. Bomze, F. Jarre, and W. Schachinger. New results on the cp-rank and related properties of co(mpletely)positive matrices. *Linear and Multilinear Algebra*, 63(2):384–396, 2015. 19

[SMBJS13] N. Shaked-Monderer, I. M. Bomze, F. Jarre, and W. Schachinger. On the cp-rank and minimal cp factorizations of a completely positive matrix. *SIAM Journal on Matrix Analysis and Applications*, 34(2):355–368, 2013. 19

[Špa08] R. Špalek. A dual polynomial for OR. *arXiv: 0803.4516*, 2008. 184

[Sta18] C. Stark. Learning optimal quantum models is NP-hard. *Phys. Rev. A*, 97:020103, 2018. 30

[SV17] J. Sikora and A. Varvitsiotis. Linear conic formulations for two-party correlations and values of nonlocal games. *Mathematical Programming*, 162(1):431–463, 2017. 19, 37, 38, 104, 140

[SVW16] J. Sikora, A. Varvitsiotis, and Z. Wei. Minimum dimension of a Hilbert space needed to generate a quantum correlation. *Physical Review Letters*, 2016. 32, 123

[Swa86] T. Swart. P = NP. Technical report, University of Guelph, 1986. Revision 1987. 16

[Sze94] M. Szegedy. A note on the theta number of Lovász and the generalized Delsarte bound. In *Proceedings of the 35th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 36–39, 1994. 147

[TRW05] K. Tsuda, G. Rätsch, and M. K. Warmuth. Matrix exponentiated gradient updates for on-line learning and Bregman projection. *Journal of Machine Learning Research*, 6:995–1018, 2005. Earlier version in NIPS'04. 189

[TS15] G. Tang and P. Shah. Guaranteed tensor decomposition: A moment approach. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 1491–1500, 2015. 68, 95

[Tsi87] B.S. Tsirel'son. Quantum analogues of the bell inequalities. the case of two spatially separated domains. *Journal of Soviet Mathematics*, 36(4):557–570, 1987. 35, 104, 105, 106, 115, 117, 170

[Tsi93] B.S. Tsirel'son. Some results and problems on quantum bell-type inequalities. *Hadronic Journal Supplement*, 8(4):329–345, 1993. 104, 108, 109, 110, 111

[Tsi06] B. Tsirelson. Bell inequalities and operator algebras. *http://www.tau.ac.il/~tsirel/download/bellopalg.pdf*, 2006. 32

[vAG18a] J. van Apeldoorn and A. Gilyén. Improvements in quantum SDP-solving with applications. arXiv: `1804.05058`, 2018. 187, 190, 223, 224, 227

[vAG18b] J. van Apeldoorn and S. Gribling. Simon's problem for linear functions. arXiv: `1810.12030`, 2018. 8

[vAGGdW17] J. van Apeldoorn, A. Gilyén, S. Gribling, and R. de Wolf. Quantum SDP-solvers: Better upper and lower bounds. In *Proceedings of the 58th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 403–414, 2017. arXiv: `1705.01843` 8, 161, 162, 187, 189, 195, 211, 224, 227

[vAGGdW18] J. van Apeldoorn, A. Gilyén, S. Gribling, and R. de Wolf. Convex optimization using quantum oracles. arXiv: `1809.00643`, 2018. 8, 224, 225, 236, 237

[Vav09] S. A. Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 20(3):1364–1377, 2009. 20

[VB96] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Rev.*, 38:49–95, 1996. 9

[VGG18] A. Vandaele, F. Gineur, and N. Gillis. Algorithms for positive semidefinite factorization. *Computational Optimization and Applications*, 71(1):193–219, 2018. 20

[Vid16] T. Vidick. Three-Player Entangled XOR Games are NP-Hard to Approximate. *SIAM Journal on Computing*, 45(3):1007–1063, 2016. 170

[VP09] T. Vértesi and K.F. Pál. Bounding the dimension of bipartite quantum systems. *Physical Review A*, 79, 2009. 104, 111

[Wat09] J. Watrous. Semidefinite programs for completely bounded norms. *Theory of Computing*, 5:217–238, 2009. 171

[Wat11] J. Watrous. Theory of quantum information. *https://cs.uwaterloo.ca/~watrous/LectureNotes.html*, 2011. Lecture notes. 25, 38, 157

[WCD08] S. Wehner, M. Christandl, and A. C. Doherty. Lower bound on the dimension of a quantum system given measured data. *Physical Review A*, 78:062112, 2008. 123

[Wed64] J. H. M. Wedderburn. *Lectures on Matrices.* Dover Publications Inc., 1964. 48

[WK12] M. K. Warmuth and D. Kuzmin. Online variance minimization. *Machine Learning*, 87(1):1–32, 2012. Earlier version in COLT'06. 189

[Wol08] R. de Wolf. A brief introduction to Fourier analysis on the Boolean cube. *Theory of Computing*, 2008. ToC Library, Graduate Surveys 1. 182

[WSV00] H. Wolkowicz, R. Saigal, and L. Vandenberghe. *Handbook of Semidefinite Programming.* International Series in Operations Research & Management Science Series. Springer, 2000. 9

[Yan91] M. Yannakakis. Expressing combinatorial optimization problems by linear programs. *Journal of Computer and System Sciences*, 43(3):441 – 466, 1991. Earlier version in STOC'88. 16, 17

Index

List of Symbols

$[n]$	The set $\{1, 2, \ldots, n\}$.
$\mathbb{N}$	The set of nonnegative integers.
$\mathbb{R}$	The set of real numbers.
$\mathbb{C}$	The set of complex numbers.
$\log$	The base-2 logarithm.
$\mathrm{conv}(V)$	The convex hull of the set of points V.
$\mathrm{Sym}(\ell)$	The set of permutations of ℓ elements.
$V_1 \sqcup V_2$	The disjoint union of the sets V_1 and V_2.

Vectors

$\mathbb{R}^n$	The set of real n-dimensional vectors.
$\mathbb{S}^{n-1}$	The $(n-1)$-dimensional unit sphere in $\mathbb{R}^n$.
$\mathbb{R}^n_+$	The set of real n-dimensional entrywise-nonnegative vectors.
$\mathbb{C}^n$	The set of complex n-dimensional vectors.
e	The all-ones vector.
e_i	The vector that equals zero everywhere except at the ith coordinate, where it equals one.
v^T	The transpose of a vector v.
v^*	The complex conjugate of the transpose of v.
$\mathcal{H}$	A Hilbert space.
$\langle u, v \rangle$	The inner product between vectors u and v.
$\|v\|_p$	The p-norm of the vector $v \in \mathbb{C}^n$: $\|v\|_p = \left(\sum_{i=1}^n \lvert v_i \rvert^p\right)^{1/p}$.
$\|v\|$	The 2-norm of the vector $v \in \mathbb{C}^n$.

Matrices

$\mathbb{R}^{m\times n}$	The set of real $m \times n$ matrices.
$\mathbb{R}_+^{m\times n}$	The set of real entrywise-nonnegative $m \times n$ matrices.
CP^n	The set of $n \times n$ completely positive matrices.
CS_+^n	The set of $n \times n$ completely positive semidefinite matrices.
S^n	The set of real symmetric $n \times n$ matrices.
S_+^n	The set of real symmetric $n \times n$ positive semidefinite matrices.
$\mathcal{E}^n$	The elliptope: the set of matrices $A \in \mathrm{S}_+^n$ that satisfy $A_{ii} = 1$ for all $i \in [n]$.
$\mathrm{Cor}(m, n)$	The set of $m \times n$ bipartite correlation matrices.
$A \succeq 0$	The matrix A is positive semidefinite.
$A \succ 0$	The matrix A is positive definite.
$\mathrm{Gram}(V)$	The Gram matrix $\big(\langle v_i, v_j\rangle\big)_{i,j}$ associated to a set of vectors $V = \{v_1, \ldots, v_n\}$.
$O(d)$	The set of $d \times d$ real orthogonal matrices.
$\mathbb{C}^{m\times n}$	The set of complex $m \times n$ matrices.
$\mathrm{Re}(A)$	The real part of a matrix $A \in \mathbb{C}^{m\times n}$.
$\mathrm{Im}(A)$	The imaginary part of a matrix $A \in \mathbb{C}^{m\times n}$.
H^n	The set of complex Hermitian $n \times n$ matrices.
H_+^n	The set of complex Hermitian $n \times n$ positive semidefinite matrices.
$\mathcal{B}(\mathcal{H})$	The set of bounded linear operators on the Hilbert space $\mathcal{H}$.
$\mathrm{Diag}(v)$	The diagonal matrix whose main diagonal is the vector v.
$\mathrm{diag}(A)$	The vector corresponding to the main diagonal of the matrix A.
$\mathrm{vec}(A)$	The vector obtained by stacking the columns of the matrix A on top of each other.
$\mathrm{Tr}(A)$	The trace of a matrix $A \in \mathbb{C}^{n\times n}$: $\mathrm{Tr}(A) = \sum_{i=1}^n A_{ii}$.
$\mathrm{tr}(A)$	The normalized trace of a matrix $A \in \mathbb{C}^{n\times n}$: $\mathrm{tr}(A) = \frac{1}{n}\sum_{i=1}^n A_{ii}$.
$\langle A, B\rangle$	The trace inner product $\mathrm{Tr}(A^*B)$ between matrices A and B.
$\mathrm{rank}(A)$	The rank of a matrix A.
$\mathrm{rank}_+(A)$	The nonnegative rank of a matrix $A \in \mathbb{R}_+^{m\times n}$.
$\text{psd-rank}_{\mathbb{R}}(A)$	The real positive semidefinite rank of a matrix $A \in \mathbb{R}_+^{m\times n}$.
$\text{cpsd-rank}_{\mathbb{C}}(A)$	The complex positive semidefinite rank of a matrix $A \in \mathbb{R}_+^{m\times n}$.
$\text{cp-rank}(A)$	The completely positive rank of a matrix $A \in \mathrm{CP}^n$.
$\text{cpsd-rank}_{\mathbb{R}}(A)$	The real completely positive semidefinite rank of a matrix $A \in \mathrm{CS}_+^n$.
$\text{cpsd-rank}_{\mathbb{C}}(A)$	The complex completely positive semidefinite rank of a matrix $A \in \mathrm{CS}_+^n$.
$\otimes$	The tensor product (also called the Kronecker product).
$\oplus$	The direct sum.
$\lVert A\rVert$	The operator norm of an operator A.

Quantum information theory

$\lvert\psi\rangle$	The Dirac notation for a column-vector $\psi \in \mathbb{C}^n$.
$\langle\psi\rvert$	The conjugate transpose of the vector $\lvert\psi\rangle$.
$C_{loc}(\Gamma)$	The set of classical bipartite correlations.
$C_{loc,s}(\Gamma)$	The set of synchronous bipartite correlations $P \in C_{loc}(\Gamma)$.
$C_q(\Gamma)$	The set of bipartite quantum correlations in the tensor model.
$C_{q,s}(\Gamma)$	The set of synchronous bipartite correlations $P \in C_q(\Gamma)$.
$C_{qc}(\Gamma)$	The set of bipartite quantum correlations in the commuting operator model.
$C_{qc,s}(\Gamma)$	The set of synchronous correlations $P \in C_{qc}(\Gamma)$.
$D_q(P)$	The entanglement dimension of a bipartite correlation $P \in C_q(\Gamma)$ in the tensor model.
$D_{qc}(P)$	The entanglement dimension of a bipartite correlation $P \in C_{qc}(\Gamma)$ in the commuting operator model.
$A_q(P)$	The average entanglement dimension of a bipartite correlation $P \in C_q(\Gamma)$.

Polynomial optimization

$\mathbf{x}$	The tuple of noncommutative symbols $x_1, \dots, x_n$.
$\langle\mathbf{x}\rangle_t$	The set of words in the noncommutative symbols $x_1, \dots, x_n$ of length at most t.
$\mathbb{R}\langle\mathbf{x}\rangle_t$	The set of noncommutative polynomials of degree at most t.
$\mathbb{R}\langle\mathbf{x}\rangle_t^*$	The set of real-valued linear functionals on $\mathbb{R}\langle\mathbf{x}\rangle_t$.
$[\mathbf{x}]_t$	The set $[\mathbf{x}]_t = [x_1, \dots, x_n]_t$ of words in the commutative symbols $x_1, \dots, x_n$ of length at most t.
$\mathbb{R}[\mathbf{x}]_t$	The set of commutative polynomials of degree at most t.
$\mathbb{R}[\mathbf{x}]_t^*$	The set of real-valued linear functionals on $\mathbb{R}[\mathbf{x}]_t$.
$M(L)$	The moment matrix associated to a linear functional $L \in \mathbb{R}\langle\mathbf{x}\rangle_t^*$ or $L \in \mathbb{R}[\mathbf{x}]_t^*$.
$\mathcal{M}_{2t}(S)$	The degree-$2t$ truncated quadratic module generated by the set of polynomials S.
$D(S)$	The scalar positivity domain associated to a set of polynomials S.
$\mathcal{D}(S)$	The matrix positivity domain associated to a set of polynomials S.
$\mathcal{D}_{\mathcal{A}}(S)$	The positivity domain associated to a set of polynomials S, in a C^*-algebra $\mathcal{A}$.
$\mathcal{I}_t(T)$	The degree-t truncated (left) ideal generated by the set of polynomials T.
$V(T)$	The scalar variety associated to a set of polynomials T.
$\mathcal{V}(T)$	The matrix variety associated to a set of polynomials T.
$\mathcal{V}_{\mathcal{A}}(T)$	The variety associated to a set of polynomials T, in a C^*-algebra $\mathcal{A}$.

Graph theory

$\alpha(\cdot)$	The stability number.
$\alpha_q(\cdot)$	The quantum stability number.
$\alpha_{qc}(\cdot)$	The commuting quantum stability number.
$\alpha_p(\cdot)$	The projective packing number.
$\chi(\cdot)$	The chromatic number of a graph.
$\chi_q(\cdot)$	The quantum chromatic number.
$\chi_{qc}(\cdot)$	The commuting quantum chromatic number.
$\chi_f(\cdot)$	The fractional chromatic number.
$\xi_f(\cdot)$	The projective rank.
$\xi_{tr}(\cdot)$	The tracial rank.
$G\square H$	The Cartesian product of the graphs G and H.
$G \ltimes H$	The homomorphic product of the graphs G and H.
$G \star H$	The graph product $G \ltimes \overline{H}$ of the graphs G and H.