Semi-definite programming and quantum information

OPEN ACCESS
IOP Publishing
Journal of Physics A: Mathematical and Theoretical
J. Phys. A: Math. Theor. **57** (2024) 163002 (93pp)
https://doi.org/10.1088/1751-8121/ad2b85

Topical Review

Semi-definite programming and quantum information

Piotr Mironowicz[1,2,3]

[1] Department of Physics, Stockholm University, S-10691 Stockholm, Sweden
[2] Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Narutowicza 11/12, 80-233 Gdańsk, Poland
[3] International Centre for Theory of Quantum Technologies, University of Gdańsk, Wita Stwosza 57, 80-308 Gdańsk, Poland

E-mail: piotr.mironowicz@gmail.com

Received 26 July 2023; revised 31 October 2023
Accepted for publication 21 February 2024
Published 8 April 2024

Abstract

This paper presents a comprehensive exploration of semi-definite programming (SDP) techniques within the context of quantum information. It examines the mathematical foundations of convex optimization, duality, and SDP formulations, providing a solid theoretical framework for addressing optimization challenges in quantum systems. By leveraging these tools, researchers and practitioners can characterize classical and quantum correlations, optimize quantum states, and design efficient quantum algorithms and protocols. The paper also discusses implementational aspects, such as solvers for SDP and modeling tools, enabling the effective employment of optimization techniques in quantum information processing. The insights and methodologies presented in this paper have proven instrumental in advancing the field of quantum information, facilitating the development of novel communication protocols, self-testing methods, and a deeper understanding of quantum entanglement.

Keywords: semidefinite programming, quantum information, quantum correlations, theta function, NPA, sum of squares, duality

Contents

List of abbreviations

DPS	Doherty–Parillo–Spedalieri
IPM	interior point method
LP	linear programming
LMI	linear matrix inequalities
MLP	Mironowicz–Li–Pawłowski
NPA	Navascués–Pironio–Acín
NV	Navascués–Vertesi
PD	positive definite
POVM	positive operator-valued measure
PPT	positive partial transpose
PSD	positive semi-definite
QSD	quantum state discrimination
SDP	semi-definite programming
SoS	sum of squares

1. Introduction

Optimization, in its various forms, has been a cornerstone of scientific and technological advancements across numerous disciplines. From engineering and economics to machine learning and operations research, optimization techniques have played a crucial role in solving complex problems and driving innovation. Over the years, different variants of optimization methods have emerged, each tailored to address specific problem structures and objectives [15, 24, 36, 40, 103, 232]. In recent decades, SDP has emerged as a powerful variant of convex optimization, offering a versatile framework for solving optimization problems involving PSD matrices. SDP has found applications in diverse fields, including control theory, signal processing, combinatorial optimization, and quantum information theory [41, 116, 233, 303].

Particularly, in the field of quantum information, SDP has proven to be an indispensable tool for characterizing and manipulating quantum correlations and probabilities [28, 326]. Quantum information theory deals with the fundamental principles governing the representation, transmission, and processing of information in quantum systems. It explores the unique properties of quantum mechanics to develop new paradigms for computation, communication, and cryptography. Quantum correlations, such as entanglement, and the manipulation of probabilities in quantum systems are essential components in designing quantum algorithms and protocols.

This paper aims to provide a comprehensive study of SDP in the context of quantum information. The outline of the paper is as follows. We present first a mathematical framework for convex optimization, covering the necessary preliminaries and notation, and provide a software overview useful for implementing techniques discussed in this work. To aid researchers in the practical implementation of SDP, the paper provides an overview of software tools, solvers, and modeling techniques in section 1.3. It discusses the different solvers available for solving SDP problems, as well as the modeling tools used to formulate and represent optimization problems. Section 1.4 contains a brief overview of the topic of probability distributions occurring in quantum mechanics and other important problems of the theory which can be effectively treated with SDP. The terms introduced there will be used in section 4.

The further discussion in section 2 encompasses sets, spaces, cones, and functions, including important concepts like Fenchel conjugate and subgradient. Duality, a fundamental aspect of optimization, is explored extensively, shedding light on its role in problem formulations and solution methods. The theory of SDP is a focal point of this paper, as it enables the optimization of PSD matrices, which are fundamental objects in quantum information theory. Next, in section 3, the work delves into the definition and characterization of positive semi-definiteness, presenting various formulations of SDP problems, such as the canonical form, the Vandenberghe and Boyd form, the so-called SDP algorithm (SDPA) form, and the Watrous symmetric form. It also discusses the duality of SDP, the treatment of complex variables in semidefinite problems imposing equality and inequality constraints, and the topic of the Schur complement and submatrices. Next, we concentrate on implementing SDPs, to provide a general understanding of the involved numerical methods. The paper also explores how solvers employ IPMs, highlighting their internal mechanisms, such as predictor–corrector methods, warm start strategies, and exploitation of the problem structure.

In the following section 4, the paper introduces basic tools and techniques in SDP that are specifically relevant to quantum information. These tools include semidefinite representations, separability criteria, Choi–Jamiołkowski isomorphism (state-channel duality), the sum of squares decomposition, and Lovász theta. Understanding and utilizing these tools are crucial for solving optimization problems involving quantum states, correlations, and probabilities. A significant portion of the paper focuses on the application of moment matrices in quantum information. Moment matrices play a vital role in capturing the correlations present in quantum systems. The paper explores correlation matrices and moment matrices, their mathematical properties, and their significance in optimization problems involving non-commuting variables. Hierarchical methods, such as the NPA hierarchy, the so-called MLP hierarchy, and the NV hierarchy, are discussed in detail for optimizing probability distributions without or with dimension constraints. Additionally, the SWAP method for self-testing in quantum information is presented.

1.1. Historical notes on optimization

The historical development of LP, can be traced back to the times of a critical need for optimal resource management during World War II. Soon after, in 1947, George Dantzig introduced the simplex method [80], marking a significant milestone in this field. The simplex method operates by starting at a vertex of a convex polytope representing feasible solutions and gradually moving toward its extreme point. It is important to note that although the simplex method algorithm exhibits exponential worst-case complexity, it has demonstrated remarkable efficiency in practical problem-solving scenarios. To understand the underlying reason for the high complexity of the simplex method, it is necessary to examine the specific instances where the algorithm visits every vertex of the feasible region, leading to exponential worst-case complexity. The Klee–Minty problem, formulated in 1970 [170], serves as an illustrative example of such instances. During the 1960s and 1970s, there was an increasing recognition of the significance of computational complexity, fueling the pursuit of efficient algorithms with polynomial time complexity.

In the 1960s, in the realm of nonlinear programming, it became a common practice to transform constrained problems into unconstrained ones through the utilization of the so-called *barrier methods* [103]. By introducing a specialized barrier function, it became possible to delineate a trajectory within the space of optimization variables known as the *central path*, which could be traversed using the well-established Newton's method. However, the prevalence of barrier methods experienced a temporary decline in the 1970s. Meanwhile, in 1979, Khachian introduced the *ellipsoid method*, the first algorithm for LP with polynomial worst-case complexity [168]. Surprisingly, despite its favorable theoretical complexity, the ellipsoid method proved to be exceedingly slow when applied to most practical problems. Consequently, before 1984, two primary methods for LP existed:

- The simplex method, which possessed exponential worst-case complexity but demonstrated practical efficiency.
- The ellipsoid method, exhibited polynomial complexity but was notably inefficient in practice.

The field of optimization has undergone a significant transformation with the introduction of IPMs. Before 1984, IPMs did not hold a prominent position until Karmarkar's groundbreaking paper, *a new polynomial-time algorithm for LP* [165], was published. Notably, it was demonstrated that IPMs were no less efficient than the simplex method for solving practical LP problems. This revelation of IPM's potential sparked what is often referred to as a *revolution in optimization* [334], leading IPMs to be recognized as one of the most significant algorithms of all time. Before 1984, there existed only minimal connections between LP and nonlinear programming. However, it was soon discovered [118] that the IPM was equivalent to a logarithmic barrier method applied to LP. This equivalence enabled the development of a unified framework, based on barrier function methods, for analyzing both linear and nonlinear problems [234].

The next significant advancement in the field of IPM came with the independent work of Alizadeh [8], Nesterov, and Nemirovskii [233, 234] in the late 1980s. They expanded the applicability of IPM to various convex optimization problems. Nesterov and Nemirovskii discovered that the key to utilizing IPM for convex problems lies in knowing a specific barrier function known as a self-concordant barrier [230]. For practical implementation, it is essential that the first and second derivatives of the barrier function can be computed easily. Vandenberghe and Boyd [314] utilized the theory developed by Nesterov and Nemirovskii

to apply the LP method given by Gonzaga and Todd [126] to SDPs. A self-concordant barrier refers to a smooth convex function defined within the interior of a given set. It diverges toward infinity as it approaches the boundary and, along with its derivatives, satisfies certain Lipschitz continuity conditions. Nesterov and Nemirovskii demonstrated that IPM can be applied to any set where such a barrier function can be formulated. Fortunately, a relatively computationally tractable self-concordant barrier is known for SDPs, *viz.* $F(X) \equiv -\ln\det X$. For a comprehensive historical overview of the development and significance of SDP, detailed information can be found in several notable references such as [105, 125, 303, 334]. These works provide an in-depth exploration of SDPs, shedding light on their emergence as a powerful tool in modern optimization. The key property of SDP problems is the fact that they may be efficiently solved numerically using IPM, as sketched further in section 3.8, and at the same time, they can express or approximate a tremendous range of scientific and engineering problems.

1.2. Preliminaries and notation

We now briefly specify the notation used in this work. In some places, the notation used is overloaded, with the same symbols having different meanings. The reason is that the paper covers a variety of different fairy-specialized topics. We decided to keep the established notation characteristic for each of the specializations. We made an effort to ensure that this does not lead to any ambiguity.

1.2.1. Spaces and sets notation. In this work, we denote by $\mathbb{N}$ the set of natural numbers (including 0), $\mathbb{N}_+$ is the set of natural numbers (excluding 0), $\mathbb{R}$ is the set of real number, and $\mathbb{C}$ is the set of complex numbers. The sets of vectors composed of real numbers, non-negative real numbers, and complex numbers, with k elements, are denoted by $\mathbb{R}^k$, $\mathbb{R}^k_+$, and $\mathbb{C}^k$, respectively. The set of real $k \times l$ matrices is denoted by $\mathbb{R}^{k\times l}$ and the set of real $n \times n$ symmetric matrices by $\mathbb{S}^n$. The set of real $n \times n$ symmetric matrices that are PSD or PD (see section 3.1 for the definitions) we denote with $\mathbb{S}^n_+$ or $\mathbb{S}^n_{++}$, $\mathbb{S}^n_{++} \subset \mathbb{S}^n_+ \subset \mathbb{S}^n$. Similarly, the set of complex $k \times l$ matrices is denoted by $\mathbb{C}^{k\times l}$, the set of Hermitian $n \times n$ matrices by $\mathbb{H}^n$, and its subset of PSD or PD matrices by $\mathbb{H}^n_+$ or $\mathbb{H}^n_{++}$, $\mathbb{H}^n_{++} \subset \mathbb{H}^n_+ \subset \mathbb{H}^n$. We refer to the pair of values k and l (for matrices of arbitrary size) or n (for square matrices) as the size of the matrix. For $n \in \mathbb{N}_+$ we denote $[n] \equiv \{1, \ldots, n\}$. The relation $\succeq$ denotes the so-called Löwner's partial order of PSD matrices [181, 197]. For two symmetric or Hermitian matrices A and B, we have $A \succeq B$ when $A - B$ is PSD.

Banach spaces, i.e. vector spaces which are complete with respect to a given norm $||\cdot||$, are denoted with letters $X, Y, \ldots$. Their continuous dual spaces, as defined below, are denoted with starred letters, $X^*, Y^*, \ldots$. The real and complex Hilbert spaces, i.e. vector spaces with an inner (scalar) product $\langle\cdot|\cdot\rangle$ that are Banach spaces with the norm induced by the inner product, are denoted with calligraphic letters, $\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \ldots$. Their continuous dual spaces are denoted by $\mathcal{X}^*, \mathcal{Y}^*, \mathcal{Z}^*, \ldots$. Usually, we consider vector spaces of real or complex matrices with the Frobenius product, defined below in (3), as the inner product; thus the spaces $\mathbb{R}^{k\times 1}$ and $\mathbb{C}^{k\times 1}$ are the ordinary k dimensional real or complex Euclidean spaces, i.e. finite-dimensional Hilbert spaces.

In some cases we use finite sets of symbols as indices of vectors or matrices; this will be particularly useful in the context of moment matrices, see section 4.6. For a set of symbols Σ we use the standard convention of set theory to denote $\mathbb{C}^\Sigma$ ($\mathbb{R}^\Sigma$) the set of all functions from Σ to $\mathbb{C}$ ($\mathbb{R}$). Since there exists a natural isomorphism between $\mathbb{C}^\Sigma$ and $\mathbb{C}^{|\Sigma|}$ ($\mathbb{R}^\Sigma$ and $\mathbb{R}^{|\Sigma|}$), all operations defined for the latter can be easily mapped to relevant operations on the

former, with an arbitrary ordering of the symbols in Σ, that will be treated here as implicit. For a metric space (X,d) we denote by $\mathrm{B}_X(x,r)$ the *closed ball* centered at $x \in X$ with radius r; $\mathrm{B}_X \equiv \mathrm{B}_X(0,1)$ is the unit closed ball.

Consider a Banach space X over the field F. For a linear functional $x^* : X \to F$ we define its norm as $||x^*|| \equiv \sup_{x\in X:||X||\leqslant 1} |x^*(x)|$. We define X^* as the *continuous dual space*, or the *topological dual space*, or simply the *dual space*, i.e. the space of all linear continuous functionals on X with this norm. The weak topology of X is the weakest topology in X in that all elements of X^* are continuous. For X^* we consider also the weak* topology, defined as the weakest topology on X^* for that every element $x \in X$ corresponds to a continuous functional on X^*. We denote the *bidual* spaces $X^{**} \equiv (X^*)^*$. Spaces for that $X = X^{**}$ are called *reflexive*. It can be seen that every finite-dimensional normed space is reflexive.

The action of a conjugate element x^* on an element x, i.e. $x^*(x)$, is further denoted as $\langle x^*, x\rangle$ to make the linearity explicit. This is to be contrasted with the inner (scalar) product $\langle\cdot|\cdot\rangle$ for Hilbert spaces. The Riesz–Fréchet representation theorem [288, p 182][138, p 31] states that every linear continuous functional x^* on a Hilbert space $\mathcal{X}$ can be represented by the inner product with a certain unique element x of $\mathcal{X}$, in the sense that $\forall_{x'\in\mathcal{H}}\langle x^*, x'\rangle = \langle x|x'\rangle$.

To provide the most explicit formulations, we usually denote the elements of a conjugate space with the star symbol *, e.g. $x^* \in \mathcal{X}^*$. The symbol is barely a notation suggesting an element of a conjugate space and has no algebraic meaning. Similarly, we optionally (with no special mathematical meaning) denote with $\cdot$ (dot) a matrix multiplication in places where it allows us to avoid ambiguity, especially to stress the presence of a scalar product of two vectors. Due to the specificity of our topic closely mixing the *explicit* numerical representation of operators as arrays of (numerical) real values with the abstract complex Hilbert formalism of quantum mechanics, we decided to use both this 'dot' notation of the scalar product, and the bra-ket notation, with the latter used in cases not directly related to the computer implementations.

We denote by $\mathsf{L}[\mathcal{X},\mathcal{Y}]$ the set of all linear operators from the Hilbert space $\mathcal{X}$ to the Hilbert space $\mathcal{Y}$. For $\mathcal{X} = \mathbb{C}^n$ and $\mathcal{Y} = \mathbb{C}^m$ this set is isomorphic to the set of complex $m \times n$ matrices, $\mathbb{C}^{m\times n}$. We use the latter whenever our considerations are directly related to computer implementations. For $A \in \mathsf{L}[\mathcal{X},\mathcal{Y}]$ we define the *adjoint operator*, $A^\dagger \in \mathsf{L}[\mathcal{Y},\mathcal{X}]$ as the unique operator satisfying the scalar product relation

$$y^\dagger \cdot Ax = \left(A^\dagger y\right)^\dagger \cdot x, \tag{1}$$

or, in the bra-ket notation, $\langle y|Ax\rangle = \langle A^\dagger y|x\rangle$, for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. We use the abbreviation $\mathsf{L}[\mathcal{X}] \equiv \mathsf{L}[\mathcal{X},\mathcal{X}]$. For real matrix space, the adjoint is the *transposition* denoted by $^\mathrm{T}$; and for complex matrices, it is the *Hermitian conjugate* denoted by †. Thus we denote by $x^\dagger \in \mathcal{X}^*$ the unique linear operator $x^\dagger : \mathcal{X} \to F : x' \mapsto \langle x|x'\rangle$, where $F = \mathbb{C}$ for complex Hilbert spaces; or by $^\mathrm{T}$ with $F = \mathbb{R}$ for real Hilbert spaces. The symbol of † used for real spaces is equivalent to $^\mathrm{T}$. We define the set of all Hermitian operators acting on a complex Euclidean vector space $\mathcal{X}$ as $\mathrm{Herm}[\mathcal{X}] \equiv \left\{X \in \mathsf{L}[\mathcal{X}] : X^\dagger = X\right\}$.

1.2.2. Matrix conventions and the Frobenius product. In this work, we use the MATLAB notation for elements of matrices. We recall a few examples. In this notation the expression $M_{k,l:m}$ means a submatrix consisting of the elements of the matrix M within the row k and with column indices in $\{l, l+1, \ldots, m\}$. The expression $M_{:,k}$ means the vector consisting of the kth column of the matrix M. $M_{k,l}$ refers to the element in kth row and lth column. For a vector v its kth element is v_k. Vectors are represented by one-column matrices. Matrix elements are

numbered from 1. The identity matrix of size d by d is denoted by $\mathbb{1}_d$; in some cases, instead of the size we specify a space $\mathcal{X}$ and then $\mathbb{1}_{\mathcal{X}}$ denotes the identity operator on $\mathcal{X}$. The zero operator and zero matrix for all spaces are denoted with 0. The Kronecker delta is denoted by $\delta_{i,j}$ and is equal 1 for $i=j$ and 0 otherwise. Trace operation is denoted as $\mathrm{Tr}[\cdot]$, and partial trace by $\mathrm{Tr}_{\{s_i\}_i}[\cdot]$, where $\{s_i\}_i$ enumerates the subsystems which are traced out. Similarly, the partial transposition of subsystems $\{s_i\}_i$ is denoted as ${}^{\mathrm{T}_{\{s_i\}_i}}$.

The function $\mathrm{vec}(\cdot)$ defines a vector containing the elements of the given matrix in column-wise order. $\mathrm{Mat}(\cdot)$ is the inverse of this function. For example, we have

$$\mathrm{vec}\left(\begin{bmatrix} a & c \\ b & d \end{bmatrix}\right) = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}. \tag{2}$$

We also use the following standard convention in which upper-case letters denote matrices, and lower-case letters denote vectors of elements of the matrices, e.g. $x = \mathrm{vec}(X) \in \mathbb{R}^{n^2}$ and $X = \mathrm{Mat}(x) \in \mathbb{R}^{n\times n}$. For two matrices $A, B \in \mathbb{R}^{m\times n}$ we define the relation $A \leqslant B$ to hold if and only if $\forall_{i=1,\ldots,m}\forall_{j=1,\ldots,n} A_{i,j} \leqslant B_{i,j}$. We define relations $A < B$, $A \geqslant B$ and $A > B$ in an analogous way. $\mathrm{Diag}[(d_i)_{i\in[n]}]$ is an n by n diagonal matrix with diagonal entries d_i.

The *Frobenius product* of two complex (or real) matrices, $A, B \in \mathbb{C}^{k\times l}$ (or $A, B \in \mathbb{R}^{k\times l}$) is defined as $\mathrm{Tr}(A^\dagger B)$ (or $\mathrm{Tr}(A^{\mathrm{T}} B)$). We follow the convention common in the literature close to implementation issues and usually denote the Frobenius product as $A \bullet B$ [10, 111, 169, 216, 293, 305, 307, 319]. It can be easily shown that

$$A \bullet B \equiv \mathrm{Tr}\left(A^\dagger B\right) = \mathrm{Tr}\left(AB^{\mathrm{T}}\right) = \sum_{i=1,\ldots,kj=1,\ldots,l} A^*_{i,j} B_{i,j} = \mathrm{vec}\,(A)^\dagger \cdot \mathrm{vec}\,B, \tag{3}$$

and similarly for real matrices. Thus, for real matrices, the Frobenius product is the sum of the elements of the element-wise product of entries of two matrices. One can also show that for real symmetric A and real antisymmetric B we have $\mathrm{Tr}(A^{\mathrm{T}}B) = \mathrm{Tr}(AB) = 0$. For real symmetric A we have $A \bullet B = \mathrm{Tr}(A^\dagger B) = \mathrm{Tr}(A^{\mathrm{T}}B) = \mathrm{Tr}(AB)$. The Frobenius product induces a *Frobenius norm* of a matrix, $||\cdot||_{\mathrm{F}}$ defined as $||A||_{\mathrm{F}} = \sqrt{\mathrm{Tr}(A^\dagger A)}$. This norm is called also a Hilbert–Schmidt norm. The Frobenius product is a direct generalization of the vector Euclidean product, as it can be seen from (3), and the $\mathbb{C}^{k\times l}$ ($\mathbb{R}^{k\times l}$) with Frobenius inner product is isomorphic with the Euclidean space $\mathbb{C}^{kl}$ ($\mathbb{R}^{kl}$). For $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ and $B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$, where $A_{11}, B_{11} \in \mathbb{C}^{n_1\times n_1}$, $A_{12}, B_{12} \in \mathbb{C}^{n_1\times n_2}$, $A_{21}, B_{21} \in \mathbb{C}^{n_2\times n_1}$, and $A_{22}, B_{22} \in \mathbb{C}^{n_2\times n_2}$, for some n_1 and n_2, we have

$$\mathrm{Tr}\left(A^\dagger B\right) = \mathrm{Tr}\left(A_{11}^\dagger B_{11}\right) + \mathrm{Tr}\left(A_{12}^\dagger B_{12}\right) + \mathrm{Tr}\left(A_{21}^\dagger B_{21}\right) + \mathrm{Tr}\left(A_{22}^\dagger B_{22}\right). \tag{4}$$

This equality easily generalizes to matrices $A = (A_{r,c})_{r,c}$ and $B = (B_{r,c})_{r,c}$ divided into arbitrary number of blocks, *viz.* $A \bullet B = \sum_{r,c} (A_{r,c} \bullet B_{r,c})$.

1.3. Software overview, usage, and implementation

The basic tool used to find solutions to SDP is SDP solvers. Modeling languages are useful supporting software aiding in formulating SDPs to be passed to a solver. We refer readers to [214] for a comprehensive overview. From the experience of the author, most of the analysis involving SDPs in quantum information is conducted using either Python or Matlab

language. The latter language has two major implementations, *viz.* the software MATLAB from MathWorks [2] and its open-source alternative OCTAVE [90].

A standard choice for an SDP solver among the NPA community (see section 4.6) using Matlab seems to be the SeDuMi solver [291, 292] created by J F Sturm, currently developed and maintained by Imre Pólik and Oleksandr Romanko under the direction of Tamás Terlaky [290]. This solver implements self-dual embedding IPM [66] and was used for instance in [226–228] and other works implementing NPA. Another SDP solver of particular interest in Matlab is SDPT3 solver [310, 312] implemented by Toh, Todd, and Tütüncü. It uses infeasible primal-dual IPM with so-called Nesterov-Todd (NT) and Helmberg-Kojima-Monteiro (HKM) search directions (see section 3.8 for the definition of the search directions). Other examples of SDP solvers include C library for semidefinite programming (CSDP) [35] by Borchers, DSDP [29], and SDPA [112]. The mentioned solvers are freely available, in most cases in open-source form. A very efficient commercial SDP solver is Mosek [1], possible to be used in Python and MATLAB, with a free license for academia. The solver particularly relevant for large problems is semidefinite programming Newton-CG augmented Lagrangian method (SDPNAL) [294, 338] that is implementing a Newton-conjugate gradient (CG) augmented Lagrangian method for SDP [346].

A popular family of solvers is the mentioned SDPA solvers by Fujisawa *et al* [112, 113, 222, 336, 337]. The SDPA solvers are using the so-called Vandenberghe and Boyd form, or the SDPA form, of SDPs, see section 3.2.2. The variants cover Matlab interface (SDPA-M), parallel implementation for large SDPs Semidefinite programming algorithm parallel version (SDPARA), higher precision arithmetics GNU multiple precision arithmetic library (SDPA-GMP), quad-double library (SDPA-QD) and double-double (SDPA-DD) structural sparsity Semidefinite programming algorithm with the positive definite matrix Completion (SDPA-C), see [335] of an overview. Semidefinite programming algorithm (SDPA) solver implements primal-dual IPM with Mehrotra type predictor–corrector, see section 3.9. When deciding solver to use, a performance benchmark should be consulted [210].

Plenty of papers [6, 19, 31, 46, 50, 63–65, 104, 191, 257, 286] uses the Python package NCPOL2SPDA [332] by Peter Wittek, currently under maintenance by Peter J Brown [333]. NCPOL2SDPA implements a framework for global polynomial optimization problems with SDP relaxations. The functionality of particular interest covers the NPA [226, 227, 254] hierarchy for non-commuting operators; Lasserre's hierarchy for commutative polynomials [184]; the *more randomness from the same data* technique [22, 239]; a hierarchy for bilevel polynomial optimization problem [159]; the Moroder's hierarchy [221]; and a hierarchy of sufficient conditions for the steerability of bipartite quantum states [177]. Note that NCPOL2SDPA is not an SDP solver but a modeling toolbox, used to reduce the human effort when formulating SDPs, and it requires a solver to be included separately. Other popular modeling tools for Python are python interface to conic optimization solvers (PICOS) [275] and CVX Python (CVXPY) [5, 84].

Popular modeling toolboxes to be used with Matlab language are yet another linear matrix inequalities processor (YALMIP) [193] and CVX [127, 128, 132]. They allow the use of various solvers, including SeDuMi, SDPT3, and Mosek. YALMIP can be supported with a package QDimSum (symmetric SDP relaxations for qudits systems) [272] that implements the hierarchy [228] using the symmetrization methods [7, 302] to enhance the performance. In appendix B.1 a sample simple execution is given with YALMIP.

An alternative to the mentioned modeling tools for Matlab and Python is the JuMP package ('Julia for Mathematical Programming') [89], which is an open-source modeling language integrated within the Julia programming environment [32]. It allows users to express a wide range of optimization problems, including linear, mixed-integer, quadratic, conic quadratic,

semidefinite, and nonlinear, in a clear and intuitive code format. JuMP allows these formulated problems to be solved using open-source and commercial solvers including CSDP [35], Mosek [1], SCS [241], and SDPA [112].

At this stage, we mention that models in YALMIP and many other modeling languages are interpreted as so-called dual problems [194], discussed in section 3.2. The dual form of SDP is given in (81), where the SDP variable Z is in a *disaggregated* form, i.e. it is a matrix composed of linear combinations of scalar variables. This is to be contrasted with the primal form of SDP (80), where the SDP variable X is treated as a single matrix variable. There are two reasons, why modeling languages prefer the dual form over the primal form. The major reason is that symbolic manipulations are much easier when the variables are disaggregated. The other reason is that it was observed that in many different fields, the dual form is more natural to formulate the problems occurring in them, see e.g. table 1 in section 4.6.

1.4. Basic problems of quantum information

Many useful functions that occur in quantum information belong to the family of the so-called semi-algebraic functions. These functions can be represented using SDP constraints, and thus are particularly relevant for this review. On the other hand, many other functions are not semi-algebraic, like the logarithm function used e.g. in the definitions of such quantities as entropies, including Shannon, quantum, or their relative or conditional variants. It would be beneficial to be able to express them, or at least their approximations as SDPs. It revealed that it is possible when the non-semi-algebraic function is approximated with a polynomial function, for instance with the support of one of the Gauss quadratures, e.g. the Radau quadrature. Recent results allowed the use of SDP to approximate the matrix logarithm function [100], and as a result, the use of SDP to efficiently optimize expressions on various entropies [98]. One recent article [47] used these methods to determine the lower bounds of the conditional von Neumann entropy certified in a device-independent approach using an extended NPA method [254] using the NCPOL2SDPA tool [332]. We provide a brief overview of the theory of semidefinite representations of semi-algebraic functions in section 4.1.

An important problem in the investigation of the properties of quantum states is to determine whether they are separable or not. An N-partite state ρ is called separable when it is written as a convex combination of product states, *viz.* [327]:

$$\rho = \sum_i p_i \rho_1^i \otimes \cdots \otimes \rho_N^i. \tag{5}$$

Determining whether a given state is separable or entangled based solely on the definition is a challenging task in practice. Thus, the so-called separability problem emerges as one of the fundamental issues in the study of entanglement. The famous Peres–Horodecki PPT criterion [151, 250] provided a necessary condition for separability of states and says that if a bipartite state $\rho_{\mathcal{AB}}$ is separable, then $\rho_{\mathcal{AB}}^{\mathrm{T}_B} \succeq 0$. Another attempt at this issue was [188], where a constructive algorithm that enables the identification of the optimal separable approximation for any density matrix associated with a finite-dimensional composite quantum system was presented. The method established a condition for separability and provided a measure of entanglement. An important connection of the separability problem with SDP was the so-called DPS method given in [86, 87], which may be considered as a direct development of the PPT criterion, providing a hierarchy of SDP approximations discussed in section 4.2.

Another notion is that of quantum channels [329]. A quantum channel is a communication channel that transmits quantum information. Quantum channels describe any form of state

evolution either in time or space governed by quantum mechanics. There are multiple different issues to be studied regarding quantum channels. One of the main research problems related to them is the characterization and classification of different types of channels, like depolarizing or amplitude-damping channels. Another research problem related to quantum channels is the development of methods for channel estimation and tomography, or the study of noisy and imperfect channels. In practice, all communication channels are subject to noise and imperfections that can degrade the quality of transmitted quantum states. Therefore, it is essential to develop methods for mitigating the effects of noise and imperfections on quantum communication. A basic tool used in modeling quantum channels, or more general maps linear $\mathsf{L}[\mathcal{H}_1, \mathcal{H}_2]$, for Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, is the Choi–Jamiołkowski isomorphism discussed in section 4.3.

The Tsirelson bound also referred to as the Cirel'son bound, is a concept in quantum mechanics that holds significance in the investigation of quantum non-locality. In essence, the Tsirelson bound establishes a maximum level of correlation achievable between two or more distant quantum systems. The first examples of such bounds were derived by Boris Tsirelson in 1980 [72]. Tsirelson bounds carry profound implications for our comprehension of quantum mechanics and its practical applications. In particular, if the correlations violate the Tsirelson bound, it implies that quantum physics cannot reproduce them. This observation significantly contributes to our understanding of entanglement. The Tsirelson bound finds practical utility in various applications of quantum information theory, including quantum cryptography and quantum teleportation. For instance, it enables the quantification of the requisite and attainable level of entanglement for secure communication through quantum cryptography. In section 4.4 we briefly describe the SoS technique and then show an example of how can it be applied to the derivation of the Tsirelson bound.

A fundamental concept, closely related to the Tsirelson bound is quantum contextuality. It refers to the property of quantum systems where the outcome of a measurement depends on the context in which it is measured. In other words, the value of a quantum property is not determined by the property itself, but by the other properties with which it is measured. This means that the same quantum system can exhibit different properties depending on how it is measured, and this property has been shown to be essential for many quantum information processing tasks. One of the approaches to the analysis of contextuality was given in [57, 58] where a relationship with the so-called Lovász theta has been established. The method revealed to be very profound [56, 133, 152, 186, 268, 304]. Here, in section 4.5 we describe the SDP methods for Lovász theta and show how to relate it to contextuality.

We will now briefly review the topic of Bell inequalities and Bell functionals [25], as this will be needed in many places in this work, especially for section 4.6. Bell inequalities are mathematical expressions that set a limit on certain probabilities, which cannot be violated in a classical physics framework but can be exceeded in quantum mechanics. The violation of Bell inequalities can be observed through experimental measurements, providing conclusive evidence that the behavior of the world cannot be explained solely by classical physics. Such groundbreaking experiments were conducted in the 1980s by Aspect *et al* [16–18]. A Bell experiment involves two or more separate parties who share a quantum state and perform measurements with different settings, without any form of communication between them. By conducting a series of such measurements and analyzing the collected data, it becomes possible to estimate the set of joint conditional probability distributions, or *behaviors*, $\{P(a,b|x,y)\}$ of the outcomes conditioned on the settings. A bipartite Bell functional is a linear functional that operates on behaviors over two parties or subsystems of the form $\sum_{a,b,x,y} \alpha_{a,b,x,y} P(a,b|x,y)$, where $\alpha_{a,b,x,y} \in \mathbb{R}$.

Classical devices can be described using the following elements. The class $\mathfrak{L}$ represents local behaviors that are in accordance with classical physics. The statistical description, i.e. the behavior, of the pair of devices is of the form

$$P(a,b|x,y) = \sum_{\lambda} P(\lambda) \cdot P_{A|X,\Lambda}(a|x,\lambda) \cdot P_{B|Y,\Lambda}(b|y,\lambda). \tag{6}$$

Here, $P(\lambda)$ represents the probability of observing the hidden state λ, and $P_{A|X,\Lambda}(a|x,\lambda)$ and $P_{B|Y,\Lambda}(b|y,\lambda)$ correspond to the conditional probabilities of obtaining results a for Alice and b for Bob, respectively, given their respective settings x and y, and the hidden state λ. $P_{\Lambda}(\lambda)$ refers to the probability distribution of hidden internal states, where λ represents a specific state and $\sum_{\lambda\in\Lambda} P_{\Lambda}(\lambda) = 1$ ensures normalization.

Next, the non-signaling devices can be characterized as follows. The class $\mathfrak{N}$ denotes non-signaling behaviors, which align with the principles of relativistic physics. $P_{A|X}(a|x)$ and $P_{B|Y}(b|y)$ represent the marginal conditional probability distributions of Alice and Bob, respectively. These distributions are derived from the behavior $\{P(a,b|x,y)\}$, and they satisfy the conditions

$$\sum_{b\in B} P(a,b|x,y) = P_{A|X,Y}(a|x,y) = P_{A|X}(a|x), \text{ and} \tag{7a}$$

$$\sum_{a\in A} P(a,b|x,y) = P_{B|X,Y}(b|x,y) = P_{B|Y}(b|y). \tag{7b}$$

These conditions ensure the consistency of the marginal distributions regardless of the settings of the other party. Non-signaling property implies that the settings chosen by one party do not have any influence on the marginal distribution observed by the other party. By considering these elements and properties, we can analyze the behavior of non-signaling devices in the context of bipartite systems. Optimization over these sets can be performed using LP.

The class $\mathfrak{Q}$ contains all behaviors that adhere to the fundamental principles of quantum physics. It is noteworthy that the class of local behaviors $\mathfrak{L}$, forms a subset of quantum behaviors, i.e. $\mathfrak{L} \subset \mathfrak{Q}$, and $\mathfrak{Q} \subset \mathfrak{N}$. A Bell functional $\mathcal{I}$ can exhibit a characteristic where its maximum value allowed on the set $\mathfrak{Q}$, denoted as I_{Q}, is strictly greater than its maximum value on the set $\mathfrak{L}$, denoted as I_{L}. The existence of such functionals is a consequence of Bell's theorem [26]. A *Bell inequality* is a statement $\mathcal{I} \leqslant I_{\mathrm{L}}$ that sets a limit on the value of this operator within the framework of local theories. We say that a Bell inequality is violated if, for a given behavior $\{P(a,b|x,y)\}$, we have $\mathcal{I} > I_{\mathrm{L}}$. The task of optimization over $\mathfrak{Q}$, in particular Bell functionals, is NP-hard, as shown by Kempe *et al* in 2008 at FOCS [166].

The statement $\{P(a,b|x,y)\} \in \mathfrak{Q}$ is true if and only if the following conditions are satisfied, involving the existence of a (finite or infinite dimensional) Hilbert space $\mathcal{H}$, a state (vector) $|\psi\rangle$ on $\mathcal{H}$, and a set of operators (measurements) $\{E_x^a, F^b{}_y\}_{a,b,x,y}$ on $\mathcal{H}$ such that:

(i) The operators E_x^a and $F^b{}_y$ are projectors. From this property, it follows that the operators correspond to observable quantities with non-negative probabilities.
(ii) Different results with the same setting, represented by E_x^a and $E_x^{a'}$, are orthogonal to each other, given by $E_x^a E_x^{a'} = 0$; similarly for Bob's measurements $F^b{}_y$ and $F_y^{b'}$. This orthogonality condition signifies that different measurement outcomes are mutually exclusive.
(iii) The sum of all operators E_x^a for a fixed x equals the identity operator $\mathbb{1}$, denoted as $\sum_a E_x^a = \mathbb{1}$. Similarly, the sum of all operators $F^b{}_y$ for a fixed y is equal to $\mathbb{1}$, expressed as $\sum_b F^b{}_y = \mathbb{1}$. These normalization conditions ensure that the probabilities of all possible outcomes sum up to 1.

(iv) The operators representing measurements for Alice, E_x^a, and those for Bob, $F^b{}_y$, commute with each other, denoted as $[E_x^a, F^b{}_y] = 0$. This commutation property indicates that the order of measurements performed by Alice and Bob does not affect the results.

(v) The joint probability distribution $P(a,b|x,y)$ can be expressed as the expectation value of the operators E_x^a and $F^b{}_y$ acting on the state $|\psi\rangle$, given by

$$P(a,b|x,y) = \langle\psi|E_x^a F^b{}_y|\psi\rangle. \tag{8}$$

This equation illustrates that the probabilities arise from performing measurements on the quantum state $|\psi\rangle$.

We note that the quantum measurements can be represented not necessarily by projectors as in the above condition (i), but by a more general set of operators called POVMs. A POVM $\{M^a\}_a$ satisfies the PSD condition $\forall_a M^a \succeq 0$ and the normalization condition $\sum_a M^a = \mathbb{1}$. Since the vector $|\psi\rangle$ can be of arbitrarily high, possibly infinite, dimension, and by Stinespring's dilation theorem [289] any POVM on a specific Hilbert space can be represented by a projective measurement within a sufficiently highly dimensional Hilbert space and any mixed state can be represented as a subsystem originating from a system in a pure state in that Hilbert space, the restriction to projectors in (i) and pure states do not lead to a loss in generality.

From the normalization conditions (iii) we see that for any fixed x (y) any of the operators $\{E_x^a\}_{a,x}$ ($\{F^b{}_y\}_{b,y}$), can be expressed using the rest of them and the identity operator. Thus instead of the full set $\{E_x^a, F^b{}_y\}_{a,b,x,y}$ we can equivalently require existence of the so-called *reduced set of operators* $\{\mathbb{1}, E_x^{\tilde{a}}, F_y^{\tilde{b}}\}_{\tilde{a},\tilde{b},x,y}$, where the index $\tilde{a}$ ($\tilde{b}$) covers all values excluding the last one. For instance, consider a scenario involving two parties, each with two settings from the set $\{1,2\}$, and obtaining two outcomes from the set $\{0,1\}$. The reduced set of operators in this case is

$$\left\{\mathbb{1}, E_1^0, E_2^0, F_1^0, F_2^0\right\}. \tag{9}$$

The schoolbook formulation of quantum mechanics involves the notion of Hilbert spaces and the properties of operators acting over them. In contrast, an effort was made to derive, at least partially, equivalent physical consequences from a direct axiomatics [51, 69, 74, 136, 137, 148, 156, 201], or principles explicitly basing on information theory. The prominent examples of such information-theoretic principles include the non-signaling [255], the non-trivial communication complexity [43], the no advantage for nonlocal computation [192], the information causality [249], the local orthogonality [106], the information content of systems [79]. Of particular interest in this work are the macroscopic locality [229] and the almost quantum set of behaviors discussed in section 4.6.1.

Now, we provide a concise overview of the key aspects pertaining to dimension-bounded scenarios [53, 114, 190, 248]. We will discuss them in detail in sections 4.7 and 4.8. Alice and Bob are assigned random inputs, x and y. Subsequently, Alice sends a message to Bob based on her input, where the message takes the form of a quantum state ρ_x of a specific dimension d. Bob receives the quantum state and performs a measurement $\{M_y^b\}_b$ on it, yielding a result b. This leads to a behavior $\{P_d(b|x,y)\}$ within a prepare-and-measure scheme, $P_d(b|x,y) = \mathrm{Tr}\left(\rho_x M_y^b\right)$. We assume the absence of entanglement between Alice and Bob in this context. Let $\mathcal{P}_d \equiv \{\{P_d(b|x,y)\}_{\{\rho_x\}_x,\{\{M_y^b\}_b\}_y},\}$ be the set of all probabilities of the discussed for in the given dimension d. We note that we have $\mathcal{P}_d \subseteq \mathcal{P}_{d+1}$, since increasing the dimension of the

communicated state we can send at most the same amount of data. A dimension witness W is a linear function of behaviors, i.e. it has the following form:

$$\sum_{b,x,y} \beta_{b,x,y} P(b|x,y). \tag{10}$$

The key property of dimension witnesses is that they allow us to distinguish the dimensions for which inclusion is strict. Using the definition of probability distributions in the prepare-and-measure scheme, we may introduce a notion of dimension witnesses which is analogous to the concept of Bell functionals.

2. Mathematical framework of optimization

A general *static optimization* problem [200], or optimization in finite-dimensional spaces, is a task of determining the values of a certain variable $x \in \mathcal{F} \subseteq \mathbb{R}^n$, called the *decision variable*, for which a given function $f_0 : \mathcal{F} \to \mathbb{R}$, called *target*, attains its minimum; these points are called *minimizers*, and their set is called the *optimal set* [24]. The whole set $\mathcal{F}$ is called the *feasible set* (and other names like *feasible region*, or *solution space*, or *search space*, are often used) [23]. A point $x_0 \in \mathcal{F}$ with the property that for all $x_1 \in \mathcal{F}$ it holds $f(x_1) \geqslant f(x_0)$ is called a *global minimum*. A point $x_0 \in \mathcal{F}$ for which there exists a neighborhood (in metric space sense) $\mathcal{N}$ such that $x_1 \in \mathcal{N} \cap \mathcal{F} \implies f_0(x_1) \geqslant f_0(x_0)$ is called a *local minimum*. Every global minimum is also a local minimum.

The further part of this section covers the crucial topic of duality in optimization. In section 2.4 we discuss general optimization in Banach spaces, covering essential techniques and concepts. Next, in section 2.5 we explore the Fenchel–Rockafellar scheme, delving into strong duality and constraint qualification, which are fundamental principles in optimization. Then, in section 2.6 we discuss an alternative, but less general, way of construction of dual problems, *viz.* the Lagrangian scheme. Lastly, in section 2.7 we delve into the more specific case of convex cone optimization and show how both dualization schemes apply to it.

2.1. Convex and linear programming

Now, let us specify what we mean by *convex optimization* problems. In simple words, these are tasks of minimization of a convex function over a convex set [24, 41]. To be more specific, the so-called *functional form* of convex problems is defined as follows. Let $m, n \in \mathbb{N}$. The commonly used general form of convex problems is the following:

$$\begin{aligned} &\text{minimize } f_0(x) \\ &\text{subject to } f_i(x) \leqslant b_i, i = 1, \ldots, m, \end{aligned} \tag{11}$$

where $f_0, \ldots, f_m : \mathbb{R}^n \to \mathbb{R}$ are convex function, i.e. for any $x_0, x_1 \in \mathbb{R}^n$ and $\lambda \in [0,1]$ they satisfy the so-called Jensen's inequality:

$$f_i(\lambda x_0 + (1-\lambda) x_1) \leqslant \lambda f_i(x_0) + (1-\lambda) f_i(x_1). \tag{12}$$

If we replace $\leqslant$ with $<$ in (12), we say that f_i is *strictly convex*. The set of points satisfying the constraints of (11), i.e. the feasible set $\mathcal{F} = \bigcap_{i=1}^m \{x \in \mathbb{R}^n : f_i(x) \leqslant b_i\} \subseteq \mathbb{R}^n$, is a convex set. A crucial property of convex programs is that all their local minimum points are also global minimum points. What is more, the optimal set for a convex problem is also a convex set.

If f_0 is strictly convex then there exists at most one global minimum. The *optimal value* or, when there is no ambiguity, simply the *value*, of an optimization problem corresponds to the optimal objective function value, representing the optimal outcome attainable for the objective function while adhering to all constraints. The *solution* to an optimization problem refers to the collection of decision variables that yield the optimal objective function value. This solution must satisfy all constraints imposed by the problem. It is worth noting that in certain scenarios, multiple optimal solutions can exist, indicating that various sets of decision variables yield the same optimal objective function value.

The terms *optimization* and *optimization problem* are often used interchangeably in the literature to refer to the task of finding the optimal (usually minimal, as above) value of an objective function. However, the terms have a subtle difference in their meaning. Optimization typically refers to the act of determining the optimal value itself. The optimization problem encompasses not only finding the optimal value but also the associated decision variables or parameters that achieve that optimal value, i.e. the solution. Thus, we distinguish the minimization problems where the task is to find both the value $\min_{x\in\mathcal{F}}[f_0(x)]$ and the solution $\operatorname{argmin}_{x\in\mathcal{F}}[f_0(x)]$, from the minimization, i.e. the task of finding the infimum $\inf_{x\in\mathcal{F}}[f_0(x)]$. Note that the infimum may not even be attained, whereas for the minimum there always exists a solution attaining it.

Particular examples of convex optimization problems are LPs and SDPs, being the main topic of this Review. We start with the formulation of LP. Let $m,n\in\mathbb{N}$, and $m\leqslant n$. The so-called primal canonical form of LP is the following optimization task in variable x:

$$\begin{aligned}&\text{minimize } c^{\mathrm{T}}\cdot x\\ &\text{subject to } Ax=b, x\geqslant 0,\end{aligned}\tag{13}$$

where $A\in\mathbb{R}^{n\times m}$, $b\in\mathbb{R}^m$, $x,c\in\mathbb{R}^n$. The dual problem of LP is

$$\begin{aligned}&\text{maximize } b^{\mathrm{T}}\cdot y\\ &\text{subject to } c-A^{\mathrm{T}}y=z,\\ &\qquad\qquad z\geqslant 0.\end{aligned}\tag{14}$$

In the above problems, the variable x is called the *primal variable*, y the *dual variable*, z the *dual slack variable*, A is the *linear constraint matrix*, b is the *right-hand side (RHS) of the linear constraint*, and c is the *linear coefficient*.

2.2. Sets and cones definitions

Let X be a Banach space over an ordered field F.

Consider a set $C\subseteq X$. The *core* of C is the set of all points in C such that for any direction d in X there exist $T_d>0$ such that for all $t\in[0,T_d]$ we have $x+td\in C$, *viz.*:

$$\operatorname{core} C\equiv\left\{x\in C:\forall_{\substack{d\in X,\\ ||d||=1}}\ \exists_{T_d>0}\forall_{t\in[0,T_d]}\,(x+td)\in C\right\}.\tag{15}$$

Contrast it with the *interior* of C defined as:

$$\operatorname{int} C\equiv\left\{x\in C:\exists_{T>0}\forall_{\substack{d\in X,\\ ||d||=1}}\ \forall_{t\in[0,T]}\,(x+td)\in C\right\}.\tag{16}$$

It is easy to see that $\operatorname{int}C\subseteq\operatorname{core}C$.

C is *convex* if

$$\forall_{x_1,x_2\in C}\forall\lambda\in[0,1]\lambda x_1+(1-\lambda)x_2\in C. \tag{17}$$

In particular, $\emptyset$ is a convex set. C is *absorbing* [36, p 244] if it is convex and $X=\bigcup_{t\geqslant 0}tC$. Obviously, $0\in \text{core}C$ for absorbing C.

A subset $K\subseteq X$ is called a *cone*, or *nonnegative homogeneous*, if and only if [342]:

$$x\in K,\lambda\in F_+ \implies \lambda x\in K, \tag{18}$$

where F_+ is the set of all non-negative scalars of F. Thus, the cone is simply the set invariant under multiplication by non-negative scalars. The cone K is *convex* if and only if [41]

$$x_1,x_2\in K \implies x_1+x_2\in K, \tag{19}$$

that can be intuitively understood as $K+K\subseteq K$.

For an arbitrary, *not* necessarily being a cone, subset $K\subseteq X$, the *topological dual cone*, or simply the *dual cone*, is defined as [39, p 16]:

$$K^*\equiv\{x^*\in X^*: x'\in K \implies \langle x^*,x'\rangle\geqslant 0\}. \tag{20}$$

The set K^* defined by (20) is always a convex cone. If X is also a Hilbert space, then by the Riesz–Fréchet representation theorem, the definition (20) is equivalent to:

$$K^*\equiv\{x\in X: x'\in K \implies \langle x|x'\rangle\geqslant 0\}. \tag{21}$$

If $K=K^*$, then K is called a *self-dual cone*. Examples of a self-dual cone for $m\in\mathbb{N}_+$ are: the *positive orthant cone* of $\mathcal{X}=\mathbb{R}^m$:

$$K_+^m\equiv\{x\in\mathbb{R}^m: x_1\geqslant 0,\ldots,x_m\geqslant 0\}=\mathbb{R}_+^m, \tag{22}$$

the *Lorentz* (or *second order*, or *quadratic*) cone:

$$K_q^m\equiv\left\{(x,t)\in\mathbb{R}^{m+1}: ||x||_2\leqslant t\right\}, \tag{23}$$

and the PSD cone $\mathbb{S}_+^n$ discussed further in section 3.1.

A set $\mathcal{S}\subseteq\mathbb{R}^n$ is called a basic closed semialgebraic set if and only if there exist a set of polynomials $\{f_i\}_{i\in[m]}$, $f_i:\mathbb{R}^n\to\mathbb{R}$, such that $\mathcal{S}=\left\{x\in\mathbb{R}^n:\forall_{i\in[m]}f_i(x)\geqslant 0\right\}$. Similarly, it is called a basic open semialgebraic set if and only if $\mathcal{S}=\left\{x\in\mathbb{R}^n:\forall_{i\in[m]}f_i(x)>0\right\}$. Compare this with algebraic sets which have the form $\mathcal{S}=\left\{x\in\mathbb{R}^n:\forall_{i\in[m]}f_i(x)=0\right\}$. A set $\mathcal{S}$ is called semi-algebraic if there exists a set $\{\mathcal{S}_{i,j}\}_{i\in[k],j\in[r_i]}$ for some $k,r_i\in\mathbb{N}_+$ such that each $\mathcal{S}_i$ is a basic closed semialgebraic set or a basic open semialgebraic set or an algebraic set and $\mathcal{S}=\bigcup_{i\in[k]}\bigcap_{j\in[r_i]}\mathcal{S}_{i,j}$. Any algebraic set is obviously semialgebraic. A consequence of the famous Tarski–Seidenberg principle [282, 297] is the fact that the set of semialgebraic sets is closed under projections, i.e. if $S\in\mathbb{R}^{n_1+n_2}$ is a semialgebraic set, then also its projection onto the first n_1 coordinates is a semialgebraic set. We refer to chapter 2 of [34] for a detailed discussion of semialgebraic sets. The positive orthant, Lorentz, and PSD cone are basic closed semialgebraic sets; the same holds for polyhedra and spectrahedra discussed in section 4.1.

2.3. Convex analysis: functions, convex conjugate, and Fenchel–Rockafellar theorem

Consider an arbitrary function $f: X \to \mathbb{R} \cup \{-\infty, +\infty\}$. The function is defined to be *proper* if it never takes the value $-\infty$, and is not identically equal to $+\infty$. The *epigraph* of f is defined as

$$\text{epi } f \equiv \{(x,r) \in X \times \mathbb{R} : r \geqslant f(x)\}, \tag{24}$$

so it is the set of all points above the graph of the function. The hypograph of f is hyp$f \equiv \{(x,r) \in X \times \mathbb{R} : f(x) \geqslant r\}$. One usually formulates the definition of a convex function in terms of Jensen's inequality (12); this is the convexity in *analytical* sense. Alternatively, epigraph allows to provide a geometric sense of convexity, *viz.* f is defined to be *convex* if epi$f$ is convex, and *concave* if $-f$ is convex; see [198, p 12] for a discussion. The *effective domain* of f is defined as

$$\text{dom } f \equiv \{x \in X : f(x) < +\infty\}. \tag{25}$$

f is defined to be *lower semi-continuous* (lsc) if epi f is a closed subset of $X \times \mathbb{R}$. The set cont f is the set of all points where f is finite and continuous.

Let $F: X \times Y \to \mathbb{R} \cup \{-\infty, +\infty\}$, with Y a linear spaces, be convex in both parameters. Let $C \subseteq X$ be a non-empty convex set. Then [41, p 88], the function

$$\theta(y) = \inf_{x \in C} F(x,y) \tag{26}$$

is convex in y, as long as

$$\forall_{y \in Y} \theta(y) \neq -\infty. \tag{27}$$

The epigraph of θ is epi $\theta = \{(x,t) : \exists_{y \in Y} (x,y,t) \in \text{epi } F\}$, and is convex, as a projection of a convex set epi F. Indeed, let $y_0, y_1 \in \text{dom } \theta$. Then

$$\forall_{\epsilon > 0} \exists_{x_0, x_1 \in C} F(x_0,y_0) \leqslant \theta(y_0) + \epsilon \text{ and } F(x_1,y_1) \leqslant \theta(y_1) + \epsilon, \tag{28}$$

and for any $\lambda \in [0,1]$ we have

$$\begin{aligned} \theta(\lambda y_0 + (1-\lambda) y_1) &= \inf_{x \in C} F(x, \lambda y_0 + (1-\lambda) y_1) \\ &\leqslant F(\lambda x_0 + (1-\lambda) x_1, \lambda y_0 + (1-\lambda) y_1) \leqslant \lambda F(x_0,y_0) + (1-\lambda) F(x_1,y_1) \\ &\leqslant \lambda \theta(y_0) + (1-\lambda) \theta(y_1) + \epsilon. \end{aligned} \tag{29}$$

Since ϵ can be arbitrarily small, Jensen's inequality (12) for θ follows.

A well-known operation of the holomorphic functional calculus is the extension of a function defined on real values to Hermitian matrices. This extension allows for the evaluation of functions that are not originally defined on matrices but can be extended to them through the use of complex analysis techniques. Any Hermitian matrix $H \in \mathbb{H}^n$ can be diagonalized by a unitary matrix U, so that $H = U \cdot \text{Diag}[(d_i)_{i \in [n]}] \cdot U^\dagger$. A function $f: \mathbb{R} \to \mathbb{R}$ can be applied to the eigenvalues, and thus the function can then be extended to the entire matrix as $f(H) \equiv U \cdot \text{Diag}[(f(d_i))_{i \in [n]}] \cdot U^\dagger$. We say that f is an *operator monotone* (or matrix monotone) when for any $M_1, M_2 \in \mathbb{H}^n$, from $M_1 \succeq M_2$ it follows that $f(M_1) \succeq f(M_2)$. Next, we say that f is *operator convex* (or matrix convex) if it satisfies Jensen's inequality (12) in Löwner's partial order, i.e. $\lambda f(M_1) + (1-\lambda) f(M_2) \succeq f(\lambda M_1 + (1-\lambda) M_2)$ for all $\lambda \in [0,1]$ [134]. Finally, we

say that f is *operator concave* (or matrix concave) when $-f$ is operator convex [4, 60]. The *matrix epigraph* of f is $\mathbf{epi}\, f \equiv \left\{(X,R) \in \mathbb{H}^n_{++} \times \mathbb{H}^n : R \succeq f(X)\right\}$, and the *matrix hypograph* of f is $\mathbf{hyp}\, f \equiv \left\{(X,R) \in \mathbb{H}^n_{++} \times \mathbb{H}^n : f(X) \succeq R\right\}$.

The (commutative) *perspective* of a function $f : \mathbb{R}^n \to \mathbb{R}$ is the function $P_f : \mathbb{R}^{n+1} \to \mathbb{R}$ defined as $P_f(x,t) \equiv t f(x \cdot t^{-1})$ with $\mathrm{dom}\, P_f = \{(x,t) : x/t \in \mathrm{dom}\, f, t > 0\}$. The operation of perspective preserves the convexity, i.e. if f is convex then P_f is convex [41, p 89]. If $M_1, M_2 \in \mathbb{H}^n$ commute, then $P_f(M_1,M_2)$ is well-defined by extending P_f to matrices [93]. To cover also the non-commutative case, the *non-commutative perspective* of an operator convex function is defined as the unique extension of the corresponding (commutative) perspective that preserves homogeneity and convexity [92]. The formula for non-commutative perspective is $\mathbf{P}_f[M_1,M_2] \equiv M_2^{1/2} \cdot f\left(M_2^{-1/2} M_1 M_2^{-1/2}\right) \cdot M_2^{1/2}$, with $\mathrm{dom}\, \mathbf{P}_f = \mathbb{H}^n \times \mathbb{H}^n_+$ [91]. For instance, the non-commutative perspective of the negative logarithm function is the operator relative entropy [107–110], *viz.*

$$\begin{aligned} S(M_1|M_2) &= M_2^{1/2} \cdot \eta\left[M_2^{-1/2} M_1 M_2^{-1/2}\right] \cdot M_2^{1/2} = \mathbf{P}_{(-x\log x)}[M_1,M_2] \\ &= -M_1^{1/2} \cdot \log\left(M_1^{-1/2} M_2 M_1^{-1/2}\right) \cdot M_1^{1/2} = \mathbf{P}_{(-\log)}[M_2,M_1] \end{aligned} \tag{30}$$

for $\eta(x) \equiv -x\log x$ and invertible M_1 and M_2 [109].

In [14, 261] the notion of the *matrix geometric mean* $M_1 \# M_2 \equiv M_1^{1/2} \cdot [M_1^{-1/2} M_2 M_1^{-1/2}]^{1/2} \cdot M_1^{1/2}$, which satisfy certain general properties [182], was introduced for PD M_1 and M_2. Its direct generalization is the so-called t-weighted matrix geometric mean

$$M_1 \#_t M_2 \equiv \mathbf{P}_{(x \mapsto x^t)}[M_1,M_2] = M_1^{1/2} \cdot \left[M_1^{-1/2} M_2 M_1^{-1/2}\right]^t \cdot M_1^{1/2}, \tag{31}$$

and thus $M_1 \# M_2 = M_1 \#_{1/2} M_2$ [33, 274]. It can be shown that t-weighted matrix geometric mean is operator concave for $t \in [0,1]$, and operator convex for $t \in [-1,0] \cup [1,2]$ [33].

The *indicator* function of C is defined as

$$I_C[x] \equiv \begin{cases} 0 & \text{if } x \in C, \\ +\infty & \text{otherwise.} \end{cases} \tag{32}$$

The indicator function $I_C[x]$ is convex if and only if, the set C is convex. It can also be shown that the indicator function is lower (upper) semi-continuous if and only if, C is closed (open).

The mean value of a random variable x we denote as $\langle x \rangle$, and the standard deviation as $\boldsymbol{\sigma}[x]$. The covariance between random variables x_i and x_j is defined as $\mathrm{cov}[x_i,x_j] \equiv \langle (x_i - \langle x_i \rangle) \cdot (x_i - \langle x_i \rangle) \rangle$, and their correlation is defined as $\mathrm{corr}[x_i,x_j] \equiv \mathrm{cov}[x_i,x_j]/(\boldsymbol{\sigma}[x_i] \cdot \boldsymbol{\sigma}[x_j])$. The variance of x is the covariance of the variable with itself, $\mathrm{var}[x] \equiv \mathrm{cov}[x,x] \geqslant 0$.

2.3.1. Fenchel conjugate. For an arbitrary function $f : X \to \mathbb{R} \cup \{+\infty\}$, where X is a Banach space, the *Fenchel conjugate* [102], or *convex conjugate*, or *Legendre transform*, or *Legendre–Fenchel transform*, or simply the conjugate, being the basic operation in convex analysis, is defined as

$$f^* : X^* \to \mathbb{R} \cup \{-\infty, +\infty\} : x^* \mapsto \sup_{x \in X} \{\langle x^*, x \rangle - f(x)\}. \tag{33}$$

The conjugate operation can be applied multiple times. For instance the function $(f^*)^*$ is defined on X^{**}. With an abuse of notation by f^{**} we denote the restriction of $(f^*)^*$ to X, so that [198, p 83]:

$$f^{**}: X \to \mathbb{R} \cup \{-\infty, +\infty\} : x \mapsto \sup_{x^* \in X^*} \{\langle x^*, x\rangle - f^*(x^*)\}. \tag{34}$$

It holds

$$\operatorname{epi} f^* = \bigcap_{x \in X} \operatorname{epi}\{\langle \cdot, x\rangle - f(x)\}. \tag{35}$$

Since for any $x \in X$ the set $\operatorname{epi}\{\langle \cdot, x\rangle - f(x)\}$ is convex and closed in weak* topology on $X^* \times \mathbb{R}$ (and the natural topology on $\mathbb{R}$), we have that f^* is always a convex and lsc function in that topology, no matter on the form of f. It is trivial to see that $f_1 \leqslant f_2 \implies f_1^* \geqslant f_2^*$ and $f^*(0) = -\inf_{x \in X} f(x)$ [198, p 82].

A direct consequence of the definition (33) is for any $f: X \to \mathbb{R} \cup \{-\infty, +\infty\}$ it holds [138, p 184]:

$$f \equiv +\infty \iff f^* \equiv -\infty \iff -\infty \in f^*(X), \tag{36}$$

and, in particular if f^* is proper, then also f is proper. From $\forall_{x \in X} \forall_{x^* \in X^*} f(x) \geqslant \langle x^*, x\rangle - f^*(x^*)$ taking supremum over x^* we get $\forall_{x \in X} f^{**}(x) \leqslant f(x)$, i.e.

$$f^{**} \leqslant f. \tag{37}$$

The inequality (37), together with the convexity and lsc properties of the Fenchel conjugate, motivate to call f^{**} a *regularization*, or a *convex lsc relaxation*. If f itself is convex and lsc, and there exist $x^* \in X^*$ and $\alpha \in \mathbb{R}$ such that $f \geqslant \langle x^*, \cdot\rangle + \alpha$ then the *Fenchel–Moreau theorem* states the equality,

$$f = f^{**}, \tag{38}$$

see [198, p 84] for the proof.

2.3.2. Subgradient and Fenchel–Rockafellar theorem. Consider an arbitrary function $f: X \to \mathbb{R} \cup \{+\infty\}$. An element $x^* \in X^*$ is called a *subgradient* of f at $x \in \operatorname{dom} f$ when

$$\forall_{x' \in X} f(x') - f(x) \geqslant \langle x^*, x' - x\rangle. \tag{39}$$

The (possibly empty) set of all subgradients of f at $x \in \operatorname{dom} f$ is called *subdifferential* and denoted by $\partial f(x)$. Directly from the definitions of subgradient and convex conjugate, it follows that

$$\forall_{x \in \operatorname{dom} f \subseteq X} \forall_{x^* \in X^*} x^* \in \partial f(x) \iff f(x) + f^*(x^*) = \langle x^*, x\rangle \implies x \in \partial f^*(x^*). \tag{40}$$

If $f(x) = f^{**}(x)$, e.g. for f proper convex lsc by the Fenchel–Moreau theorem, then the implication in (40) becomes equivalence. Note that a part of (40), the so-called *Fenchel–Young inequality*,

$$\forall_{x \in X} \forall_{x^* \in X^*} f(x) + f^*(x^*) \geqslant \langle x^*, x\rangle, \tag{41}$$

always holds [36, p 51]. The notion of the subgradient does not require the function f to be convex. Nonetheless, it uses global properties of f and is most useful in the context of convex functions.

The Fenchel–Rockafellar theorem provides sufficient conditions for the subdifferential of a convex function $f : X \to \mathbb{R} \cup \{+\infty\}$ to be non-empty at x, i.e. $\partial f(x) \neq \emptyset$ [37, p 121], *viz.*:

(i) f is lsc and $x \in \mathrm{core}(\mathrm{dom}\, f)$, or
(ii) $x \in \mathrm{cont}\, f$.

2.4. Optimization in Banach spaces

For the rest of this section, let X and Y be Banach spaces. The duality between product space $X \times Y$ and $X^* \times Y^*$ is given by $\langle (x^*, y^*), (x, y) \rangle \equiv \langle x^*, x \rangle + \langle y^*, y \rangle$.

Now, we provide a discussion of the basic concept of duality in optimization. The treatment we provide is relatively extensive, yet we find this topic to be of particular importance, and, additionally, often quite confusing. We discuss the two major duality schemes of Fenchel–Rockafellar and Lagrangian. For SDP the two schemes lead to the same results, but since most of the works refer to either of those two, it is useful to recognize and understand both. We consider the general approach, not limited to convex optimization unless explicitly stated. As we will show, for an optimization problem formulated as a minimization task, we can formulate a related maximization task with the property of the so-called *duality* meaning that the value of any feasible solution of the former is at least as large as the value of any feasible solution of the latter. The former optimization task is called a *primal* problem, and the latter a *dual* problem.

For both Fenchel–Rockafellar and Lagrangian schemes, we consider a single, possibly non-convex, function $F : X \times Y \to \mathbb{R} \cup \{+\infty\}$. The *primal problem* is defined as

$$\begin{aligned} &\text{minimize } F(x, 0) \\ &\text{subject to } x \in X, \end{aligned} \tag{42}$$

and the primal optimization is $\inf_{x \in X} \{F(x, 0)\}$; its value is called the *primal value* and denoted by p. A value of $x \in X$ for that the primal value p is attained, if exists, is called a *primal solution*. The *primal function*, or the *target function* is

$$f(x) \equiv F(x, 0)\,. \tag{43}$$

Actually, in most of the cases, one is interested in some, domain-specific, target function f, and in this sense, F is secondary to f. For a given f there exist many different possible functions F that satisfy (43). For this reason, F is called a *perturbation function* of f.

The *dual problem* is defined as

$$\begin{aligned} &\text{maximize } -F^*(0, y^*) \\ &\text{subject to } y^* \in Y^*, \end{aligned} \tag{44}$$

and the dual optimization is $\sup_{y^* \in Y^*} \{-F^*(0, y^*)\}$, where $F^* : X^* \times Y^* \to \mathbb{R} \cup \{-\infty, +\infty\}$ is the Fenchel conjugate with respect to both variables, i.e.

$$F^*(x^*, y^*) = \sup_{(x,y) \in X \times Y} \{\langle (x^*, y^*), (x, y) \rangle - F(x, y)\}\,. \tag{45}$$

Its value is called the *dual value* and denoted by d; a value of $y \in Y^*$ for that the value d is attained, if exists, is called the *dual solution*. The *dual function* is [41, p 216]:

$$g(y^*) \equiv -F^*(0, y^*). \tag{46}$$

The *value function* is defined as

$$\theta : Y \to \mathbb{R} \cup \{-\infty, +\infty\} : y \mapsto \inf_{x \in X} F(x, y). \tag{47}$$

The duality property is called *weak* when the value p of the optimal solution of the primal problem is greater or equal to the value d of the optimal solution of the dual problem, i.e. $p \geqslant d$. This property can be obtained directly from the definition of the Fenchel conjugate giving:

$$F(x, 0) + F^*(0, y^*) \geqslant \langle (0, y^*), (x, 0) \rangle = 0. \tag{48}$$

The duality is called *strong*, if the equality of the optimal primal and dual solution holds, i.e. $p = d$. The difference $p - d \geqslant 0$ is called the *duality gap*. The *duality gap* Δ is defined as

$$\Delta \equiv \begin{cases} 0 & \text{if } p = d \in \{-\infty, +\infty\} \\ p - d & \text{otherwise.} \end{cases} \tag{49}$$

For the strong duality to hold, we need equality in (48). Since $\langle (0, y^*), (x, 0) \rangle = 0$, from (40) it follows that the strong duality for $(\bar{x}, \bar{y}^*) \in X \times Y^*$ is equivalent to [198, pp 101–2]:

$$(0, \bar{y}^*) \in \partial F(\bar{x}, 0), \tag{50}$$

and if $F(\bar{x}, 0) = F^{**}(\bar{x}, 0)$, then also to [198, p 103]:

$$(\bar{x}, 0) \in \partial F^*(0, \bar{y}^*). \tag{51}$$

If the perturbation functions F is proper and is both convex and lsc in the second parameter, then it is called a *dualizing parametrization* [271] of the minimization $\inf_{x \in X}\{F(x, 0)\}$, resp. of the primal problem (42). Thus, F provides a family of optimizations $\inf_{x \in X}\{F(x, y)\}$, resp. problems, parameterized by the so-called *parameter* variable y [198, pp 100–1]. We stress that the same primal problem (42) is obtained with any other function $F' : X \times Y \to \mathbb{R} \cup \{+\infty\}$ satisfying $F(\cdot, 0) \equiv F'(\cdot, 0)$, but a different function $F' \neq F$ may lead to a different dual problem (44).

2.5. Fenchel–Rockafellar dualization scheme

First, for the Fenchel–Rockafellar duality [36, 37, 41, 138], consider a bounded linear map $A : X \to Y$, and two, possibly non-convex, functions, $f : X \to \mathbb{R} \cup \{+\infty\}$ and $g : Y \to \mathbb{R} \cup \{+\infty\}$. For the triple (A, f, g) define

$$F(x, y) \equiv f(x) + g(Ax + y). \tag{52}$$

The primal problem (42) is thus

$$\begin{aligned} &\text{minimize } f(x) + g(Ax) \\ &\text{subject to } x \in X. \end{aligned} \tag{53}$$

The *value function* (47) is equal

$$\theta : Y \to \mathbb{R} \cup \{-\infty, +\infty\} : y \mapsto \inf_{x \in X} (f(x) + g(Ax + y)) \tag{54}$$

with $\theta(0) = p$. We have [198, p 106]:

$$\begin{aligned} F^*(x^*, y^*) &= \sup_{(x,y) \in X \times Y} \{\langle (x^*, y^*), (x, y) \rangle - f(x) - g(Ax + y)\} \\ &= \sup_{(x,y) \in X \times Y} \{\langle x^* + A^* y^*, x \rangle + \langle -A^* y^*, x \rangle + \langle y^*, y \rangle - f(x) - g(Ax + y)\} \\ &= \sup_{(x,y) \in X \times Y} \{\langle x^* + A^* y^*, x \rangle + \langle -y^*, Ax + y \rangle - f(x) - g(Ax + y)\} \\ &= \sup_{x \in X} \left\{ \langle x^* + A^* y^*, x \rangle - f(x) + \sup_{y \in Y} (\langle -y^*, Ax + y \rangle - g(Ax + y)) \right\} \\ &= \sup_{x \in X} \{\langle x^* + A^* y^*, x \rangle - f(x) + g^*(-y^*)\} \\ &= f^*(x^* + A^* y^*) + g^*(-y^*). \end{aligned} \tag{55}$$

The dual problem (44) is thus equal

$$\begin{aligned} &\text{maximize } -f^*(A^* y^*) - g^*(-y^*) \\ &\text{subject to } y^* \in Y^*. \end{aligned} \tag{56}$$

The weak duality, *viz.* $p \geqslant d$, can be derived directly from the Fenchel–Young inequality (41). Indeed from the adjoint operator definition (1) we get

$$\begin{aligned} &\forall_{x \in X} \forall_{y^* \in Y^*} (f(x) + f^*(A^* y^*)) + (g(Ax) + g^*(-y^*)) \\ &\quad \geqslant \langle A^* y^*, x \rangle + \langle -y^*, Ax \rangle = \langle y^*, Ax \rangle - \langle y^*, Ax \rangle = 0. \end{aligned} \tag{57}$$

2.5.1. Strong duality. Knowing that the weak duality $p \geqslant d$ holds, to show the strong duality, we need to establish when $p \leqslant d$. Suppose that the subdifferential $\partial\theta$ is non-empty at 0. We will show that this suffices for the strong duality, and in section 2.5.2 provide the so-called constraint qualification conditions that ensure this. Let $y^* \in \partial\theta(0) \subseteq Y^*$. From the definition of subgradient for any $x \in X$ we have $\forall_{y \in Y} \theta(y - Ax) - \theta(0) \geqslant \langle y^*, y - Ax \rangle$, and thus, from the definition of the value function (54) we get

$$\begin{aligned} &\exists_{y^* \in Y^*} \forall_{x \in X} \forall_{y \in Y} \theta(0) \leqslant \theta(y - Ax) - \langle y^*, y - Ax \rangle \\ &\quad \leqslant f(x) + g(y) - \langle y^*, y - Ax \rangle = (f(x) + \langle A^* y^*, x \rangle) + (g(y) - \langle y^*, y \rangle). \end{aligned} \tag{58}$$

Since the inequality holds for all $x \in X$ and $y \in Y$, taking the infimum of these variables by the definition of the convex conjugate (33) we get $\exists_{y^* \in Y^*} \theta(0) \leqslant -f^*(-A^* y^*) - g^*(y^*)$ or, equivalently by negating the sign of y^* (as also $-y^* \in Y^*$), we get

$$\exists_{y^* \in Y^*} \theta(0) \leqslant -f^*(A^* y^*) - g^*(-y^*). \tag{59}$$

This, by (56), shows that $p = \theta(0) \leqslant d$, and thus

$$\partial\theta(0) \neq \emptyset \implies p \leqslant d, \tag{60}$$

and the strong duality holds.

2.5.2. The decoupling lemma. Now, we will discuss the so-called decoupling lemma [37, 38] providing a sufficient criterion, *viz.* certain *constraint qualification* condition, for the strong duality of the Fenchel–Rockafellar scheme. Since this topic is crucial for duality in convex optimization, for the sake of completeness, we provide in this work in appendix A a summary of the proof of the theorem, following [37, p 127nn].

Let us assume that $f: X \to \mathbb{R}$ and $g: Y \to \mathbb{R}$ are both convex and that the map $A: X \to Y$ is linear and bounded. We provide sufficient conditions for θ as defined in (54) to have non-empty subdifferential at 0, i.e. $\partial\theta(0) \neq \emptyset$. We use the Fenchel–Rockafellar theorem, see section 2.3.2. One can directly check that θ when f and g are convex, then θ given by (54) is a convex function with $\mathrm{dom}\theta = \mathrm{dom}g - A\mathrm{dom}f$, where the set difference is the Minkowski difference. Indeed, $F(x,y)$ defined as (52) is convex as a sum of convex f and g is of the form (26), and satisfies the condition (27). The decoupling lemma states that under the condition that both f and g are lsc and

$$0 \in \mathrm{core}\,(\mathrm{dom}\,\theta)\,, \tag{61}$$

the function θ defined as in (54) is continuous at 0. Then, from the Fenchel–Rockafellar theorem stated in section 2.3.2, it follows that $\partial\theta$ is non-empty at 0, as required for the proof of strong duality as given in section 2.5.1.

2.6. Lagrangian dualization scheme

Next, for the Lagrangian duality [138, 198], consider a single, again possibly non-convex, function F. Nonetheless, again, if F is convex in both parameters, then θ is of the form (26) and satisfies (27), thus θ is convex in this case. The Lagrangian of F is defined as [138, 198, 271]

$$\mathcal{L}: X \times Y^* \to \mathbb{R} \cup \{-\infty, +\infty\} : (x, y^*) \mapsto -\sup_{y \in Y} \{\langle y^*, y\rangle - F(x,y)\}\,, \tag{62}$$

where one can easily recognize the Fenchel conjugate with respect to the parameter (i.e. the second) variable.

The Lagrangian allows reformulating the primal problem (42). From the definition (62) it follows that

$$\forall_{x \in X} \forall_{y^* \in Y^*} \forall_{y \in Y} F(x,y) \geqslant \mathcal{L}(x, y^*) + \langle y^*, y\rangle. \tag{63}$$

When F is a dualizing parametrization, then by the Fenchel–Moreau theorem, see (38), the equality in (63) holds. From (63) we have for the primal optimization, see (42):

$$\inf_{x \in X} \{F(x,0)\} \geqslant \inf_{x \in X} \sup_{y^* \in Y^*} \mathcal{L}(x, y^*)\,. \tag{64}$$

Also the dual optimization, see (44), can be easily expressed with the Lagrangian, *viz.* [198, p 109]:

$$\begin{aligned} \sup_{y^* \in Y^*} \{-F^*(0, y^*)\} &= \sup_{y^* \in Y^*} \left\{ - \sup_{x \in X, y \in Y} \{\langle (0, y^*), (x,y)\rangle - F(x,y)\} \right\} \\ &= \sup_{y^* \in Y^*} \left\{ -\sup_{x \in X} \sup_{y \in Y} \{\langle y^*, y\rangle - F(x,y)\} \right\} = \sup_{y^* \in Y^*} \inf_{x \in X} \mathcal{L}(x, y^*)\,. \end{aligned} \tag{65}$$

The inner infimum of the last expression is sometimes used as an alternative definition [41, p 216] of the dual function (46):

$$g(y^*) \equiv \inf_{x\in X} \mathcal{L}(x,y^*). \tag{66}$$

A direct consequence of (64) and (65) is another proof of the weak duality:

$$\inf_{x\in X}\{F(x,0)\} \geqslant \inf_{x\in X}\sup_{y^*\in Y^*} \mathcal{L}(x,y^*) \geqslant \sup_{y^*\in Y^*}\inf_{x\in X} \mathcal{L}(x,y^*) = \sup_{y^*\in Y^*}\{-F^*(0,y^*)\}. \tag{67}$$

The second inequality follows from the well-known max–min inequality [313]. A value $(\bar{x},\bar{y}^*)\in X\times Y^*$ is defined to be a *saddle point* of $\mathcal{L}$ when

$$\forall_{x\in X}\forall_{y^*\in Y^*}\, \mathcal{L}(\bar{x},y^*) \leqslant \mathcal{L}(\bar{x},\bar{y}^*) \leqslant \mathcal{L}(x,\bar{y}^*), \tag{68}$$

i.e. in other words, when

$$\sup_{y^*\in Y^*} \mathcal{L}(\bar{x},y^*) = \mathcal{L}(\bar{x},\bar{y}^*) = \inf_{x\in X} \mathcal{L}(x,\bar{y}^*). \tag{69}$$

From (67) we also directly get that for F being a dualizing parametrization the strong duality for $(\bar{x},\bar{y}^*)$ is equivalent to saying that $(\bar{x},\bar{y}^*)$ is a saddle point of $\mathcal{L}$, see e.g. [198, p 110] for a proof.

2.7. Convex cone optimization and duality

We have introduced the general framework for optimization and discussed its duality. Now, we concentrate on a particular problem of convex cone programming, i.e. the optimization over variables belonging to a convex cone [41, 81, 214, 235]. We write convex cone optimization problems as:

$$\begin{aligned} &\text{minimize } c^\dagger x \\ &\text{subject to } \mathcal{A}x = b, \\ &\qquad\qquad x\in K\subseteq\mathcal{X}, \end{aligned} \tag{70}$$

in the primal form, see (42), and in the dual form, see (44), as:

$$\begin{aligned} &\text{maximize } \langle y^*, b\rangle \\ &\text{subject to } c^\dagger - \mathcal{A}^\dagger y^* = z^*, \\ &\qquad\qquad z^*\in K^*\subseteq\mathcal{X}^*, \end{aligned} \tag{71}$$

where K is a nonempty, closed convex cone in an Euclidean space $\mathcal{X}$, see (19), $\mathcal{A}:\mathcal{X}\to\mathbb{R}^m$ is a linear operator, the operator $\mathcal{A}^\dagger:(\mathbb{R}^m)^*\to\mathcal{X}^*$ is its adjoint, $b\in\mathbb{R}^m$, $y^*\in(\mathbb{R}^m)^*$ and $c\in\mathcal{X}$. Note, that the spaces $\mathbb{R}^m$ and $(\mathbb{R}^m)^*$ are isomorphic with the transposition operation as the isomorphism.

To derive (71) from (70) we need a parametrization of the family of problems [198, pp 111–2]. *One* of the possibilities is to introduce a variable y used as the parameter for the linear constraints and take

$$F(x,y) = \begin{cases} c^\dagger x + I_{\{x:\mathcal{A}x-b=y\}}[x] & \text{if } x\in K, \\ +\infty & \text{otherwise} \end{cases} \tag{72}$$

where we used the indicator function (32). We stress that this is only one of the multiple examples of a dualizing parametrization (note that the indicator is over a convex closed set, and thus is convex and lsc): the most direct and simple, but yet arbitrary. To get the dual problem (44) we calculate

$$\begin{aligned} -F^*(0,y^*) &= -\sup_{x\in X, y\in Y}\{\langle y^*,y\rangle - F(x,y)\} = \inf_{x\in K, y\in Y}\left\{c^\dagger x + I_{\{x:\mathcal{A}x-b=y\}}[x] - \langle y^*,y\rangle\right\} \\ &= \inf_{x\in K}\left\{c^\dagger x - \langle y^*, \mathcal{A}x-b\rangle\right\} = \inf_{x\in K}\left\{\langle c^\dagger - \mathcal{A}^\dagger y^*, x\rangle + \langle y^*, b\rangle\right\}. \end{aligned} \tag{73}$$

The term $\langle c^\dagger - \mathcal{A}^\dagger y^*, x\rangle$ is non-negative for all $x \in K$ if and only if $c^\dagger - \mathcal{A}^\dagger y^*$ belongs to the dual cone K^*; and if a negative value can be attained for some $x \in K$, then the infimum is $-\infty$. Thus

$$-F^*(0,y^*) = \begin{cases} \langle y^*, b\rangle & \text{if } c^\dagger - \mathcal{A}^\dagger y^* \in K^*, \\ -\infty & \text{otherwise.} \end{cases} \tag{74}$$

The problem (71) is derived as $\sup_{y^*\in Y^*}\{-F^*(0,y^*)\}$, see (44), by introducing $z^* = c^\dagger - \mathcal{A}^\dagger y^*$.

The same result can be equivalently achieved, but more step by step, with the approach using the Lagrangian (62), which if $x \in K$ for F given by (72) is [41, p 266]:

$$\begin{aligned} \mathcal{L}(x,y^*) &= -\sup_{y\in Y}\{\langle y^*,y\rangle - F(x,y)\} = \inf_{y\in Y}\{F(x,y) - \langle y^*,y\rangle\} \\ &= \inf_{y:\mathcal{A}x-b=y, y\in Y}\left\{c^\dagger x - \langle y^*,y\rangle\right\} = \langle c^\dagger - \mathcal{A}^\dagger y^*, x\rangle + \langle y^*, b\rangle \end{aligned} \tag{75}$$

and $\mathcal{L}(x,y^*) = +\infty$ if $x \notin K$. The dual is derived as, *see* (65) and (73):

$$\sup_{y^*\in Y^*}\inf_{x\in X}\mathcal{L}(x,y^*) = \sup_{y^*\in Y^*}\inf_{x\in K}\mathcal{L}(x,y^*). \tag{76}$$

In particular, we see that the dual function (66) is the same as in (74):

$$g(y^*) = \inf_{x\in X}\mathcal{L}(x,y^*) = \inf_{x\in K}\mathcal{L}(x,y^*) = -F^*(0,y^*) = \begin{cases} \langle y^*, b\rangle & \text{if } c^\dagger - \mathcal{A}^\dagger y^* \in K^*, \\ -\infty & \text{otherwise.} \end{cases} \tag{77}$$

3. Theory of SDP

In this section, we delve into the foundational concepts and principles underlying the field of SDP. The section 3.1 elucidates the fundamental properties and criteria for positive semi-definiteness of matrices, which form the basis for semidefinite optimization problems. In section 3.2 we investigate various primal and dual formulations present in the literature. Next, section 3.3 explores the duality theory associated with SDP, highlighting the relationships between primal and dual problems, and section 3.4 discusses how a solution of a dual problem can provide a useful linear (affine) bound on a range of parameterized primal problems. Finally, sections 3.5 and 3.6 cover specialized topics, shedding light on the utilization of complex variables, the incorporation of slack and surplus variables, and the treatment of mixed problems and equalities in the context of SDP. In section 3.7 we discuss simple tricks related

to the Schur complement. Then, in section 3.8 we briefly discuss implementations of SDP solvers, and in section 3.9 we outline selected internal solver mechanisms that may impact the performance.

The following overview can be supplemented with numerous other applications and constructions. These include the famous MAX-CUT and MAX-k-SAT relaxations by Goemans and Williamson [123], finding maximum eigenvalues, matrix norms optimizations, and combinatorial optimization problems [8, 9, 30, 121, 130, 140, 215, 240, 323, 325].

3.1. Definition and characterization of positive semidefiniteness

Discussion of SDP requires us to introduce the concept of *PD* or simply *positive* matrices, as well as *PSD* or simply *semi-definite* matrices. We denote a positive (or semi-positive) matrix M by $M \succ 0$ (or $M \succeq 0$). PSD matrices are also referred to as *non-negative definite* or simply *non-negative*. Several equivalent definitions or characterizations of such matrices can be found in the literature, and here we present three of them. Thus, a symmetric matrix $M \in \mathbb{R}^{n\times n}$ is considered positive (or non-negative) if all its eigenvalues are positive (or non-negative). Alternatively, M is positive (or non-negative) if and only if for all $x \in \mathbb{R}^n$ with $x \neq 0$, it holds that $x^{\mathrm{T}}Mx > 0$ (or $x^{\mathrm{T}}Mx \geqslant 0$). Similarly, for a Hermitian matrix $M \in \mathbb{C}^{n\times n}$, it is PD (or PSD) if and only if for all $x \in \mathbb{C}^n$ with $x \neq 0$, we have $x^{\dagger}Mx > 0$ (or $x^{\dagger}Mx \geqslant 0$). The former definition based on eigenvalues seems to offer a greater intuitive understanding, while the latter is more prevalent in the existing literature on the subject. A more comprehensive exploration of the properties of PD and PSD matrices can be found in [150, 209]. It should be noted that a real PD (PSD) matrix satisfies the conditions for Hermitian matrices, making it a complex PD (PSD) matrix as well. Conversely, for a complex PD (PSD) matrix $M = M^{\mathrm{R}} + iM^{\mathrm{I}}$ (where M^{R} and M^{I} are real symmetric and antisymmetric matrices), we observe that $\forall_{x\in\mathbb{C}^n\ x\neq 0} x^{\dagger}\frac{1}{2}(M+M^{\dagger})x = x^{\dagger}M^{\mathrm{R}}x \geqslant 0$, implying that the matrix M^{R} is a real PD (PSD) matrix. It is evident that PSD matrices of size n form a convex cone $\mathbb{S}^n_+$, as indicated in equation (19), and this cone is self-dual.

Now we state two very important properties characterizing PSD matrices by their possible decompositions [209, 316]. It can be shown that for a Hermitian (symmetric) matrix $M \in \mathbb{C}^{n\times n}$ ($M \in \mathbb{R}^{n\times n}$) we have that $M \succeq 0$ is equivalent to each of the following statements:

(i) There exists $L \in \mathbb{C}^{n\times n}$ ($L \in \mathbb{R}^{n\times n}$) such, that $M = L^{\dagger}L$ ($M = L^{\mathrm{T}}L$), and L is a lower triangular matrix.
(ii) There exists a set of vectors $\{v_i\}_{i\in[n]}$, $v_i \in \mathbb{C}^n$ ($v_i \in \mathbb{R}^n$), such that $M_{i,j} = v_i^{\dagger}\cdot v_j$ ($M_{i,j} = v_i^{\mathrm{T}}\cdot v_j$).

In the first of these characterizations, a non-unique matrix L is called the *Cholesky decomposition* of M. The second characterization is equivalent to the existence of a matrix $B \in \mathbb{C}^{n\times n}$ ($B \in \mathbb{R}^{n\times n}$), such that $M = B^{\dagger}B$ ($M = B^{\mathrm{T}}B$), so, in other words, M is a multiplication table of vectors $\{v_i\}_i$ being columns of B. Some authors [324, 326] existence of such matrix B use as the definition of positive semi-definiteness. We say that M is a Gram matrix, or a Gramian. The relation between the existence of a set of vectors and PSD property can be generalized to infinite-dimensional spaces [206]. Trivially, the set of vectors is linearly independent if and only if the determinant of its corresponding Gram matrix is non-zero.

The notion of PD and PSD is also characterized by the Sylvester criteria, formulated as follows. Let M be an $m \times n$ matrix, and consider sets $I \subseteq [m]$ and $J \subseteq [n]$ of equal sizes. Let $(M)_{I,J}$ be a submatrix with elements contained in rows from I and columns from J. The determinant of $(M)_{I,J}$ or, in other words, the determinant of a square submatrix obtained by removing

some rows and columns of a larger matrix, is called a minor of the matrix. If $I = J$, then the minor is called *principal*. If $I = J = [k]$, for $k \leqslant n, m$, then the principal minor is called *leading*. Sylvester's criteria provide necessary and sufficient conditions for positive definiteness and semi-definiteness of a Hermitian matrix [117]. Sylvester's criterion for positive definiteness states that a Hermitian matrix is PD if and only if, all its leading principal minors are positive. Sylvester's criterion for positive semi-definiteness states that a Hermitian matrix is PSD if and only if, all principal minors are non-negative. For example an SDP constraint $\begin{bmatrix} 1 & x \\ x & 1 \end{bmatrix} \succeq 0$ implies by the second Sylvester's criterion that $|x| \leqslant 1$.

For the Löwner's partial order $\succeq$ it can be easily shown that if $A, B \succeq 0$, then $A + B \succeq 0$. If we multiply a PSD matrix by a non-negative constant, we get another PSD matrix. Thus the set of PSD matrices forms a pointed convex cone. It also follows for $A, B \succeq 0$ that $\mathrm{Tr}(AB) \geqslant 0$ and $A^{\frac{1}{2}}$ exists and is PSD.

3.2. Formulations of semidefinite optimization problems

In the literature there exist a couple of equivalent formulations of SDPs, each has both primal and dual forms. The author prefers the so-called standard or canonical form of SDP given below in (80) and (81) in section 3.2.1, and used in many of the classical textbooks [15, 36, 116], reviews [231, 306], SDP fundamental papers [10, 216, 293, 305, 307] and implementations [214, 292, 309, 310, 312], sometimes with slight changes in labeling [27], different notations for the Frobenius product (3), and more general form of conic formulations [66, 236]. Another important formulation is the one used by Vandenberghe and Boyd [41, 315], which we provide in section 3.2.2. This form seems to be preferred in many quantum information papers [42, 95, 160, 227] with direct influence of [315], which is apparently the default reference to the SDPs. The third important formulation was given by Watrous in his lecture notes [324] and textbook [326], see section 3.2.3 below. It has an elegant symmetric form and also is used in many quantum information books and papers [61, 311], especially involving quantum channels [54, 183, 199, 207, 251].

The paradigmatic part of all the formulations are LMIs, i.e. expressions of the form [40]:

$$F(x) \succeq 0, \text{where} \tag{78a}$$

$$F(x) \equiv F_0 + \sum_{i \in [m]} x_i F_i, \tag{78b}$$

$x \in \mathbb{R}^m$ is a variable, and F_i, for $i = 0, \ldots, m$, are symmetric constant matrices $\mathbb{R}^{n \times n}$. The origin of LMIs is in control theory including solving Lyapunov stability problems, and their interconnection with convex optimization has been noted e.g. by Pyatnitskii and Skorodinskii [40, 262]. In fact, SDPs can be intuitively viewed as optimization problems with linear target functions and LMIs as constraints. Any SDP can be formulated as either a primal or dual problem of the formulations given below. From the form of constraints (78*a*) it directly follows that they are convex:

$$F(\lambda x + (1-\lambda)x') = \lambda F(x) + (1-\lambda) F(x') \succ 0 \tag{79}$$

for $\lambda \in [0, 1]$. The linear target function is obviously also convex. Thus the SDP problems are convex, so we can use the methods of section 2. We also note that any number of LMIs can be reformulated as a single LMI involving block-diagonal matrices, with each block referring to a relevant LMI. We refer to [68] for an overview of applications of LMIs.

3.2.1. The canonical or standard form. Again, let $m, n \in \mathbb{N}$, $m \leqslant \frac{n(n+1)}{2}$. An SDP problem in a *canonical*, or *standard*, *primal* form is the following optimization task in a variable $X \in \mathbb{S}^n$:

$$\begin{aligned} &\text{minimize } C \bullet X \\ &\text{subject to } A_i \bullet X = b_i, \text{for } i \in [m] \\ &\qquad X \succeq 0, \end{aligned} \tag{80}$$

where $C \in \mathbb{S}^n$ and $A_1, \ldots, A_m \in \mathbb{S}^n$ are symmetric matrices. The matrices A_i, C, and vector $b \in \mathbb{R}^m$ define the SDP problem. Note that the fact that these matrices are symmetric is not restrictive. For a symmetric matrix X and a matrix C we have $C \bullet X = \mathrm{Tr}\left(\frac{1}{2}(C + C^{\mathrm{T}})X\right)$, and thus we may always take a symmetric matrix $\frac{1}{2}(C + C^{\mathrm{T}})$ instead of C. We assume that $A_1, \ldots, A_m$ are linearly independent (otherwise we can reduce this set). Obviously, LP problem (13) may be written in the form of SDP (80), if X is constrained to be a diagonal matrix, with the diagonal entries used as the x variable. Thus, LP can be considered as a particular case of SDP. The goal expression $C \bullet X = \mathrm{Tr}(CX)$, but the former notation is more often used; similarly $A_i \bullet X = \mathrm{Tr}(A_i X)$.

A canonical *dual* SDP problem for (80) is the optimization task in variables $y \in \mathbb{R}^m$ and $Z \in \mathbb{S}^n$ of the following form

$$\begin{aligned} &\text{maximize } b^{\mathrm{T}} \cdot y \\ &\text{subject to } C - \sum_{i \in [m]} y_i A_i = Z \\ &\qquad Z \succeq 0. \end{aligned} \tag{81}$$

Some authors [45, pp 39–40] rewrite the canonical form (80) and (81) with substitutions F_i instead of A_i, $-C$ instead of C, and $-\lambda_i$ instead of y_i, turning the primal problem to maximization, and the dual problem to minimization.

Similarly as in LP, X is called the *primal variable*, y the *dual variable*, Z the *dual slack variable*, $\{A_i\}$ are *linear constraint matrices*, b is the *RHS of the linear constraint*, and C is the *linear coefficient*. If $X, Z \in \mathbb{R}^{n \times n}$ and $y \in \mathbb{R}^m$ satisfies conditions specified by (80) and (81), then they are called a *feasible solution*. A feasible variable X is called a *primal solution*, and feasible variables Z and y constitute the *dual solution*. An optimal solution is required to be feasible. The values of $C \bullet X$ and $b^{\mathrm{T}} \cdot y$ are called the values of the primal and dual solutions, or *values of the problem*, respectively. We have $C \bullet X \geqslant b^{\mathrm{T}} \cdot y$. Usually, an SDP solver is expected to find both primal and dual solutions. If either $C = 0 \in \mathbb{R}^{n \times n}$ or $b = 0$, then such a problem is called *feasibility problem* and refers to finding whether *any* solution of given, the primal or dual, problem exists. As the dual form is often delivered from the Lagrange duality 2.6, and y, in that case, plays the role of Lagrange multipliers, this name is also usually attributed to the dual variable y [41].

The fact that primal formulation refers to minimization and dual to maximization problems, is not restrictive. We can always change the sign of the matrix C or the vector b to get the desired optimization problem fitting into the standard form in (80) and (81). What is more, a problem formulated in one of the forms given by (80) and (81) may be reformulated in the other one. The issue of choosing the proper formulation is not always obvious and can have a very significant impact on the difficulty of the problem to a solver [194]. This can be illustrated by the example in table 1 showing the sizes of some SDP problems in dual and primal formulations. Generally, one should choose the formulation which leads to a smaller number of constraints, given by the number m unless some special properties of the structure of the formulation can be used to further simplify the process of solving the problem, see e.g. section 3.9.

3.2.2. The Vandenberghe and Boyd and the SDPA forms. In the formulation popularized by [41, 315] the primal optimization task is:

$$\begin{aligned} &\text{minimize } c^{\mathrm{T}} \cdot x \\ &\text{subject to } F(x) \succeq 0, \end{aligned} \tag{82}$$

where $F(x)$ is given by (78*b*). As stated in [315] the aim of this formulation is to make the primal formulation 'as *explicit* as possible'. The dual of (82) is

$$\begin{aligned} &\text{maximize } -\mathrm{Tr}\left[F_0 Z\right] \\ &\text{subject to } \mathrm{Tr}\left[F_i Z\right] = c_i, \text{for } i \in [m], \\ &\qquad Z \succeq 0, \end{aligned} \tag{83}$$

where the variable is a symmetric matrix $Z \in \mathbb{R}^{n\times n}$.

The form where F_0 takes an opposite sign is often referred to as the SDPA, see section 1.3 for a discussion. Stated explicitly, keeping the original notation and naming of the variables (note using the label Y instead of Z in the dual), the SDPA primal form is [113]:

$$\begin{aligned} &\text{minimize } \sum_{i\in[m]} c_i x_i, \\ &\text{subject to } X \equiv -F_0 + \sum_{i\in[m]} F_i x_i \succeq 0, \end{aligned} \tag{84}$$

and the SDPA dual form is:

$$\begin{aligned} &\text{maximize } \mathrm{Tr}\left[F_0 Y\right] \\ &\text{subject to } \mathrm{Tr}\left[F_i Y\right] = c_i, \text{for } i \in [m], \\ &\qquad Y \succeq 0. \end{aligned} \tag{85}$$

3.2.3. The Watrous symmetric form. The third common form of SDPs is given by Watrous [324, 326]. This form is designed to show the symmetry between the primal and dual problems and is particularly convenient for quantum channel analysis. For two complex Euclidean spaces $\mathcal{X}$ and $\mathcal{Y}$, a semidefinite program in the Watrous form is defined as a triple (Φ, A, B), where $\Phi : \mathsf{L}[\mathcal{X},\mathcal{X}] \to \mathsf{L}[\mathcal{Y},\mathcal{Y}]$ is a Hermitian and trace-preserving map, $A \in \mathrm{Herm}\mathcal{X}$, and $B \in \mathrm{Herm}(\mathcal{Y})$. The primal problem in the Watrous form is:

$$\begin{aligned} &\text{maximize } \langle A, X\rangle \\ &\text{subject to } \Phi(X) = B, \\ &\qquad X \succeq 0, \end{aligned} \tag{86}$$

and the dual is:

$$\begin{aligned} &\text{minimize } \langle B, Y\rangle \\ &\text{subject to } \Phi^{\dagger}(Y) \succeq A, \\ &\qquad Y \in \mathrm{Herm}(\mathcal{Y}), \end{aligned} \tag{87}$$

with a remark that in the original notation of [324, 326] Watrous uses * instead of † to denote the Hermitian conjugate, $\geqslant$ instead of $\succeq$ to denote the Löwner's partial order, and $X \in \mathrm{Pos}(\mathcal{X})$ instead of $X \succeq 0$.

3.2.4. The Kronecker-canonical form for convex cones. Here we briefly show the methodology that offers an alternative way of expressing the canonical formulation of (80) and (81), resembling the LP formulations in (13) and (14). This formalism, in fact, is more encompassing and applicable to a wide range of conic optimization problems, as discussed in section 2.7. The general formulation (70) and (71) is highly convenient and valuable, as it facilitates a seamless transition between primal and dual formulations for any convex cone by establishing the corresponding dual cone.

It can be easily verified that for any real matrices A, B, and C, the relationship $(A \otimes B)\mathrm{vec}(C) = \mathrm{vec}(BCA^{\mathrm{T}})$ holds. We define a matrix $\mathcal{A} \equiv [a_1; \cdots; a_m] \in \mathbb{R}^{n^2 \times m}$, where $a_i = \mathrm{vec}(A_i)$, referring to the matrices in (80). Hence, a_i represents the ith column of $\mathcal{A}$. Consequently, we have $\mathrm{vec}(\sum_{i\in[m]} y_i A_i) = \mathcal{A}y$ and $A_i \bullet X = (\mathcal{A}^{\mathrm{T}}x)_i$, where $(\mathcal{A}^{\mathrm{T}}x)_i$ denotes the ith element of the vector $\mathcal{A}^{\mathrm{T}}x$, $c = \mathrm{vec}(C)$, and $x = \mathrm{vec}(X)$. When K represents the self-dual cone of real or convex PSD n by n matrices, substituting these expressions into (70) and (71) yields an alternative and equivalent formulation for the canonical SDP, commonly used, for instance, in [214, 236, 291, 309], as illustrated in (111*a*) and (111*b*).

3.3. Duality of SDP

Recall that an important property of primal and dual formulations is that any feasible solution to a primal problem provides an upper bound on all feasible solutions to the dual problem. The weak duality property of SDP, *viz.* $C \bullet X \geqslant b^{\mathrm{T}} \cdot y$ is derived as follows:

$$\begin{aligned} C \bullet X - b^{\mathrm{T}} \cdot y &= \left(Z + \sum_{i\in[m]} y_i A_i \right) \bullet X - b^{\mathrm{T}} \cdot y \\ &= \mathrm{Tr}\,(ZX) + \sum_{i\in[m]} y_i \cdot \mathrm{Tr}\,(A_i X) - b^{\mathrm{T}} \cdot y = \mathrm{Tr}\,(XZ) \geqslant 0. \end{aligned} \tag{88}$$

In LP, the value of the primal and dual problems are always equal meaning the strong duality. Now, we provide a sufficient condition for strong duality to occur in SDP, as it is observed in many cases. Let p^* be the optimal value of the primal SDP problem, and d^* be the optimal value of the dual SDP problem. One can show that it holds $p^* = d^*$ if at least one of the conditions is satisfied [9, 140]:

(i) There exist $y \in \mathbb{R}^m$, such that $C - \sum_{i\in[m]} y_i A_i \succ 0$, i.e. the dual problem is strictly feasible (then also the value d^* is attained).
(ii) There exists $X \succ 0$, such that $A_i \bullet X = b_i$, i.e. the primal problem is strictly feasible (then also the value p^* is attained).

These statements are called the Slater conditions [285].

One of the most confusing and intriguing questions in the theory of SDP is asking whether the dual of the dual form is the primal form. The affirmative answer can be derived in the following way:

$$\sup_y \left\{ b^{\mathrm{T}} \cdot y : C - \sum_i (A_i y_i) \succeq 0 \right\} = \sup_y \inf_{X \succeq 0} \left\{ b^{\mathrm{T}} \cdot y + \left(C - \sum_i (A_i y_i) \right) \bullet X \right\}$$
$$\leqslant \inf_{X \succeq 0} \sup_y \left\{ C \bullet X + \sum_i y_i \cdot (b_i - A_i \bullet X) \right\} = \inf_{\substack{X \succeq 0, \\ A_i \bullet X = b_i}} \{ C \bullet X \} . \tag{89}$$

Indeed:

$$\inf_{X \succeq 0} \left\{ b^{\mathrm{T}} \cdot y + X \bullet Z \right\} = \begin{cases} b^{\mathrm{T}} \cdot y & \text{if } Z \succeq 0 \\ -\infty & \text{otherwise} \end{cases}, \text{ and} \tag{90a}$$

$$\sup_y \left\{ C \bullet X + \sum_i y_i \cdot (b_i - A_i \bullet X) \right\} = \begin{cases} C \bullet X & \text{if } \forall_i A_i \bullet X = b_i \\ +\infty & \text{otherwise} \end{cases}. \tag{90b}$$

3.4. Affine bounds from dual problems

We will now show how by solving a dual problem one can get a linear bound on the solution of a parameterized family of primal problems. For the sake of illustration, we will use the SDPA form (84) and (85) with additional linear variables (see section 3.6 for further discussion), but similar reasoning can be applied to pairs of primal and dual problems of any other form.

Suppose that for given $k, m, n \in \mathbb{N}_+$, and $c \in \mathbb{R}^m$, $\{F_i\}_{i=0}^m \subset \mathbb{S}^n$ and $\{q_j\}_{j \in [k]} \subset \mathbb{R}^m$ we want to find a lower bound on a function $S : \mathbb{R}^k \to \mathbb{R}$ defined as:

$$S(v) \equiv \inf_{x \in \mathbb{R}^m} \left\{ \sum_{i \in [m]} c_i x_i : \forall_{j \in [k]} q_j \cdot x - v_j = 0; -F_0 + \sum_{i \in [m]} F_i x_i \succeq 0 \right\}. \tag{91}$$

For fixed value of $v \in \mathbb{R}^k$ the value of $S(v)$ is given by the solution of the following SDP, see (84):

$$\begin{aligned} &\text{minimize} \sum_{i \in [m]} c_i x_i, \\ &\text{subject to } X \equiv -F_0 + \sum_{i \in [m]} F_i x_i \succeq 0, \\ &\qquad q_j \cdot x - v_j = 0 \text{ for } j \in [k] . \end{aligned} \tag{92}$$

Following the Lagrangian dualization scheme from section 2.6, we get

$$\begin{aligned} S(v) &= \inf_{x\in\mathbb{R}^m} \sup_{\substack{\beta\in\mathbb{R}^k \\ Y\in\mathbb{S}^n_+}} \left\{ \sum_{i\in[m]} c_i x_i - \left(\sum_{j\in[k]} \beta_k \cdot (q_j \cdot x - v_j) \right) - \left(-F_0 + \sum_{i\in[m]} F_i x_i \right) \bullet Y \right\} \\ &\geqslant \sup_{\substack{\beta\in\mathbb{R}^k \\ Y\in\mathbb{S}^n_+}} \inf_{x\in\mathbb{R}^m} \left\{ F_0 \bullet Y + \sum_{j\in[k]} \beta_j v_j + \sum_{i\in[m]} x_i \cdot \left(c_i - \sum_{j\in[k]} \beta_j q_{ji} - F_i \bullet Y \right) \right\} \\ &= \sup_{\substack{\beta\in\mathbb{R}^k \\ Y\in\mathbb{S}^n_+}} \left\{ F_0 \bullet Y + \sum_{j\in[k]} \beta_j v_j : \forall_{i\in[m]} c_i - \sum_{j\in[k]} \beta_j q_{ji} - F_i \bullet Y = 0 \right\} = \sup_{(\beta,Y)\in\mathcal{F}} \tilde{S}_{\beta,Y}(v), \end{aligned} \tag{93}$$

where $\mathcal{F} \equiv \left\{ (\beta, Y) \in \mathbb{R}^k \otimes \mathbb{S}^n_+ : \forall_{i\in[m]} \sum_{j\in[k]} \beta_j q_{ji} + F_i \bullet Y = c_i \right\}$ is the feasible set of the problem, and $\tilde{S}_{\beta,Y}(v) \equiv F_0 \bullet Y + \sum_{j\in[k]} \beta_j v_j$. Thus, the dual of (92) is:

$$\begin{aligned} &\text{maximize } \mathrm{Tr}\left[F_0 Y\right] + \sum_{j\in[k]} \beta_j v_j \\ &\text{subject to } \mathrm{Tr}\left[F_i Y\right] + \sum_{j\in[k]} \beta_j q_{ji} = c_i, \text{for } i \in [m], \\ &\qquad Y \succeq 0. \end{aligned} \tag{94}$$

In consequence, by taking any feasible pair (β, Y) one obtains an affine lower bound $S(v) \geqslant \tilde{S}_{\beta,Y}(v)$. From the strong duality, it follows that the bound is tight for (β, Y) being the optimal solution of (94).

3.5. *Complex variables in semidefinite problems*

In section 3.2.1 we considered SDPs where the problems were defined by real matrices and vectors, and the optimization was carried over real-valued variables. One of the first works showing how to reduce a problem where some of the elements of the problem involve complex numbers was [122].

Let $B \in \mathbb{C}^{n\times n}$ be a Hermitian matrix, B^{R} and B^{I} its real and imaginary parts, respectively, i.e. $B = B^{\mathrm{R}} + iB^{\mathrm{I}}$ and $B^{\mathrm{I}} = -B^{\mathrm{I}}$. Then $B \succeq 0$ if and only if

$$\begin{bmatrix} B^{\mathrm{R}} & -B^{\mathrm{I}} \\ B^{\mathrm{I}} & B^{\mathrm{R}} \end{bmatrix} \succeq 0. \tag{95}$$

Indeed, for any complex vector $w = u + iv \in \mathbb{C}^n$ we have

$$\begin{aligned} w^\dagger B w = &\left(u^{\mathrm{T}} - iv^{\mathrm{T}}\right)\left(B^{\mathrm{R}} + iB^{\mathrm{I}}\right)(u + iv) = \left(u^{\mathrm{T}}B^{\mathrm{R}}u + v^{\mathrm{T}}B^{\mathrm{R}}v - u^{\mathrm{T}}B^{\mathrm{I}}v + v^{\mathrm{T}}B^{\mathrm{I}}u\right) \\ &+ i\left(u^{\mathrm{T}}B^{\mathrm{R}}v - v^{\mathrm{T}}B^{\mathrm{R}}u + u^{\mathrm{T}}B^{\mathrm{I}}u + v^{\mathrm{T}}B^{\mathrm{I}}v\right) \geqslant 0 \end{aligned} \tag{96}$$

if and only if $\begin{bmatrix} u^{\mathrm{T}} & v^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} B^{\mathrm{R}} & -B^{\mathrm{I}} \\ B^{\mathrm{I}} & B^{\mathrm{R}} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \succeq 0$. This is because $u^{\mathrm{T}}B^{\mathrm{I}}u = v^{\mathrm{T}}B^{\mathrm{I}}v = 0$ and $u^{\mathrm{T}}B^{\mathrm{R}}v = v^{\mathrm{T}}B^{\mathrm{R}}u$. Thus any SDP problem defined in terms of complex vectors and Hermitian matrices

can be stated as a problem involving only real vectors with symmetric matrices. For instance, let us consider the case when both the linear coefficient C and the primal variable X are complex matrices, $C = C^{\mathrm{R}} + iC^{\mathrm{I}}$, with $C^{\mathrm{R}}, C^{\mathrm{I}} \in \mathbb{R}^{n\times n}$ and $X \in \mathbb{C}^{n\times n}$. When we reframe this as a real-valued SDP, then the target function takes the form of $\begin{bmatrix} C^{\mathrm{R}} & -C^{\mathrm{I}} \\ C^{\mathrm{I}} & C^{\mathrm{R}} \end{bmatrix}$, see (95), and the primal real-valued variable $X^{(\mathrm{R})} = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix}$ does not require explicit constraints. Instead, the resulting complex variable X is retrieved as $X \equiv 2X_{11} + i\left(X_{12} - X_{12}^{\mathrm{T}}\right)$.

We now briefly discuss the formulation of complex SDP problems for which the target is given by real linear coefficient C. Consider $C \in \mathbb{S}^n$, and $X \in \mathbb{H}^n$. Let $X = X^{\mathrm{R}} + iX^{\mathrm{I}}$, where $X^{\mathrm{R}} \in \mathbb{S}^n$ and $X^{\mathrm{I}} \in \mathbb{R}^{n\times n}$. We have:

$$\mathrm{Tr}\,(CX) = \mathrm{Tr}\left(CX^{\mathrm{R}}\right) + i\,\mathrm{Tr}\left(CX^{\mathrm{I}}\right) = \mathrm{Tr}\left(CX^{\mathrm{R}}\right). \tag{97}$$

Since X^{I} is antisymmetric, the Frobenius product of symmetric and antisymmetric matrix is always equal to 0. Thus if C is real and we are interested only in finding the value of the solution, then we can ignore the imaginary part occurring in the problem.

3.6. Slack and surplus variables, mixed problems and equalities

Slackness and complementary slackness are both concepts used in optimization theory, particularly in convex optimization. Slackness refers to the idea that in an optimal solution, some of the inequality constraints are satisfied with equality, i.e. there is no *slack* or excess capacity in the system. The extent to which they diverge from the equality can be expressed as a new PSD variable, which then can be introduced to convert inequality constraints to equality constraints, as elucidated below. The concept of slack variables is commonly used in LP and SDP. Recall that in LP and SDP formulations, the objective function is optimized subject to a set of constraints, where the constraints can be in the form of equalities or inequalities. In the case of linear constraints of the form $Ax \leqslant b$, where A is an $m \times n$ matrix and b is a column vector of length m, introducing a slack variable $\tilde{x}_i$ for each constraint i allows us to convert the inequality constraint into an equality constraint. The idea is to add a non-negative variable $\tilde{x}_i$ to the left-hand side of the ith constraint so that the resulting expression becomes equality. Specifically, if the ith constraint is:

$$a_{i1}x_1 + \cdots a_{in}x_n \leqslant b_i \tag{98}$$

then we can add a slack variable $\tilde{x}_i \geqslant 0$ to obtain an expression that is equivalent to the previous constraint:

$$a_{i1}x_1 + \cdots a_{in}x_n + \tilde{x}_i = b_i. \tag{99}$$

The new variable $\tilde{x}_i$ is called a slack variable because it measures the amount by which the left-hand side of the ith constraint falls short of the RHS b_i. If the left-hand side is already equal to b_i, then $\tilde{x}_i$ is zero.

On the other hand, if we have inequality constraints of the form $Ax \geqslant b$, then we introduce a surplus variable $\tilde{x}_i$ that measures the amount by which the left-hand side of the ith constraint exceeds the RHS b_i. Specifically, we add a non-negative variable $\tilde{x}_i$ to the left-hand side of the ith constraint, so that the resulting expression becomes an equality:

$$a_{i1}x_1 + \cdots a_{in}x_n - \tilde{x}_i = b_i. \tag{100}$$

Therefore, we can see that the use of slack and surplus variables allows us to convert any inequality constraint into an equality constraint, which makes it easier to fit into the canonical form.

On the contrary, complementary slackness, refers to the idea that for an optimal solution, the primal variables and the corresponding dual variables are *complementary*, in the sense that their product is zero. In other words, slackness is a condition that holds between the primal variables (e.g. the decision variables in an LP) and the primal constraints, while complementary slackness is a condition that holds between the primal solution and the dual solution in a convex optimization problem. For instance, when the strong duality of SDP holds, we have $p^* = d^*$, i.e. $C \bullet X = b^{\mathrm{T}} \cdot y = A_i \bullet X \cdot y$. Thus $0 = (C - \sum_{i\in[m]} y_i A_i) \bullet X = \mathrm{Tr}\,[ZX]$. This means that if the solutions of primal and dual SDP problems exist, then strong duality is equivalent to complementary slackness. Summing up, whereas slackness conditions tell us which constraints are active in the optimal solution (i.e. satisfied with equality), the complementary slackness conditions tell us which constraints are binding (i.e. have nonzero dual variables) and which are nonbinding (i.e. have zero dual variables). Both conditions are important for understanding and characterizing optimal solutions in optimization problems.

One often considers the so-called mixed LP-SDP cone. The primal mixed problem in variables $(x_{\mathrm{L}}, X_{\mathrm{S}}) \in \mathbb{R}^{n_{\mathrm{L}}} \times \mathbb{S}^{n_{\mathrm{S}}\times n_{\mathrm{S}}}$ is of the following form:

$$\begin{aligned} &\text{minimize } c_{\mathrm{L}}^{\mathrm{T}} \cdot x_{\mathrm{L}} + C_{\mathrm{S}} \bullet X_{\mathrm{S}} \\ &\text{subject to } \left(A_{\mathrm{L}}^{\mathrm{T}}\right)_{i,:} \cdot x_{\mathrm{L}} + A_{\mathrm{S}i} \bullet X_{\mathrm{S}} = b_i, \text{for } i \in [m] \\ &\qquad x_{\mathrm{L}} \geqslant 0, X_{\mathrm{S}} \succeq 0 \end{aligned} \tag{101}$$

where $A_{\mathrm{L}} \in \mathbb{R}^{n_{\mathrm{L}}\times m}$, $b \in \mathbb{R}^m$, $x_{\mathrm{L}}, c_{\mathrm{L}} \in \mathbb{R}^{n_{\mathrm{L}}}$, $C_{\mathrm{S}} \in \mathbb{S}^{n_{\mathrm{S}}\times n_{\mathrm{S}}}$ and $A_{\mathrm{S}1}, \ldots, A_{\mathrm{S}m} \in \mathbb{S}^{n_{\mathrm{S}}\times n_{\mathrm{S}}}$. The dual mixed problem in variables $(y, z_{\mathrm{L}}, Z_{\mathrm{S}}) \in \mathbb{R}^m \times \mathbb{R}^{n_{\mathrm{L}}} \times \mathbb{S}^{n_{\mathrm{S}}\times n_{\mathrm{S}}}$ is given by the formula:

$$\begin{aligned} &\text{maximize } b^{\mathrm{T}} \cdot y \\ &\text{subject to } c_{\mathrm{L}} - A_{\mathrm{L}} y = z_{\mathrm{L}}, C_{\mathrm{S}} - \sum_{i\in[m]} y_i A_{\mathrm{S}i} = Z_{\mathrm{S}}, \\ &\qquad z_{\mathrm{L}} \geqslant 0, Z_{\mathrm{S}} \succeq 0. \end{aligned} \tag{102}$$

We note that as any LP can be reformulated as SDP, the mixed problems are not more general than the SDP problems. One can see that to embed an LP in SDP it is sufficient to place the n_{L} linear variables on the diagonal of a new SDP variable of size $n_{\mathrm{L}} + n_{\mathrm{S}}$, where n_{S} is the size of the SDP original variable. On the other hand, mixed problems are useful for efficient solver implementations, as the numerical methods needed to solve SDP are more expensive in terms of computational effort than LP. Thus, when stating the problem in the mixed form, one may obtain a reduction of the computational cost of the solver.

We now discuss, how equality constraints are expressed in the canonical form of SDP. Recall that in the primal canonical SDP (80), equalities have the form $A_i \bullet X = b_i$ for $i \in [m]$, where m is one of the two parameters describing the size of the problem. Thus, to add a linear constraint on variables within X we add a new matrix A_i *increasing* size of the problem to $m+1$. On the other hand, if we add a linear constraint in the dual form (81), we can do one of the following. The first possibility is to eliminate one of the variables y_i, e.g. with lower–upper decomposition (LU) or QR decomposition, and thus *reduce* size of the problem to $m-1$. This simplifies the SDP but requires an additional effort of variable elimination, which itself requires some computational resources and is not always implemented. For instance, the elimination is performed in YALMIP when the user passes an option `'removeequalities'` to

the model, as discussed in appendix B.1. The second possibility is to reframe the optimization problem over $(x_{\mathrm{F}}, X_{\mathrm{S}}) \in \mathbb{R}^{n_{\mathrm{F}}} \times \mathbb{S}^{n_{\mathrm{S}} \times n_{\mathrm{S}}}$ for some $n_{\mathrm{F}}, n_{\mathrm{S}} \in \mathbb{N}$ in a form different than the canonical form, see (101), *viz.*

$$\begin{aligned} &\text{minimize } c_{\mathrm{F}}^{\mathrm{T}} \cdot x_{\mathrm{F}} + C_{\mathrm{S}} \bullet X_{\mathrm{S}} \\ &\text{subject to } \left(A_{\mathrm{F}}^{\mathrm{T}}\right)_{i,:} \cdot x_{\mathrm{F}} + A_{\mathrm{S}i} \bullet X_{\mathrm{S}} = b_i, \text{for } i \in [m] \\ &\qquad\qquad X_{\mathrm{S}} \succeq 0. \end{aligned} \tag{103}$$

where $A_{\mathrm{F}} \in \mathbb{R}^{n_{\mathrm{F}} \times m}$, $b \in \mathbb{R}^m$, $x_{\mathrm{F}}, c_{\mathrm{F}} \in \mathbb{R}^{n_{\mathrm{F}}}$, $C_{\mathrm{S}} \in \mathbb{S}^{n_{\mathrm{S}} \times n_{\mathrm{S}}}$ and $A_{\mathrm{S}1}, \ldots, A_{\mathrm{S}m} \in \mathbb{S}^{n_{\mathrm{S}} \times n_{\mathrm{S}}}$, and there is no constraint on x_{F}. The dual problem of (103) in variables $(y, z_{\mathrm{F}}, Z_{\mathrm{S}}) \in \mathbb{R}^m \times \{0\}^{n_{\mathrm{F}}} \times \mathbb{S}^{n_{\mathrm{S}} \times n_{\mathrm{S}}}$ is of the following form:

$$\begin{aligned} &\text{maximize } b^{\mathrm{T}} \cdot y \\ &\text{subject to } c_{\mathrm{F}} - A_{\mathrm{F}} y = z_{\mathrm{F}}, \\ &\qquad\qquad C_{\mathrm{S}} - \sum_{i \in [m]} y_i A_{\mathrm{S}i} = Z_{\mathrm{S}}, Z_{\mathrm{S}} \succeq 0. \end{aligned} \tag{104}$$

The forms (103) and (104) are sometimes called the standard form with free variables [135]. Note that the trivial cone $\{0\}^{n_{\mathrm{F}}}$ appearing in (104) is the dual cone of $\mathbb{R}^{n_{\mathrm{F}}}$ appearing in (103). Since this implies $z_{\mathrm{F}} = 0$, the first condition in (104) is equivalent to $A_{\mathrm{F}} y = c_{\mathrm{F}}$, providing a way to express equality constraints on the dual variable y. This possibility of treating the equality constraint requires the solver to be able to deal with free variables in the primal problem. This is beyond the capabilities of the usual IPMs as discussed in section 3.8, see also [208]. The standard way of dealing with free variables, implemented in SeDuMi and SDPT3, is to represent a vector $x_{\mathrm{F}} \in \mathbb{R}^{n_{\mathrm{F}}}$ as a difference of two vectors $x_{(+)}, x_{(-)} \in \mathbb{R}_+^{n_{\mathrm{F}}}$ as $x_{\mathrm{F}} = x_{(+)} - x_{(-)}$. This guides us to the third possibility of representing equality constrain $(A_{\mathrm{F}})_{i,:} \cdot y = c_{\mathrm{F}i}$, as two inequalities:

$$\begin{aligned} (A_{\mathrm{F}})_{i,:} \cdot y &\geqslant c_{\mathrm{F}i} - \epsilon, \\ (A_{\mathrm{F}})_{i,:} \cdot y &\leqslant c_{\mathrm{F}i} + \epsilon, \end{aligned} \tag{105}$$

for some small ϵ. In consequence, equalities even in the dual form are increasing the complexity. For instance, YALMIP allows the user to specify, how to treat the explicitly stated equality constraints with the mentioned option `'removeequalities'`, or, in case the chosen solver requires it, YALMIP does it automatically with the function `solveequalities`. A detailed discussion of conversion the problems (103) and (104) to the canonical form (80) and (81) is provided in [175].

3.7. Schur complement and submatrices

Consider a partition of a square matrix $M \in \mathbb{C}^{(n_1+n_2)\times(n_1+n_2)}$ to submatrices, *viz.* $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$, where $A \in \mathbb{C}^{n_1 \times n_1}$ and $D \in \mathbb{C}^{n_2 \times n_2}$ are square matrices. Recall that the principal submatrix of a square matrix is the special case of a submatrix where the same rows and columns are removed. Thus A and D are principal submatrices. It is easy to see that a principal submatrix of a PSD matrix is also PSD. For instance consider an arbitrary vector $v \in \mathbb{C}^{n_1+n_2}$ with non-zero entries only in the first n_1 positions, and a vector $v' \in \mathbb{C}^{n_1}$ with entries equal to the first n_1 entries of v. Obviously, since $v^\dagger M v \geqslant 0$ it also holds that $v'^\dagger A v' \geqslant 0$.

Assume the submatrix D to be invertible. For numerical implementations, it would also be desirable for D to be well-conditioned to give accurate results under finite-precision arithmetic upon inverting it. The Schur complement of block D is given by $A - BD^{-1}C$ and denoted M/D. Similarly, $M/A \equiv D - CA^{-1}B$ [41, 115, 158, 281]. We have $M \succeq 0 \implies M/D \succeq 0$; and for symmetric M (i.e. $B = C^{\mathrm{T}}$):

$$M \succ 0 \iff A, M/A \succ 0, \text{ and} \tag{106a}$$

$$A \succ 0 \implies [M \succeq 0 \iff M/A \succeq 0]. \tag{106b}$$

Obviously, the same holds for D. The Schur complement method [343] allows reducing a large system of equations to a smaller one, involving only a subset of variables, e.g. $M\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$, by solving two simpler equations, namely $(A - BD^{-1}C)x_1 = y_1 - BD^{-1}y_2$, and then $Cx_1 + Dx_2 = y_2$. The reason why this method is useful is that one usually needs $O(n^3)$ operations to solve linear equations with n variables, and thus it is profitable to decompose the initial equation into two smaller equations.

Schur complement is also a tool for introducing the following useful trick. Consider an LMI:

$$\begin{bmatrix} t & c(x) \\ c^{\mathrm{T}}(x) & F(x) \end{bmatrix} \succeq 0, \tag{107}$$

where $x \in \mathbb{R}^k$ for some k, $F(x) \in \mathbb{R}^{m\times m}$ is a linear matrix expression of the form (78*b*) for some m, $c : \mathbb{R}^k \to \mathbb{R}^m$ is a linear function, and t is a positive number. From (106*b*) it follows

$$t \geqslant c(x) \cdot F^{-1}(x) \cdot c^{\mathrm{T}}(x). \tag{108}$$

The expression (108) allows for expression fairy generic non-linear constraints as LMIs, e.g. taking $k = 2$, $m = 1$ with $c(x) = x_1$ and $F(x) = x_2$ we get the constraint $t \geqslant \frac{x_1^2}{x_2}$. Similarly, from Schur lemma it follows that for $A, B, R \in \mathbb{H}^n$:

$$\begin{bmatrix} A & R \\ R & B \end{bmatrix} \succeq 0 \Leftrightarrow B \succeq RA^{-1}R \Leftrightarrow \left[A^{-1/2}BA^{-1/2}\right]^{1/2} \succeq A^{-1/2}RA^{-1/2} \Leftrightarrow A\#B - R \succeq 0, \tag{109}$$

where the $1/2$-weighted matrix geometric mean is given by (31).

3.8. How does a solver use IPMs?

Even though the topic of implementation of IPM is not specific to quantum information, from our experience, it is useful to have at least a general understanding of how actually a solver is deriving its results. This helps to identify potential problems, estimate the difficulties, and interpret the outputs of the solver. An important concept in path-following IPMs is the *central path*, which consists of a set of feasible solutions (X, y, Z) parameterized by a non-negative variable ν [124, 178, 179, 218, 219, 256, 345]. The central path is defined by the following conditions:

$$X, Z \succeq 0 \quad \text{(PSD matrices)}, \tag{110a}$$

$$A_i \bullet X = b_i \text{ for } i \in [m] \quad \text{(linear constraints)}, \tag{110b}$$

$$C - \sum_{i\in[m]} y_i A_i = Z \quad \text{(dual feasibility)}, \tag{110c}$$

$$XZ = \nu\mathbb{1} \quad \text{(approx. complementary slackness)}, \tag{110d}$$

where $\{A_i\}$, b_i, and C are problem-specific matrices for the canonical formulation described in section 3.2.1. The central path captures a family of solutions that gradually approaches the optimal solution as ν increases. By following this path, usually employing the concept of the Newton steps, IPMs efficiently navigate the feasible region of the SDP problem toward the optimal solution. Equivalently (110) can be written in Kronecker-canonical form, see section 3.2.4:

$$\mathcal{A}^{\mathrm{T}}x = b \tag{111a}$$

$$c - \mathcal{A}y = z, \tag{111b}$$

$$XZ = \nu\mathbb{1}, \tag{111c}$$

together with $X, Z \succeq 0$. Further in the text, the expression $\mathrm{Tr}(ZX)$ is referred to as the *gap*. Note that the strong duality of an SDP problem implies the complementary slackness, see section 3.6, stating that the optimal primal and dual variables are orthogonal, i.e.

$$\mathrm{Tr}\,(ZX) = 0, \tag{112}$$

meaning that the gap is equal to 0. On the other hand, it should be stressed that the gap (112) and the duality gap are closely related but different terms. The former is defined in terms of the primal and dual variables X and Z, even if they do not provide a feasible solution, i.e. even if they do not satisfy the conditions in (80) and (81). The latter is defined as (49) and expresses the difference between the optimal primal and dual solutions of the problem. We note here that practical implementations of SDP solvers usually find solutions that are not feasible in a strict sense. Instead, the solutions satisfy the condition from (80) and (81) only with some accuracy. Here we discuss the expressions we use further in this work to evaluate primal and dual infeasibility. See [214] for more details on the issue of infeasibility norms. Let $c \equiv \mathrm{vec}(C)$, $x \equiv \mathrm{vec}(X)$ and $z \equiv \mathrm{vec}(Z)$, as in the Kronecker-canonical form, see section 3.2.4. Let us define the following terms, *viz.* the residuals for feasibility conditions in (80) and (81), *see* (111*a*) and (111*b*):

$$r_p \equiv b - \mathcal{A}^{\mathrm{T}}x \in \mathbb{R}^m, \tag{113a}$$

$$r_d \equiv c - \mathcal{A}y - z \in \mathbb{R}^{n^2}. \tag{113b}$$

Using the above conditions (111) we get that the Newton step $(\Delta X, \Delta y, \Delta Z)$ is supposed to satisfy the following formulae:

$$\begin{aligned} \mathcal{A}^{\mathrm{T}}\Delta x &= r_p, \\ \mathcal{A}\Delta y + \Delta z &= r_d. \end{aligned} \tag{114}$$

If the method assumes that both r_p and r_d are zero, $r_p = r_d = 0$, it is referred to as a feasible IPM. Otherwise, it is considered an infeasible IPM. The conditions (114) are supposed to iteratively bring the variables (X, y, Z) closer to the feasibility constraints, (111*a*) and (111*b*). On the other hand, note that those two equations do not determine fully the solution; yet we still need to consider (111*c*) somehow. At the same time, from the strong duality, we get that the optimal solution, (X^*, y^*, Z^*) has the property that it is on the central path at the point with $\nu = 0$, i.e. the gap is 0, see section 3.3. Thus, the Newton step should not only ensure feasibility but also reduce the gap between primal and dual solutions. Actually, the condition

imposed on the Newton step from the condition (111*c*) is most problematic for another reason. The problem arises from the fact that the matrices $X^{(i)}$ and $Z^{(i)}$, where i stands for the current iteration, possibly do not commute. For this reason, the following form of conditions on the target of the Newton step is imposed:

$$\Theta_\nu(X,Z) = \nu\mathbb{1} \in \mathbb{R}^{n\times n}, \tag{115}$$

where $\Theta_\nu(X,Z)$ is a symmetrization of XZ; we discuss a couple of symmetrizations below. The algorithm stops when residual norms and the gap $|\mathrm{Tr}(XZ)|$, are all less than the specified threshold. Examples of the primal and dual residual norms used e.g. in [211] are $\frac{1}{1+|b|_\mathrm{F}}\left|b-\mathcal{A}^\mathrm{T}x\right|_\mathrm{F}$, and $\frac{1}{1+|c|_\mathrm{F}}\left|c-\mathcal{A}y-z\right|_\mathrm{F}$, respectively, where we recognize normalized norms of the expressions (113*a*) and (113*b*).

The problem of efficiently solving the Newton system (111) numerically is discussed in detail in chapter 5 of [211]. It is worth noting that by employing the Schur complement method, see section 3.7, we can reduce the system of equations to a smaller size. Solving a linear system of k equations typically requires $O(k^3)$ floating point operations (FlOps). In the initial system, we have $k = 2n^2 + m$ equations, where $n > m$. Consequently, obtaining the solution requires $O(n^6)$ FlOps. However, the Schur complement equation has a size of m and only requires $O(m^3)$ FlOps.

Now let us examine the search directions discussed earlier. Consider the condition stated in (111*c*), namely $XZ = \nu\mathbb{1}$. When we take a step, we have the equation $(X+\Delta X)(Z+\Delta Z) = \nu\mathbb{1}$, or, neglecting the second-order term $\Delta X\Delta Z$, we have $\Delta XZ + X\Delta Z = \nu\mathbb{1} - XZ$. It is required that the steps ΔX and ΔZ be symmetric. The second equation in (114) reveals that ΔZ is always symmetric, given the numerical precision. However, the same cannot be said for ΔX, which may not be symmetric. Consequently, there is a necessity to symmetrize (111*c*). For instance, in 1998 the following natural symmetrization of (111*c*), called AHO, was introduced by Alizadeh *et al* [10]:

$$\Theta_\nu^{\mathrm{AHO}}(X,Z) \equiv \left(\frac{1}{2}(XZ+ZX) = \nu\mathbb{1}\right). \tag{116}$$

While the search direction defined by this symmetrization holds historical significance, it is no longer widely utilized by the majority of SDP solvers. For instance, recent implementations of SDPT3 [310] have omitted this particular search direction. Another search direction, called HKM, has been introduced independently by Helmberg *et al* [141], Kojima *et al* [180] and Monteiro [216], and is used by many modern solvers. The HKM has the following primal form:

$$\Theta_\nu^{\mathrm{HKM,primal}}(X,Z) \equiv \left(Z^{\frac{1}{2}}XZ^{\frac{1}{2}} = \nu\mathbb{1}\right), \tag{117}$$

and the dual form $\Theta_\nu^{\mathrm{HKM,dual}}(X,Z) \equiv (X^{\frac{1}{2}}ZX^{\frac{1}{2}} = \nu\mathbb{1})$. A third important search direction is the Nesterov and Todd [235, 236], or NT, direction. Its definition is more involved than in the cases of AHO and HKM. To define it, we consider a matrix W satisfying $W^{-1}XW^{-1} = Z$. With the aid of the matrix W, the symmetrization of (111*c*) can be formulated as $\frac{1}{2}(W^{-\frac{1}{2}}XZ + ZXW^{-\frac{1}{2}}) = \nu W^{-1}$.

The work of Monteiro and Zhang (MZ) from 1998 [217, 220, 344], introduced the following family of search directions, which includes the three mentioned directions, *viz.* AHO, HKM, and NT. The linear transformation of MZ is given by the following formula

$$H_\mathrm{P}(M) \equiv \frac{1}{2}\left(PMP^{-1} + P^{-T}M^\mathrm{T}P^\mathrm{T}\right), \tag{118}$$

with P being an invertible matrix. Next, we define $\Theta_\nu(X,Z) \equiv (H_{\mathrm{P}}(XZ) = \nu\mathbb{1})$. By selecting different invertible matrices P, we can observe that each of the search directions discussed can be obtained. Specifically, when $P = \mathbb{1}$, we obtain the AHO search direction. On the other hand, choosing $P = Z^{\frac{1}{2}}$ and $P = X^{-\frac{1}{2}}$ leads to the primal and dual HKM search directions, respectively. Furthermore, if we consider an invertible matrix P satisfying the condition $P^{\mathrm{T}}P = W^{-1}$, we obtain the NT search direction.

Once the search direction has been determined, the next step in an IPM is to select the appropriate step length in that direction. This step length is crucial for ensuring the feasibility of the iterates. Specifically, the IPM chooses a step-length, represented by a pair of constants α and β in the range $(0,1]$, such that the following conditions are satisfied:

$$X + \alpha\Delta X \succeq 0, \tag{119a}$$

$$Z + \beta\Delta Z \succeq 0. \tag{119b}$$

It is important to note that while Newton's method is used to determine the direction, the iterative solver does not necessarily take the full Newton step. To provide further clarity, it is worth mentioning the following points. According to [308], the value of t that solves the optimization problem defined by M and ΔM can be obtained in the following way. The objective is to maximize t subject to the constraint $M + t\Delta M \succeq 0$. The formula to compute this step length is given by $\max\left(\mathrm{eig}(C^{-\mathrm{T}}\Delta M C^{-1})\right)$, where C represents the Cholesky decomposition of M. By using this formula, we can determine the appropriate step lengths α and β for the IPM. It is important to note that convergence proofs of IPM algorithms often necessitate that at each step, the current solution is in close proximity (in some defined sense) to the central path. This requirement imposes additional constraints on the step length. However, for the sake of efficiency, these constraints may be relaxed, at the cost of losing the guarantee of convergence. Furthermore, it is worth highlighting that a common cause of failure in IPM algorithms arises during the Schur complement matrix decomposition. This issue tends to occur when the iterates approach the optimal solution and the primal variable and dual slack variables become nearly orthogonal, resulting in $\mathrm{Tr}(XZ) \approx 0$. In such cases, the Schur complement matrix may become ill-conditioned, leading to numerical instability or inaccurate results.

3.9. *Solver internal mechanisms: predictor–corrector, warm start, problem structure*

Now, we will discuss several internal mechanisms employed by solvers to provide a deeper understanding of their functionality and potential challenges or performance gains. We begin by introducing the concept of predictor–corrector, shedding light on its significance and usage within solvers. Additionally, we explore the application of perturbations to iterates, which can prove beneficial when tackling numerical issues while solving complex problems with stringent constraints. Furthermore, we emphasize the importance of leveraging the structure of specific problems to unlock potential solver optimizations and fine-tune performance. By highlighting these aspects, we aim to provide insights into the diverse approaches and strategies that can be employed in solver implementations.

As we mentioned above, when discussing the constraint (111*c*), we expect an SDP solver to approach the value $\nu = 0$. But here a question arises: should this be done immediately or gradually? And in the latter case: *how* gradually? One of the most popular answers has been given by Mehrotra [205], who introduced the so-called predictor–corrector mechanism. The predictor–corrector method is a powerful numerical technique widely used in solvers for solving complex mathematical problems efficiently. This iterative algorithm combines two essential steps: prediction and correction. In the prediction step, an approximate solution is computed based

on the available information. This predicted solution is then refined in the correction step by incorporating additional calculations or adjustments to improve its accuracy and convergence toward the proper solution.

The first step of the predictor–corrector mechanism is the calculation of the predictor direction, sometimes referred to as an *affine scaling direction* [292], which employs an aggressive strategy to advance along the central path. In this approach, the target for the Newton step is set to $\nu = 0$ in (115), indicating that the predictor step aims to approach the optimal solution. Let $(\delta X, \delta y, \delta Z)$ represent the predictor step, with the step-length determined by $\alpha_{\rm P}$ and $\beta_{\rm P}$. These quantities are subsequently used to compute the value of ν that will serve as the objective for the Newton step direction in the subsequent iteration of the SDP solver. The predictor step itself is not taken but rather employed to derive a second-order correction for the corrector, or *centering*, direction. The actual procedure of calculation of the new value of ν is quite complicated. We refer reader to section 6.5 of [211] for more details. Just to provide some taste of the method we mention that e.g. in SDPT3 solver the value $\alpha_{\rm P}$ is upper bounded by a certain user-specified parameter gam, and the following formula for the corrector step [307, 310]:

$$\frac{1}{n}\mathrm{Tr}\,(XZ)\cdot\left(\frac{\mathrm{Tr}\,((X+\alpha_{\rm P}\delta X)\,(Z+\beta_{\rm P}\delta Z))}{\mathrm{Tr}\,(XZ)}\right)^{\text{expon\_used}}, \tag{120}$$

where expon_used is a variable whose value is either a constant or is determined with some other algorithm based on another user-specified parameter `expon`. Next, in the corrector part of the iteration, one sets the new value of ν and calculates the Newton step $(\Delta X, \Delta y, \Delta Z)$ for the second time, with a different RHS. Often, to the goal $\nu\mathbb{1}$ in (111*c*), an additional term $F(X, Z, \delta X, \delta Z)$ with a second order correction is added. Finally, the step-lengths $\alpha_{\rm C}$ and $\beta_{\rm C}$ for the corrector step are computed to ensure the preservation of positivity. In the subsequent iteration, the IPM algorithm sets the following:

$$X \equiv X + \alpha_{\rm C}\Delta X, \tag{121a}$$
$$y \equiv y + \beta_{\rm C}\Delta y, \tag{121b}$$
$$Z \equiv Z + \beta_{\rm C}\Delta Z. \tag{121c}$$

The concept of warm start in solvers plays a crucial role in optimizing computational efficiency and reducing solution times for various optimization problems. Warm start refers to the technique of providing an initial feasible solution to a solver, obtained as a guess, or some generic scheme, or has been calculated from an already solved similar or related problem. Rather than starting the solver from scratch, the warm start approach utilizes the information from a previously solved problem to accelerate the convergence of subsequent iterations. By leveraging this initial solution, warm start techniques can significantly improve the overall performance of solvers. We will only briefly overview the warm start in solvers, and refer to section 6.3 of [211] for a detailed discussion. It has been observed that it is desirable for the initial iterate to have the magnitude of at least the same order as the optimal solution. The following method of cold-start is used in SDPT3 [310]: $X^{(0)} \equiv \xi\,\mathbb{1}$, $Z^{(0)} \equiv \eta\mathbb{1}$, and $y^{(0)}$ is the zero vector of the relevant dimension, where $\xi \equiv \max(10, \sqrt{n}, n\max_{i\in[m]} \frac{1+|b_i|_{\rm F}}{1+|A_i|_{\rm F}})$ and $\eta \equiv \max(10, \sqrt{n}, |C|_{\rm F}, \max_{i\in[m]} |A_i|_{\rm F})$. In [211] we proposed and analyzed warm start strategies for NPA problems.

Another mechanism used in SDP solvers, which proved to be advantageous to certain scenarios, is the utilization of perturbations during iterations. Perturbations involve introducing slight modifications to the current iterates under specific conditions. The primary purpose of

employing perturbation strategies is to prevent solvers from becoming stuck near the boundary of e.g. the PSD cone. While the primary objective of perturbations is not to reduce the number of iterations or CPU time, but rather to circumvent failures, it has been observed that in instances where the solver does not encounter failures, the most efficient solution tends to be one without perturbations [211]. By incorporating perturbations into the iterative process, the solver can navigate away from critical regions and explore a wider solution space, potentially avoiding numerical instabilities or convergence issues. While we will not delve into the details of perturbation strategies, we present a simple example in algorithm 1 to illustrate their form. The purpose of this strategy is to strike a balance between improving the duality gap and ensuring feasibility. In our observations of SDPs occurring in NPA [211], we have found that the reduction of the size of the gap often presents the greatest challenge. Specifically, the iterates tend to reach the feasibility threshold relatively quickly, and the majority of iterations are dedicated to reducing the gap. It is important to note that the efficiency and effectiveness of perturbation strategies may vary depending on the problem and solver being employed. In cases where the solver encounters failures, perturbations can provide a crucial mechanism to overcome such issues and continue the iterative process. However, it is worth noting that in scenarios where the solver operates smoothly without failures, perturbations may introduce additional computational overhead without providing substantial benefits in terms of solution quality or convergence speed. Overall, the use of perturbations in iterations offers a valuable approach to enhance the robustness and reliability of solvers when tackling problems involving the PSD cone. Nevertheless, the decision to incorporate perturbations should be made considering the specific problem characteristics, solver behavior, and desired trade-offs between reliability and computational efficiency.

Algorithm 1. Example of a perturbation of the iteratively improved solution in an SDP solver.

if $\text{gap} > 100 \cdot \epsilon_{\text{P}}$ **then**
  $X \leftarrow X + 0.01 \cdot t_p \cdot \mathbb{1}$
end if
if $\epsilon_{\text{P}} > 100 \cdot \epsilon_{\text{D}}$ **then**
  $Z \leftarrow Z + 0.1 \cdot t_p \cdot \mathbb{1}$
end if

The last mechanism we mention is exploiting the specific structure of the problem by a solver to enhance its performance. It allows for tailored solver optimizations that can significantly enhance performance. By exploiting the problem structure, solvers can take advantage of inherent characteristics such as symmetry, sparsity, or specific constraints to reduce computational complexity. This approach enables the solver to focus computational resources on the most relevant parts of the problem, leading to faster convergence and more efficient solutions. Additionally, by understanding the problem structure, solvers can employ specialized algorithms and techniques that are specifically designed to leverage the problem's unique properties, further improving solution quality and computational efficiency. For instance, in [211] we have proposed a special data format to improve the performance of operations taken by a solver dedicated to the problems occurring in the dual formulation of NPA problems, where such properties as the sparsity and entries pattern was taken into account.

One of the first generic approaches exploiting sparsity patterns was given by Fujisawa *et al* [111], where the methods to leverage the sparsity of the problem matrices to improve efficiency were explored. The computation of the Schur complement matrix, as discussed in section 3.7, is often recognized as the most computationally intensive step in solving an SDP problem. One strategy they employ to support the Schur complement calculations is reordering

the matrices $\{A_i\}_i$. They demonstrate that the most effective reordering is one that arranges the matrices in a non-increasing order based on the number of nonzero elements f_i. Additionally, due to the symmetry of the Schur complement B, only entries $B_{i,j}$ with $j \geqslant i$ need to be directly evaluated. Fujisawa *et al* presented three different methods for computing the element $B_{i,j}$, with the choice depending on the sparsity patterns of matrices A_i and A_j. For highly sparse matrices, they utilize the formula $B_{i,j} = \sum_{a,b,c,d\in[n]} (A_i)_{c,d} W_{c,a} W_{d,b} (A_j)_{a,b}$ to compute the Schur complement. The calculation of this quantity requires $(2f_i + 1)f_j$ multiplications. The work [318] introduces the so-called *correlative sparsity pattern graph*, which relates to a certain sparse structure in the objective and constraint polynomials of unconstrained and inequality-constrained sparse polynomial optimization problems. The graph is used to obtain sets of the supports for SoS polynomials and get the improved performance of relevant SDPs. Sparsity patterns that are more specific to non-commutative polynomial optimization, particularly relevant to quantum information problems, were investigated in [172]. The method suggested partitioning the input variables into cliques based on the so-called *correlative sparsity pattern* exhibited by the polynomials present in the objective function and constraints. In [320] another particular type of sparsity occurring in the input data for large-scale sparse noncommutative polynomial optimization problems, called *term sparsity* is introduced.

4. Constructions of SDP useful for quantum information

In this section, we provide an overview of popular constructions which are used as building blocks for more complicated optimization problems used in quantum information. In section 4.1 we discuss semidefinite representations of semialgebraic functions which allow e.g. to express approximations of various types of quantum entropies as SDPs. In section 4.2 we provide an overview of the separability criteria of quantum states originating from the famous PPT criterion. Next, in section 4.3 we describe the Choi–Jamiołkowski isomorphism and highlight its relation to PSD constraints. Then, in section 4.4 the SoSs decomposition important for polynomial optimization is discussed, and in section 4.5 we describe the famous Lovász θ function. Afterward, we will discuss the application of moment matrices in the realm of quantum information and explore different aspects of their use. These include section 4.6 which discusses the relationship between correlation matrices and the so-called moment matrices, together with the NPA hierarchy which is a method for optimization over non-commuting variables; sections 4.7 and 4.8, which investigates three distinct methods for optimizing probability distributions or behaviors subject to dimension constraints.

4.1. Semidefinite representations of semialgebraic sets

Spectrahedron is defined as a geometric object that can be characterized as a solution set of an LMI (78*a*) or, in other words, it is an intersection of the PSD cone with a linear affine subspace. This representation allows spectrahedra to capture the feasible regions of SDPs. When spectrahedra are subjected to linear or affine transformations, the resulting shapes are referred to as *projected spectrahedra*, or *spectrahedral shadows*, or *SDP representable sets* [145]. These projected spectrahedra retain the convexity property and also belong to the class of semialgebraic sets. It is worth noting that while every spectrahedral shadow is a convex semialgebraic set, but the converse statement, which was posed as a question by Nemirovski in [231], previously conjectured [145] to be true until 2017 [280], does not hold in general. This means that not all convex semialgebraic sets can be represented as spectrahedra. The notion of spectrahedra was introduced in [267]; see [317] for an overview, and [28] for a detailed discussion of

their relation to the geometry of quantum states. An example of a spectrahedron of particular interest to quantum information is the spectraplex, which is the set of all PSD matrices in a given dimension with trace 1, i.e. the normalized quantum states.

To be more specific, we say that a set $\mathcal{S} \subseteq \mathbb{R}^n$ is LMI representable or has an LMI representation [147] if there exists a set of $n+1$ symmetric $n \times n$ matrices $\{A_i\}_{i\in\{0\}\cup[n]} \subset \mathbb{S}^n$ such that

$$\mathcal{S} = \left\{ x \in \mathbb{R}^n : A_0 + \sum_{i\in[n]} A_i x_i \succeq 0 \right\}. \tag{122}$$

The question of which closed convex sets can be SDP represented for $n=2$ was first posed by Parrilo and Sturmfels in [247]. Suppose that for some $N \in \mathbb{N}_+$ there exist the following two sets of symmetric $n \times n$ matrices: $\{A_i\}_{i\in\{0\}\cup[n]} \subset \mathbb{S}^n$ and $\{B_j\}_{j\in[N]} \subset \mathbb{S}^n$, such that $\mathcal{S}$ is a projection to $\mathbb{R}^n$ of the set

$$\hat{\mathcal{S}} = \left\{ (x,u) \in \mathbb{R}^{n+N} : A_0 + \sum_{i\in[n]} A_i x_i + \sum_{j\in[N]} B_j u_j \succeq 0 \right\}, \tag{123}$$

so that $\mathcal{S} = \left\{ x \in \mathbb{R}^n : \exists_{u\in\mathbb{R}^N} (x,u) \in \hat{\mathcal{S}} \right\}$. Then, we say that $\mathcal{S}$ is *SDP representable*, or has an *SDP representation* or a *lifted LMI representation*, or is a spectrahedral shadow [280]. For instance, from (109) it follows that the hypograph of the matrix geometric mean, *viz.* $\mathbf{hyp}[\#] = \left\{ (X,Y,R) \in \mathbb{H}^n_{++} \times \mathbb{H}^n_{++} \times \mathbb{H}^n : X\#Y - R \succeq 0 \right\}$, has an SDP representation. As shown in [99, prop. 1], for a odd $p \in \mathbb{N}_+$, and $l \in \mathbb{N}_+$, with $p < 2^l$, it holds that

$$\mathbf{hyp}\left[\#_{p/2^l}\right] = \left\{ (X,Y,R_l) : \exists_{(R_i)_{i\in[l-1]}\subset\mathbb{H}^n}, \forall_{i\in[l]} \begin{bmatrix} X\#_{m_i}Y & R_i \\ R_i & R_{i-1} \end{bmatrix} \succeq 0, R_0 = Y \right\}, \tag{124}$$

where $(m_i)_{i\in[l]}$ is the binary expansion of $p/2^l$, i.e. $p/2^l = \sum_{i\in[l]} m_i/2^{l-i+1}$ and $m_0 = 0$. Thus, there exist an SDP representation of $\mathbf{hyp}[\#_{p/2^l}]$ consisting of l LMI, each of size $2n$ by $2n$.

A necessary condition for a set to be LMI representable is to be convex and basic closed semialgebraic. If a set is SDP representable, then it might not be basic closed semialgebraic, but it must be convex semialgebraic. In [145, 146] sufficient conditions for SDP representability were given. It was also shown that the set of all SDP representable sets is closed under taking linear images or preimages, finite intersections, or convex hulls of finite unions [145, 238]. In [237] it was shown that the interior of an SDP representable set is again an SDP representable set. The result of [279] was that closed convex hulls of one-dimensional semialgebraic sets are also SDP representable. The seminal work [245] showed how to construct a complete family of SDPs of polynomial size, which can be used to prove the infeasibility of a finite set of polynomial constraints.

Since the feasible set of SDP is a semialgebraic set, SDPs cannot be directly used to model non-semialgebraic sets and functions, even if they are convex. The work [100] provided a method to approximate certain useful non-semialgebraic sets with SDP representations of relatively small size. Consider a non-semialgebraic concave function $g : \mathbb{R} \to \mathbb{R}$. Suppose it has an integral representation $g(x) = \int_0^1 f_t(x)\mathrm{d}t$, and that the integral can be approximated, e.g. using one of the Gauss quadratures [263], and then $g(x) \approx r_m(x) \equiv \sum_{j\in[m]} w_j f_{t_j}(x)$, where the weights w_j and nodes t_j depend on the quadrature. The quantity m that defines how many terms occur in the quadrature is called the order of the quadrature. The order of the quadrature determines the accuracy of the approximation, with higher orders resulting in more

accurate approximations. For a function of particular interest, the logarithm, we have $\log(x) = \int_0^1 \frac{x-1}{t\cdot(x-1)+1}\mathrm{d}t$. For any fixed t, the hypograph of the concave function $f_t(x) \equiv \frac{x-1}{t\cdot(x-1)+1}$ has an SDP representation, *viz.* $f_t(x) \geqslant r$ if and only if $\begin{bmatrix} x-1-r & -\sqrt{tr} \\ -\sqrt{tr} & 1-rt \end{bmatrix} \succeq 0$. One can show that the matrix hypograph of f_t for any $t \in [0,1]$ has the following SDP representation [100, proposition 2]:

$$\begin{aligned} \mathbf{hyp}\, f_t &= \left\{(X,R) \in \mathbb{H}^n_{++} \times \mathbb{H}^n : (X-\mathbb{1})\cdot\left[t\cdot(X-\mathbb{1})+\mathbb{1}\right]^{-1} \succeq R\right\} \\ &= \left\{(X,R) \in \mathbb{H}^n_{++} \times \mathbb{H}^n : \begin{bmatrix} X-\mathbb{1}-R & -\sqrt{tR} \\ -\sqrt{tR} & \mathbb{1}-tR \end{bmatrix} \succeq 0\right\}. \end{aligned} \tag{125}$$

The approximation $\log x \approx r_m(x)$ is most accurate for small values of x. Since we have $\log(x) = \frac{1}{h}\log(x^h)$, it is often beneficial to use another approximation, *viz.* $r_{m,k}(x) \equiv 2^k r_m(x^{1/2^k})$. It can be shown that $r_{m,k}$ is operator concave [100, proposition 3] and $\mathbf{P}_{r_{m,k}}[Y,X] = 2^k \mathbf{P}_{r_m}[X\#_{2^{-k}}Y, X]$ [100, equation (18)]. What is more, consider $V, X \in \mathbb{H}^n_{++}$ and $R \in \mathbb{H}^n$. It holds [100, p 272] $R' \succeq \mathbf{P}_{-r_m}[Y,X]$ if and only if there exist $(R'_i)_{i\in[m]} \subset \mathbb{H}^n$ such that $R' = \sum_{i\in[m]} R'_i$ and $\forall_{i\in[m]} \begin{bmatrix} Y-X+R'_i & \sqrt{t_i}R'_i \\ \sqrt{t_i}R'_i & X+t_iR'_i \end{bmatrix} \succeq 0$.

From (30) and $\log x \approx r_{m,k}(x)$ we can expect $S(X|Y) \approx \mathbf{P}_{(-r_{m,k})}[Y,X]$ and thus is some sense:

$$\begin{aligned} \mathbf{epi}\, S(X|Y) \approx K^n_{m,k} &\equiv \left\{(X,Y,R') \in \mathbb{H}^n_+ \times \mathbb{H}^n_+ \times \mathbb{H}^n : R' \succeq \mathbf{P}_{(-r_{m,k})}[Y,X]\right\} \\ &= \left\{(X,Y,R') \in \mathbb{H}^n_+ \times \mathbb{H}^n_+ \times \mathbb{H}^n : R' \succeq -2^k \mathbf{P}_{r_m}[X\#_{2^{-k}}Y, X]\right\}. \end{aligned} \tag{126}$$

This, together with the SDP representation (124), shows that the set $K^n_{m,k}$ has an SDP representation [100, theorem 3]. This, in consequence, allows optimizations over quantum entropies [46, 98] using SDP.

4.2. DPS conditions of separability

The DPS [86, 87] method is a powerful technique used to determine the separability of multipartite quantum states, by providing a hierarchy of SDP relaxations that approximate the separability conditions. To be more specific, DPS introduces a hierarchy of conditions involving partial transpositions allowing for a test of separability, with strength increasing with the hierarchy level. This approach is relevant also for studying the separability of multipartite quantum systems with more than two parties. The DPS method utilizes a series of SDP relaxations, where each relaxation, pertaining to higher levels, introduces additional variables and constraints. Thus, at each level of the hierarchy, the DPS method formulates an SDP. The PSD conditions play a crucial role in these relaxations, as they express the constraint that the obtained solutions are physically valid quantum states. Roughly speaking, in the DPS method for a given state $\rho_{\mathcal{AB}}$ one asks whether there exist a hierarchy of symmetric extensions, i.e. a family of states $\rho_{\mathcal{AB}_1\cdots\mathcal{B}_N}$ defined for any N, such that $\forall_{i\in[N]}\rho_{\mathcal{AB}} = \mathrm{Tr}_{\mathcal{B}_j:j\neq i}[\rho_{\mathcal{AB}_1\cdots\mathcal{B}_N}]$. It happens that the state $\rho_{\mathcal{AB}}$ is separable if and only if such a hierarchy exists for each natural N. The state $\rho_{\mathcal{AB}}$ is separable if and only if a hierarchy of symmetric extensions exists for every natural number N. For a given fixed value of N, the task of verifying the existence of a symmetric extension is equivalent to an SDP. Consequently, an algorithm can be devised by incrementally examining the extendability condition for increasing values of N. This algorithm is guaranteed to terminate if the initial state $\rho_{\mathcal{AB}}$ is entangled, thus it detects the non-separability. However, if the state is separable, the algorithm will continue indefinitely without termination.

Let us provide a more detailed overview of the concept of DPS. Consider a state $\rho_{\mathcal{AB}}$ residing in the composite Hilbert space $\mathcal{A}\otimes\mathcal{B}$, which exhibits separability, meaning that it can be expressed as a convex combination of pure product states:

$$\rho_{\mathcal{AB}} = \sum_i \lambda_i \left|\phi_i\right\rangle\!\left\langle\phi_i\right|_{\mathcal{A}} \otimes \left|\varphi_i\right\rangle\!\left\langle\varphi_i\right|_{\mathcal{B}}, \tag{127}$$

where the coefficients λ_i satisfy the conditions $\sum_i \lambda_i = 1$ and $\lambda_i \geqslant 0$. Consider a Hilbert space $\mathcal{A}^k \otimes \mathcal{B}^l$, and let $\tilde{\rho}$ be a state on that space. If $\tilde{\rho}$ satisfies the condition:

$$\mathrm{Tr}_{\mathcal{A}^{k-1}\mathcal{B}^{l-1}}\left[\tilde{\rho}\right] = \rho_{\mathcal{AB}}, \tag{128}$$

where the partial trace is taken over all but the first copy of each space, then $\tilde{\rho}$ is called an *extension* [97, 264] of ρ. Let $\mathcal{S}_{\mathcal{A}}$ represent the set of all permutation operators among copies of the space $\mathcal{A}$, and the same applies to B. The state extension $\tilde{\rho}$ is considered symmetric if, for every $P \in \mathcal{S}_{\mathcal{A}} \otimes \mathcal{S}_B$, the following condition holds:

$$\tilde{\rho} = P\tilde{\rho}P. \tag{129}$$

On the other hand, the state extension $\tilde{\rho}$ is classified as PPT if $\tilde{\rho}$ remains positive after applying any partial transposition on the subsystems. When $\rho_{\mathcal{AB}}$ is separable, it is guaranteed that for any values of k and l, there exists an extension $\tilde{\rho}$ that is a PPT symmetric extension of $\rho_{\mathcal{AB}}$. The fundamental principle behind the DPS hierarchy is to examine whether a PPT symmetric extension of $\rho_{\mathcal{AB}}$ exists for fixed values of k and l, and if not, this implies that $\rho_{\mathcal{AB}}$ is not separable. As the constraints of PPT symmetric extension can be expressed as SDPs, the DPS method enables optimization over a relaxation of the set of separable states on the given spaces. As the values of k and l increase, the relaxation approaches the actual set of all separable states more closely. Therefore, starting from a PPT symmetric extension state $\tilde{\rho}$, it is possible to construct a state ϱ on $\mathcal{A}\otimes\mathcal{B}$ that is, in a certain sense, *close* to being separable.

The method can be applied analogously to more than two parties, as in the following example involving three subsystems. We will now demonstrate an application of DPS, which involves a method for representing quantum states and measurements using a single SDP variable. Consider three unit vectors: $\left|\Phi^{\lambda}\right\rangle = \sum_{i\in[d_{\mathcal{A}}]}\sum_{j\in[d_{\mathcal{B}}]}\phi_{ij}\left|ij\right\rangle_{\mathcal{AB}}$, $\left|u^{\lambda}\right\rangle = \sum_{i\in[d_{\mathcal{A}}]}\mu_i^{\lambda}\left|i\right\rangle_{\mathcal{A}}$, and $\left|v^{\lambda}\right\rangle = \sum_{j\in[d_{\mathcal{B}}]}\nu_j^{\lambda}\left|j\right\rangle_{\mathcal{B}}$. Here, λ represents a global hidden variable with an arbitrary probability distribution p^{λ}. Next, we define the following operator

$$W_{\mathcal{AB}\mathcal{A}'\mathcal{B}'} \equiv \sum_{\lambda} p^{\lambda}\cdot\left[\left|\Phi^{\lambda}\right\rangle\!\left\langle\Phi^{\lambda}\right|_{\mathcal{AB}} \otimes \left|u^{\lambda}\right\rangle\!\left\langle u^{\lambda}\right|_{A'} \otimes \left|v^{\lambda}\right\rangle\!\left\langle v^{\lambda}\right|_{\mathcal{B}'}\right]. \tag{130}$$

For two subsystems S_1, S_2, both of dimension d_S, the SWAP operator is defined in the following way:

$$\mathrm{SWAP}\left(S_1,S_2\right) \equiv \sum_{i,j\in[d_S]} \left|ij\right\rangle\!\left\langle ji\right|_{S_1S_2}. \tag{131}$$

The idea that a tensor product of the density matrix ρ and measurements $\{M\}$ contain all elements necessary to express the probabilities of the form $\mathrm{Tr}(\rho M)$ was formulated in [327]. In [223], where the so-called Navascués–de la Torre–Vértesi (NTV) SDP hierarchy was introduced, it was recognized that this idea in combination with DPS hierarchy allows approximating the probabilities $\mathrm{Tr}(\rho M)$ as entries in SDP variables. NTV provided one of the

methods to approximate the set of quantum correlations with dimension constraints; other such methods are discussed in section 4.7. A similar approach was used in [341, equation (4)] to express a constraint of purity of the state, and in consequence to develop a method of providing rank constraints on the considered operators. Direct calculations show that $\mathrm{Tr}\left[W_{\mathcal{AB}\mathcal{A}'\mathcal{B}'}\left(\mathrm{SWAP}(\mathcal{A},\mathcal{A}')\otimes\mathrm{SWAP}(\mathcal{B},\mathcal{B}')\right)\right]$ is equal to:

$$\begin{aligned}\sum_{\lambda} p^{\lambda}\cdot\mathrm{Tr}\left[\tilde{W}^{\lambda}_{\mathcal{AB}\mathcal{A}'\mathcal{B}'}\right] &= \sum_{\lambda} p^{\lambda}\cdot\left[\sum_{i_1,i_2\in[d_{\mathcal{A}}]}\sum_{j_1,j_2\in[d_{\mathcal{B}}]}\phi^{\lambda}_{i_1j_1}\phi^{\lambda*}_{i_2j_2}\mu^{\lambda}_{i_2}\mu^{\lambda*}_{i_1}\nu^{\lambda}_{j_2}\nu^{\lambda*}_{j_1}\right]\\ &= \sum_{\lambda} p^{\lambda}\cdot\mathrm{Tr}\left[|\,\Phi^{\lambda}\rangle\langle\Phi^{\lambda}\,|_{\mathcal{AB}}\left(|\,u^{\lambda}\rangle\langle u^{\lambda}\,|_{\mathcal{A}}\otimes|\,v^{\lambda}\rangle\langle v^{\lambda}\,|_{\mathcal{B}}\right)\right],\end{aligned} \tag{132}$$

where $\tilde{W}^{\lambda}_{\mathcal{AB}\mathcal{A}'\mathcal{B}'}$ is defined as:

$$\sum_{\substack{i_1,i_2,i_3,\\ i_4,i_5,i_6\in[d_{\mathcal{A}}]}}\sum_{\substack{j_1,j_2,j_3,\\ j_4,j_5,j_6\in[d_{\mathcal{B}}]}}\phi^{\lambda}_{i_1j_1}\phi^{\lambda*}_{i_2j_2}\mu^{\lambda}_{i_3}\mu^{\lambda*}_{i_4}\nu^{\lambda}_{j_3}\nu^{\lambda*}_{j_4}|\,i_1j_1i_3j_3\rangle\langle i_2j_2i_4j_4|i_5j_5i_6j_6\rangle\langle i_6j_6i_5j_5\,|_{\mathcal{AB}\mathcal{A}'\mathcal{B}'}. \tag{133}$$

The resulting expression is the probability of projection of the state on some projective measurements. With similar calculations, we also obtain:

$$\mathrm{Tr}\left[W_{\mathcal{AB}\mathcal{A}'\mathcal{B}'}\left(\mathrm{SWAP}\left(\mathcal{A},\mathcal{A}'\right)\otimes\mathbb{1}_{\mathcal{B}\mathcal{B}'}\right)\right]=\sum_{\lambda}p^{\lambda}\cdot\mathrm{Tr}\left[|\,\Phi^{\lambda}\rangle\langle\Phi^{\lambda}\,|_{\mathcal{AB}}\left(|\,u^{\lambda}\rangle\langle u^{\lambda}\,|_{\mathcal{A}}\otimes\mathbb{1}_{\mathcal{B}}\right)\right]. \tag{134}$$

We see that operators created in a similar way like (130) contain entries expressing Frobenius products (3) of a state and measurements, and SWAP operators provide a tool to extract them to obtain quantum probabilities. It is easy to generalize the above formulae to cover cases involving e.g. more projective measurements and more parties. Obviously, the operator (130) is separable, and this constraint is imposed with the discussed DPS method.

The DPS method implementation is given in appendix B.4, together with an example of a Tsirelson bound calculation using NTV method.

4.3. Choi–Jamiołkowski isomorphism and quantum channels

The Choi–Jamiołkowski isomorphism introduced in 1972 by Jamiołkowski in [155] and, independently in 1975 by Choi in [70] is a fundamental concept in quantum information theory that establishes a correspondence between quantum states and quantum channels, see [161] for a detailed discussion and historical remarks. The isomorphism also called a *state-channel duality*, provides a mathematical framework to represent quantum channels as density matrices, enabling the study and manipulation of quantum processes using tools from quantum state theory. We say that a map is PSD when it is transforming PSD matrices to PSD matrices; it is *completely positive* if $\mathcal{E}\otimes\mathbb{1}$ is PSD for $\mathbb{1}$ acting over arbitrary space; a linear map is a *trace preserving* when the trace of the input matrix is equal to the trace of the output matrix. The Choi–Jamiołkowski isomorphism is defined as follows. Given a linear PSD map $\mathcal{E}:\mathbb{C}^{d\times d}\to\mathbb{C}^{d'\times d'}$, i.e. $\mathcal{E}\in\mathsf{L}\left[\mathbb{C}^{d\times d},\mathbb{C}^{d'\times d'}\right]$, that transforms input states on $\mathbb{C}^{d\times d}$ to output states on $\mathbb{C}^{d'\times d'}$, its corresponding Choi matrix $J(\mathcal{E})\in\mathbb{C}^{d'\times d'}\otimes\mathbb{C}^{d\times d}$ is the PSD matrix defined in the following way:

$$J(\mathcal{E})\equiv\sum_{i,j\in[d]}\mathcal{E}\left[|\,i\rangle\langle j\,|\right]\otimes|\,i\rangle\langle j\,|=d\cdot(\mathcal{E}\otimes\mathbb{1}_d)\left[|\,\Phi^{+}\rangle\langle\Phi^{+}\,|\right], \tag{135}$$

where $|\Phi^+\rangle \equiv \frac{1}{\sqrt{d}}\sum_{i\in[d]}|i\rangle\otimes|i\rangle$ is the maximally entangled state.

Given a PSD matrix, it is possible to reconstruct the corresponding quantum channel. This reconstruction process allows us to extract useful information about the properties and behavior of the quantum channel. Let us consider a state $\rho=\sum_{k,l\in[d]}\rho_{k,l}|k\rangle\langle l|\in\mathbb{C}^{d\times d}$. Then

$$\mathbb{1}_{d'}\otimes\rho^{\mathrm{T}}=\left(\sum_{m\in[d']}|m\rangle\langle m|\right)\otimes\left(\sum_{k,l\in[d]}\rho_{k,l}|l\rangle\langle k|\right)=\sum_{m\in[d']}\sum_{k,l\in[d]}\rho_{k,l}|m\rangle\langle m|\otimes|l\rangle\langle k|, \tag{136}$$

and we have

$$\begin{aligned}J(\mathcal{E})\cdot\left(\mathbb{1}_{d'}\otimes\rho^{\mathrm{T}}\right)&=\sum_{i,j\in[d]}\sum_{k,l\in[d]}\sum_{m\in[d']}\rho_{k,l}\cdot\left(\mathcal{E}\left[|i\rangle\langle j|\right]\otimes|i\rangle\langle j|\right)\cdot\left(|m\rangle\langle m|\otimes|l\rangle\langle k|\right)\\&=\sum_{i,j,k,l\in[d]}\sum_{m\in[d']}\rho_{k,l}\cdot\left(\mathcal{E}\left[|i\rangle\langle j|\right]\cdot|m\rangle\langle m|\right)\otimes\left(|i\rangle\langle j|\cdot|l\rangle\langle k|\right)\\&=\sum_{i,j,k\in[d]}\sum_{m\in[d']}\rho_{k,l}\cdot\left(\mathcal{E}\left[|i\rangle\langle j|\right]\cdot|m\rangle\langle m|\right)\otimes|i\rangle\langle k|\\&=\sum_{i,j,k\in[d]}\rho_{k,l}\cdot\mathcal{E}\left[|i\rangle\langle j|\right]\otimes|i\rangle\langle k|\equiv Y.\end{aligned} \tag{137}$$

Then, we take the partial trace of Y over the second subspace, removing the input space:

$$\begin{aligned}\mathrm{Tr}_2 Y&=\sum_{l\in[d]}\langle l|Y|l\rangle=\sum_{i,j,k,l\in[d]}\rho_{k,l}\cdot\mathcal{E}\left[|i\rangle\langle j|\right]\langle l|i\rangle\langle k|l\rangle\\&=\sum_{i,j\in[d]}\rho_{l,j}\mathcal{E}\left[|l\rangle\langle j|\right]=\mathcal{E}\left[\sum_{i,j\in[d]}\rho_{l,j}|l\rangle\langle j|\right]=\mathcal{E}\left[\rho\right].\end{aligned} \tag{138}$$

Thus we have the following crucial property:

$$\mathrm{Tr}_2\left[J(\mathcal{E})\cdot\left(\mathbb{1}_{d'}\otimes\rho^{\mathrm{T}}\right)\right]=\mathcal{E}\left[\rho\right]. \tag{139}$$

A direct consequence of (139) is that for a POVM $\{M^b\}_b$ on $\mathbb{C}^{d'\times d'}$, we have $\mathrm{Tr}\left[J(\mathcal{E})\cdot\left(M^b\otimes\rho^{\mathrm{T}}\right)\right]=\mathrm{Tr}\left[\mathcal{E}\left[\rho\right]M^b\right]$, which is the probability of the outcome b of the POVM applied to the output state of the channel.

It can be shown, that any linear map $\mathcal{E}$ is completely PSD if and only if its Choi matrix (135) $J(\mathcal{E})$ is PSD. Similarly, the Choi–Jamiołkowski isomorphism captures the property of trace preserving with the constraint that the Choi matrix after tracing out the output subsystem is equal to the identity operator on the input subsystem, i.e.:

$$\mathrm{Tr}_1\left[J(\mathcal{E})\right]=\mathbb{1}_d. \tag{140}$$

Trivially, for $J(\mathcal{E})$ defined in (135) and trace preserving $\mathcal{E}$ we have $\mathrm{Tr}\left[\mathcal{E}\left[|i\rangle\langle j|\right]\right]=\mathrm{Tr}[|i\rangle\langle j|]=\delta_{i,j}$ and thus $\mathrm{Tr}_1[J(\mathcal{E})]=\mathbb{1}_d$. Conversely, consider a matrix $X'\in\mathbb{C}^{d'\times d'}\otimes\mathbb{C}^{d\times d}$ satisfying

$\mathrm{Tr}_1[X] = \mathbb{1}_d$. Let $X' = \sum_{i,j\in[d]} X'_{i,j} \otimes |i\rangle\langle j|$. Then since $\mathrm{Tr}[|i\rangle\langle j|] = \delta_{i,j}$, we have $\forall_{i,j}\mathrm{Tr}[X'_{i,j}] = \delta_{i,j}$, and thus for any ρ it holds that:

$$\mathrm{Tr}\left[X' \cdot \left(\mathbb{1}_{d'} \otimes \rho^{\mathrm{T}}\right)\right] = \mathrm{Tr}_2\left[\sum_{i,j\in[d]} \mathrm{Tr}_1\left[X'_{i,j} \otimes \left(|i\rangle\langle j| \cdot \rho^{\mathrm{T}}\right)\right]\right] = \mathrm{Tr}_2\left[\sum_{i\in[d]} |i\rangle\langle i| \cdot \rho^{\mathrm{T}}\right] = \mathrm{Tr}[\rho]. \tag{141}$$

It is possible to employ the Choi–Jamiołkowski isomorphism to express other properties of quantum channels. Consider a channel $\mathcal{E}_{A'B'\leftarrow AB}$ transforming states states on $\mathbb{C}^{n_A\times n_A} \otimes \mathbb{C}^{n_B\times n_B}$ to states on $\mathbb{C}^{n_{A'}\times n_{A'}} \otimes \mathbb{C}^{n_{B'}\times n_{B'}}$, for some n_A, $n_{A'}$, n_B, and $n_{B'}$, with the Choi matrix $J(\mathcal{E}_{A'B'\leftarrow AB}) \in \mathbb{C}^{n_{A'}\times n_{A'}} \times \mathbb{C}^{n_{B'}\times n_{B'}} \otimes \mathbb{C}^{n_A\times n_A} \otimes \mathbb{C}^{n_B\times n_B}$. One usually interprets A and B as inputs and A' and B' as outputs of Alice and Bob, respectively. For instance, one can show that the marginal output state of Alice is a result of a fixed operation on the marginal input state of Alice, or, in other words, the channel is non-signaling from Bob to Alice [78] if and only if $\mathrm{Tr}_{B'}[J(\mathcal{E}_{A'B'\leftarrow AB})] = \mathrm{Tr}_{BB'}[J(\mathcal{E}_{A'B'\leftarrow AB})] \otimes \mathbb{1}_B$ [187, equation (22)]; similar relation holds for channels non-signaling from Alice to Bob. Suppose that Alice and Bob are controlling ancillary subsystems, $\tilde{A}$ and $\tilde{B}$, respectively. A channel is called PPT-preserving when it transforms a bipartite PPT state $\rho_{A\tilde{A}B\tilde{B}}$, i.e. a state satisfying $\rho^{\mathrm{T}_{B\tilde{B}}}_{A\tilde{A}B\tilde{B}} \succeq 0$, to a biparitites PPT state [265, 266]. In [266] it was shown that $\mathcal{E}_{A'B'\leftarrow AB}$ is PPT-preserving if and only if $[J(\mathcal{E}_{A'B'\leftarrow AB})]^{\mathrm{T}_{BB'}} \succeq 0$. We refer to [187] for a detailed discussion of other properties of quantum channels possible to be expressed with constraints in SDPs.

In summary, the PSD condition is essential in the Choi–Jamiołkowski isomorphism as it ensures the validity of the represented quantum channels. It provides a mathematical framework to analyze and manipulate quantum processes, allowing for the exploration of various properties and applications in quantum information theory. The discussed methods are, in particular, used to express bounds on various types of channel capacities as SDPs [96, 296, 322] or for channels discrimination [330].

4.4. SoS decomposition of polynomials

The SoS technique is a powerful tool used in SDP to represent nonnegative polynomials as sums of squares of other polynomials. Recall that in SDP, the goal is to optimize a linear objective function subject to LMI constraints. However, many optimization problems involve nonnegative polynomials, and checking the nonnegativity of a polynomial can be challenging. The SoS technique provides a way to approximate these nonnegative polynomials using sums of squares, which can be readily handled in SDP. This technique is useful in many areas of mathematics, including optimization, control theory, and signal processing. One application of the SoS decomposition is in optimization problems. In particular, it can be used to determine whether a polynomial is non-negative over a given domain. This is important in optimization because many optimization problems involve minimizing or maximizing a polynomial subject to certain constraints. By using the SoS decomposition, one can determine whether the polynomial is non-negative over the feasible region, which can help in finding the optimal solution. The method found various applications in multiple areas of science and is particularly useful for polynomial optimization, robust control, and polynomial system analysis, as it allows for tractable representation and computation of nonnegative polynomials in SDP frameworks [67, 157, 195, 243–246, 258].

The main idea behind the SoS technique is to express a nonnegative polynomial as an SoSs of lower-degree polynomials. This is achieved by introducing additional variables and using

semidefinite constraints. Specifically, the nonnegative polynomial is decomposed into a SoSs of polynomials, where each polynomial is multiplied by a semidefinite matrix. The positivity of the original polynomial is then guaranteed by the positivity of the semidefinite matrices. Let us consider a homogeneous polynomial $h(x)$, where $x \in \mathbb{R}^n$, and all terms of $h(x)$ have a degree of $2m$. We say that $h(x)$ is a SoS polynomial if and only if, for some k there exists a set $\{g_i\}_{i\in[k]}$, where each g_i is a polynomial of degree m and $h(x) = \sum_{i\in[k]} g_i(x)^2$. Obviously, any SoS polynomial is always positive. Moreover, the interesting property of the SoS polynomials is that any real non-negative polynomial can be approximated arbitrarily closely by a sequence of SoS polynomials, known as the SoS hierarchy [185]. For any polynomial $g_i(x)$, we can express it as the inner product of a vector v_i and the basis of monomials $x^{(m)}$ of degree m. Mathematically, this can be written as $g_i(x) = v_i^{\mathrm{T}} \cdot x^{(m)}$. The basis $x^{(m)}$ consists of monomials of degree m, and the dimension of this basis is given by $d \equiv \binom{n+m-1}{m}$, thus $v_i \in \mathbb{R}^d$. If SoS exist, then a set $\mathcal{S} \equiv \{v_i\}_{i=1}^k$ also exist and:

$$\begin{aligned} h(x) &= \sum_{i\in[k]} g_i(x)^2 = \sum_{i\in[k]} \left(v_i^{\mathrm{T}} \cdot x^{(m)}\right)^2 = \sum_{i\in[k]} \left(x^{(m)\mathrm{T}} \cdot v_i\right)\left(v_i^{\mathrm{T}} \cdot x^{(m)}\right) \\ &= \sum_{i\in[k]} x^{(m)\mathrm{T}} \left(v_i v_i^{\mathrm{T}}\right) x^{(m)} = x^{(m)\mathrm{T}} \left(\sum_{i\in[k]} v_i v_i^{\mathrm{T}}\right) x^{(m)} \equiv x^{(m)\mathrm{T}} M x^{(m)}. \end{aligned} \tag{142}$$

Hence, even if the set $\mathcal{S}$ is not explicitly known, it is evident that $M \equiv \sum_{i\in[k]} v_i v_i^{\mathrm{T}} \succeq 0$, $M \in \mathbb{S}^{d\times d}$. This observation implies that verifying whether a polynomial $h(x)$ is SoS is equivalent to determining the existence of a PSD matrix $M \succeq 0$ that satisfies the relation $h(x) = x^{(m)\mathrm{T}} M x^{(m)}$.

Now, let us consider a symmetric matrix $H \in \mathbb{S}^d$, not necessarily PSD, such that $h(x) = x^{(m)\mathrm{T}} H x^{(m)}$. Constructing such a matrix is straightforward, as it suffices to assign the relevant coefficients from $h(x)$ onto the diagonal elements of H. Let $\{N_i\}_{i\in[D]}$, for some D, $\forall_i N_i \in \mathbb{S}^d$, be a basis of the space of all symmetric d by d matrices satisfying the equation $x^{(m)\mathrm{T}} N_i x^{(m)} = 0$. The dimension of this space depends on m, i.e. the degree of the polynomial $h(x)$, as well as n, the number of variables. The objective now is to verify the feasibility of the so-called Gram representation of the SoS polynomial. This representation is expressed as $H + \sum_i y_i N_i \succeq 0$, where y_i are coefficients. Finally, the feasibility of the Gram representation of the SoS polynomial is examined by checking if the matrix $H + \sum_i y_i N_i$ is PSD. This assessment allows for the determination of whether the given polynomial can be represented as an SoS.

The non-commutative analog of SoS called *sum of Hermitian squares* or *non-commutative SoS* [204] was introduced in [144]. We say that a Hermitian polynomial $p(X)$ in non-commutative variables $\mathbf{X} = (X_i)_i$ is a non-commutative SoS (or, simply, SoS) when there exist polynomials $(r_j(X))_j$ such that $p(X) = \sum_j r_j^\dagger r_j$. For operators used as non-commuting variables, being SoS means that the polynomial of the operators is PSD, meaning, in particular, that its expectation value is non-negative for all quantum states. The non-commutative polynomial is *weighted SoS* (WSoS) generated by a collection of Hermitian polynomials in non-commutative variables $\mathcal{P}$ when it is of the form [85]:

$$\underbrace{\sum_j r_j^\dagger r_j}_{\text{SoS}} + \underbrace{\sum_k \sum_l s_{k,l}^\dagger p_k s_{k,l}}_{\text{weighted term}} \tag{143}$$

for $p_i \in \mathcal{P}$, and some polynomials $(r_j(X))_j$ and $(s_{k,l}(X))_{k,l}$. An algorithm for finding a sum of Hermitian squares decompositions for Hermitian polynomials in non-commuting variables

based on SDP was given in [173]. The SoS decompositions were used to provide the so-called self-testing of quantum states and measurements in [21, 202, 339], see [295] for a review and discussion.

As a direct example of the application of SoS in quantum information, we will analyze the so-called quantum moment problem. In this problem, we ask whether, for a given probability distribution, there exists a quantum state and measurements that produce such distribution (see section 1.4). A complementary problem is to decide whether a given instance of the quantum moment problem is unsatisfiable. A crucial tool for this purpose is one of the versions of the Positivstellensatz. Positivstellensatze are theorems in real algebraic geometry that provide a way to determine whether a polynomial is positive on a semi-algebraic set, or if it can be written as an SoS; see [34, chapter 4] of an introduction for the commutative variables case, and [143, 278] for the non-commutative variables case.

Suppose that the quantum setup of interest involves the measurement operators $\mathbf{X} = (X_i)_i$. We briefly show that many of the desired properties of measurements can be expressed as a requirement that certain polynomials of these operators vanish. One of the requirements is that the measurements over different subsystems commute. Indeed, the condition $[X_i, X_j] = 0$ is equivalent to statement the Hermitian polynomial $i[X_i, X_j]$ is equal zero. If for a certain set $\mathcal{M}$ the operators $\{X_i\}_{i\in\mathcal{M}}$ are constituting a projective measurement, then the normalization condition is expressed as the requirement that the Hermitian polynomial $\mathbb{1} - \sum_{i\in\mathcal{M}} X_i$ is equal zero; similarly the idempotency condition of X_i is expressed as $X_i^2 - X_i$ is equal zero. Another simple constraint possible to be expressed with vanishing Hermitian polynomials is the requirement that X has eigenvalues in $\{+1, -1\}$; this is equivalent to the requirement that $\mathbb{1} - X^2$ is equal zero. Let us define $\mathcal{P}$ to be the set of all Hermitian polynomials of one of these forms, as well as their negatives, where each of the polynomials imposes a constraint that is desired in a given scenario.

Consider an expression $G \equiv \sum_{\mathbf{a},\mathbf{x}} \alpha_{\mathbf{a}|\mathbf{x}} p_{\mathbf{a}|\mathbf{x}}[\mathbf{X}]$. We are interested in finding the Tsirelson bound [72] of such an expression. If for any quantum state $|\Phi\rangle$ and measurement operators $\mathbf{X}$ it holds $\langle\Phi|G[\mathbf{X}]|\Phi\rangle \leqslant q$ for some $q \in \mathbb{R}$, then trivially $q\cdot\mathbb{1} - G \succeq 0$. In [85, theorem 4.3] the following form of Positivstellensatz was given: if $\left[\forall_{\mathbf{X}} \left(\forall_{p\in\mathcal{P}} p[\mathbf{X}] = 0 \implies \forall_{|\Phi\rangle} \langle\Phi|(q\cdot\mathbb{1} - G)|\Phi\rangle > 0\right)\right]$, then $v\cdot\mathbb{1} - G$ is WSoS. In other words, if the Hermitian polynomial $q\cdot\mathbb{1} - G$ in non-commuting variables $\mathbf{X}$ is a PD operator (expressed as $\forall_{|\Phi\rangle}\langle\Phi|(q\cdot\mathbb{1} - G)|\Phi\rangle > 0$) under the assumption that $\mathbf{X}$ are quantum measurements (expressed as $\forall_{p\in\mathcal{P}} p[\mathbf{X}] = 0$), then it can be written in the WSoS form. Thus, this Positivstellensatz says that if there exists no quantum state and measurements attaining the value q, then $q\cdot\mathbb{1} - G$ is WSoS.

Now, the question is, how to derive the value of q using SDP. We do not have information regarding the degree of polynomials $(r_j(X))_j$ and $(s_{k,l}(X))_{k,l}$ occurring in (143). The result of [85, section 5] is a hierarchy of relaxations allowing to get a sequence q_n such that $\lim_{n\to\infty} q_n = q$, with $q_n \geqslant q$. For the level n of the relaxation, we require that r_j and $s_{k,l}$ are of degree at most n and $n-1$, respectively.

Consider the first level, $n = 1$ in the CHSH scenario [73], with $G = A_1B_1 + A_1B_2 + A_2B_1 - A_2B_2$, where E_x^a and $F^b{}_y$ are commuting projective measurement operators of Alice and Bob, respectively, see section 1.4, and $A_x \equiv E_x^0 - E_x^1$ and $B_y \equiv F^0{}_y - F^1{}_y$. The basis of polynomials of degree 1 in variables occurring in G is e.g. $x^{(1)} = [A_1; A_2; B_1; B_2]$. The SoS part of WSoS (143) is thus $\sum_j r_j^\dagger r_j = x^{(1)T} M x^{(1)}$, see (142), for some $M \succeq 0$. The only constraint expressed in $\mathcal{P}$ are those imposing that A_x and B_y have eigenvalues in $\{+1, -1\}$; let us denote the relevant constraint polynomials as $p_x^{(A)} \equiv (\mathbb{1} - A_x^2) \in \mathcal{P}$ and $p_y^{(B)} \equiv (\mathbb{1} - B_y^2) \in \mathcal{P}$, respectively. Indeed, the polynomials $p_x^{(A)}$ and $p_y^{(B)}$ vanish if and only if these eigenvalue constraints

on A_x and B_y are satisfied. Since $s_{k,l}$ are at this level all of degree $n-1=0$, they are real numbers; let us denote them as $\gamma_x^{(A)} \in \mathbb{R}$ and $\gamma_y^{(B)} \in \mathbb{R}$, respectively. The WSoS decomposition (143) is now of the form:

$$q_1 \cdot \mathbb{1} - (A_1B_1 + A_1B_2 + A_2B_1 - A_2B_2) = x^{(1)\mathrm{T}} M x^{(1)} + \sum_{x\in[2]} \gamma_x^{(A)} A_x + \sum_{y\in[2]} \gamma_y^{(B)} B_y. \tag{144}$$

To find the value q_1 we can solve the problem of minimizing q_1 subject to (144) and $M \succeq 0$; this clearly is an SDP. The result is

$$\begin{aligned} &2\sqrt{2}\cdot\mathbb{1} - (A_1B_1 + A_1B_2 + A_2B_1 - A_2B_2) \\ &\quad = \frac{1}{2\sqrt{2}}\left(r_1^\dagger r_1 + r_2^\dagger r_2\right) + \frac{1}{\sqrt{2}}\left(p_1^{(A)} + p_2^{(A)} + p_1^{(B)} + p_2^{(B)}\right), \end{aligned} \tag{145}$$

where $r_1 \equiv A_1 + A_2 - \sqrt{2}B_1$ and $r_2 \equiv A_1 - A_2 - \sqrt{2}B_2$. The Tsirelson bound is thus $2\sqrt{2}$.

4.5. Lovász theta and contextuality

The concept of zero-error capacity plays a significant role in information theory as it pertains to the flawless transmission of information through a communication channel. This notion is of utmost importance as it guarantees the reliability and precision of data transfer, which holds immense significance in diverse domains, including telecommunications, computer networking, and cryptography. The notion of a zero-error capacity of a channel represented by a graph was introduced by Shannon in 1956 in the paper *The zero error capacity of a noisy channel* [283], as defined below. The calculation of this entity, unfortunately, poses significant challenges. Lovász addressed this problem in [196] by formulating an SDP relaxation known as the Lovász θ function, or Lovász number. The introduction of this function had a profound influence not only in classical and quantum information theories [71, 76, 88] but also in related fields such as graph theory [164, 171]. Its impact transcends disciplinary boundaries, highlighting its importance and wide-ranging implications.

Consider an alphabet consisting of n letters that need to be communicated through an erroneous channel. We can represent this communication scenario using a graph G, where each vertex corresponds to a letter from the alphabet. The edges of the graph indicate the possible confusion between letters, based on the communication model being considered. It is evident that the count of one-letter messages that are guaranteed not to be confused is equivalent to the size of the largest independent set in the graph, denoted as $\alpha(G)$.

In the context of zero-error communication in the asymptotic limit, where multiple uses of the communication channel and non-trivial coding schemes are allowed, it often becomes possible to transmit a higher average number of letters per channel use. To illustrate this concept, let us consider the number of k-letter messages that can be transmitted without confusion, denoted as $\alpha(G^k)$. It is observed that $\alpha(G^k) \geqslant \alpha(G)^k$, indicating that the size of the largest independent set, $\alpha(G)$, raised to the power of k provides a lower bound on the number of distinct messages that can be encoded without the risk of confusion. For instance, if a single-letter message allows for $l = \alpha(G)$ different messages without confusion, then with k letters, we can encode at least l^k distinct messages without the risk of confusion. As an example, consider the cycle graph C_5 with five vertices, where $\alpha(C_5) = 2$ and $\alpha(C_5^2) = 5$.

The *Shannon capacity* of a graph G is a measure defined as follows. It is denoted by $\Theta(G)$ and is given by the supremum over all values of k of the expression $\alpha(G^k)^{\frac{1}{k}}$, i.e. over all possible lengths of messages encoding a chunk of information:

$$\Theta(G) = \sup_k \alpha\left(G^k\right)^{\frac{1}{k}}. \tag{146}$$

Here, $\alpha(G^k)$ represents the size of the largest independent set in the graph G^k, which is obtained by taking the strong product of G with itself k times. The Shannon capacity provides an important characterization of the graph's ability to transmit information without errors, and it is widely used in the field of information theory.

To provide the formulation of SDP relaxation of the Shannon capacity, and thus also of the independence number of a graph, introduce the concept of the *strong product* of two graphs. Let us consider a pair of graphs, G and H, and define their strong product as $G \boxtimes H$. The vertex set of $G \boxtimes H$, denoted as $V(G \boxtimes H)$, is the Cartesian product of the vertex sets of G and H, i.e. $V(G \boxtimes H) = V(G) \times V(H)$. In this construction, a vertex (x_1, y_1) in $G \boxtimes H$ is adjacent to another vertex (x_2, y_2) if and only if one of the following conditions holds:

- vertex x_1 is adjacent to vertex x_2 in G, and vertex y_1 is adjacent to vertex y_2 in H;
- vertex x_1 is equal to vertex x_2, and vertex y_1 is adjacent to vertex y_2 in H; or
- vertex x_1 is adjacent to vertex x_2 in G, and vertex y_1 is equal to vertex y_2.

This construction is known as the strong product of graphs. To extend this notion, we define $G^1 = G$, and for $k+1$, we have $G^{k+1} = G^k \boxtimes G$.

An *orthonormal representation* (OR) of G in $\mathbb{R}^d$ for some d is a set of vectors $\{|u_i\rangle\}_{i \in V(G)} \subset \mathbb{R}^d$ satisfying $\langle u_i, u_j \rangle = 0$ for all pairs of non-adjacent vertices $i, j \in V(G)$. We denote by $\mathcal{OR}(G)$ the set of all OR of G in any dimension. The *value* of the OR is

$$\min_{\substack{|\Psi\rangle \in \mathbb{R}^d, \\ ||\Psi\rangle| = 1|}} \max_{i \in V(G)} \frac{1}{|\langle \Psi, u_i \rangle|^2}. \tag{147}$$

Any Ψ for that the minimum in (147) is attained, is called a *handle* of OR. If $\{|u_i\rangle\}_{i \in V(G)}$ and $\{|v_j\rangle\}_{j \in V(H)}$ are OR of graphs G and H, then $\{|u_i\rangle \otimes |v_j\rangle\}_{i \in V(G), j \in V(H)}$ is an OR of the graph $G \boxtimes H$ [196, p 2]. If the vectors are in $\mathbb{C}^d$ instead of $\mathbb{R}^d$, then the term *orthogonal embedding* is used instead of OR. The *orthogonal rank* of G, denoted $\xi(G)$ is the smallest positive integer d such that there exists an orthogonal embedding [44, 59]. Lovász's function, denoted as $\theta(G)$, is defined as the minimum of (147) over $\mathcal{OR}(G)$.

Lovász's $\theta(G)$ plays an important role in the study of the Shannon capacity. It possesses the property that the Shannon capacity $\Theta(G)$, is bounded from above by $\theta(G)$, $\Theta(G) \leqslant \theta(G)$. To provide an SDP formulation of $\theta(G)$, let us consider a set $\mathcal{A}$ consisting of all symmetric matrices $\{A_i\}_i$ that satisfy the following conditions: for any two nodes i and j in the graph G, if $i = j$ or if i and j are not adjacent in G, then the entry A_{ij} is set to 1. The remaining entries of these matrices are left unconstrained. The value of $\theta(G)$ is equal to the minimum of the largest eigenvalue among all matrices in the set $\mathcal{A}$ [196, theorem 3]. This relaxation technique provides an effective approach to approximate the Shannon capacity of a graph and is possible to be expressed as an SDP. Indeed, the constraints defining the set $\mathcal{A}$ are linear, and the SDP is

$$\begin{aligned} &\text{minimize } \lambda \\ &\text{subject to } X \in \mathcal{A} \\ &\qquad \lambda \cdot \mathbb{1} - X \succeq 0. \end{aligned} \tag{148}$$

Alternatively, it can be shown [196, theorem 5] that

$$\theta(G) = \max_{\{|u_i\rangle\}_i \in \mathcal{OR}(\bar{G})} \max_{\substack{|\Psi\rangle \in \mathbb{R}^d, \\ ||\Psi\rangle|=1}} \sum_{i \in V(G)} |\langle \Psi, u_i \rangle|^2, \tag{149}$$

where $\bar{G}$ is the complementary graph of G.

In [129] Grötschel, Lovász and Schrijver introduced the weighted version of θ function, intending to derive the maximum weight independent sets in perfect graphs. Let $\mathbf{w} = (w_i)_{i \in V(G)}$ be weights of nodes in a weighted graph G. The generalization of the θ function (147) to weighted graphs is defined as [131, p 4]:

$$\theta(G, \mathbf{w}) \equiv \min_{\{|u_i\rangle\}_i \in \mathcal{OR}(G)} \min_{\substack{|\Psi\rangle \in \mathbb{R}^d, \\ ||\Psi\rangle|=1}} \max_{i \in V(G)} \frac{w_i}{|\langle \Psi, u_i \rangle|^2}. \tag{150}$$

A direct generalization of (149) is [131, theorem 2.3]:

$$\theta(G, \mathbf{w}) = \max_{\{|u_i\rangle\}_i \in \mathcal{OR}(\bar{G})} \max_{\substack{|\Psi\rangle \in \mathbb{R}^d, \\ ||\Psi\rangle|=1}} \sum_{i \in V(G)} w_i |\langle \Psi, u_i \rangle|^2. \tag{151}$$

The value of (151) can also be expressed as SDP [129].

The fundamental role of the Lovász θ for quantum correlations and contextuality was recognized in [57] and developed in [58]. The results state that the maximal value of correlations allowed by quantum mechanics is given by the Lovász number of the so-called *exclusivity graph*. Here we briefly sketch the results, and we refer to [3] for a discussion and further advancements.

The exclusivity graph of a multi-partite correlation experiment represents the possible events with vertices and the exclusion of pairs of events by edges. We say that the events e_1 and e_2 are exclusive if and only if there exist two jointly measurable observables (tests) μ_i and μ_j that distinguish between them. The experiments with space-like separated tests are Bell inequalities [25, 73] as discussed in section 1.4. More general scenarios are non-contextual inequalities, which distinguish between theories in which outcomes are predefined from contextual theories, including quantum mechanics [120, 176, 287]. For instance, in the CHSH Bell experiment [73] there are four tests, each providing binary results, *viz.* two measurements performed by Alice, and two measurements performed by Bob.

Consider a positive linear combination of events, or positive non-contextual game expression, of the form $\sum_i w_i P(e_i)$, with all $w_i > 0$. The CHSH Bell inequality [73] can be rewritten in this form as

$$\sum_{a \in \{0,1\}} P(a,a|0,0) + P(a,a|0,1) + P(a,a|1,0) + P(a,\neg a|1,1) \leqslant 3. \tag{152}$$

The exclusivity graph of the positive non-contextual game expression is the induced subgraph of the exclusivity graph of the experiment, see figure 1. In [58] it was shown that from (151) it follows that the attainable upper bound on the positive non-contextual game expression in quantum mechanics is exactly $\theta(G, \mathbf{w})$, where G is the exclusivity graph of the positive non-contextual game expression [273].

The work [82] reveals another association between the Lovász number and fundamental quantum phenomena, *viz.* the uncertainty relations [139, 167, 270], which characterize the

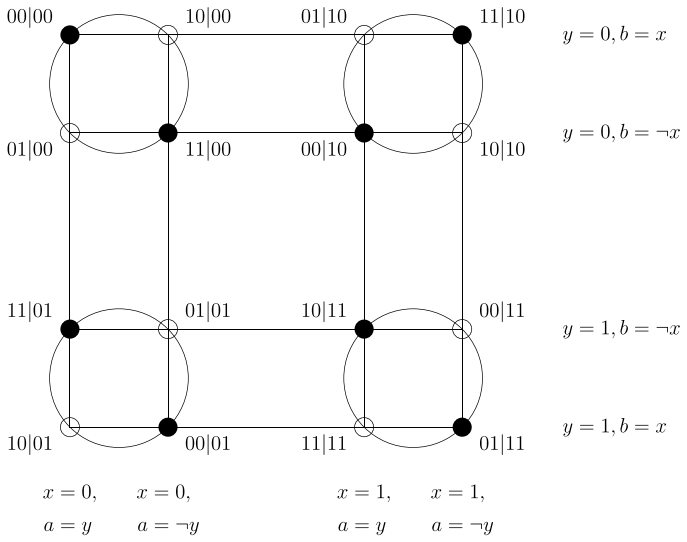

Figure 1. Exclusivity graph for the two-partite Bell scenario with two settings and two outcomes for each of the parties. The events are labeled as $ab|xy$, with the settings of Alice and Bob denoted as x and y, and their outcomes as a and b. Sets of pairwise exclusive events are those lying on the same line or in the same circle. The exclusivity graph for the positive non-contextual game expression (152) is the induced subgraph containing only the black nodes.

limitations in precisely predicting outcomes of simultaneous measurements in quantum mechanics. The findings have practical implications and can be applied to formulate entropic uncertainty relations, separability criteria, and entanglement witnesses.

The mentioned above orthogonal rank $\xi(G)$ has a direct relation to the one-round quantum communication complexity of calculation of a function $f(x,y)$, which is equal to $\lceil \log_2(\xi(G)) \rceil$, with the promise that the joint input $(x,y) \in \mathcal{D} \subset V(G) \times \mathcal{Y}$. The vertices i and j in G are connected if and only if $\exists_{y \in \mathcal{Y}} (i,y) \in \mathcal{D} \wedge (j,y) \in \mathcal{D}$ [83, theorem 8.5.2]. As discussed in [225], the so-called *almost quantum* or $\mathfrak{Q}_{1+AB}$ SDP relaxation of the set of quantum behaviors, see section 4.6, is also closely related to the Lovász θ.

4.6. Correlation matrices, moment matrices, and optimization over non-commuting variables

Consider a sequence of real-valued random variables $\mathcal{S} = (x_1, \ldots x_n)$. The covariance matrix of $\mathcal{S}$ is defined as the matrix whose entries are given by relevant covariations of pairs of variables, $\mathrm{Cov}[\mathcal{S}] = [\mathrm{cov}[x_i, x_j]]_{x_i, x_j \in \mathcal{S}} \in \mathbb{R}^{\mathcal{S} \times \mathcal{S}}$. The rows and columns, in this case, could be indexed with the numbers of the variables, and thus belong to $[n]$ and then it would be $\mathrm{Cov}[\mathcal{S}] \in \mathbb{R}^{n \times n}$. Instead, we take a more generic approach and index the rows and columns with the variables themselves. Such matrices, indexed by labels or other expressions are called *moment matrices*. Now, consider a vector of constant coefficients $v = (v_1, \ldots v_n) \in \mathbb{R}^{\mathcal{S} \times 1}$. We have

$$v^{\mathrm{T}} \mathrm{Cov}\,[\mathcal{S}]\, v = \sum_{i,j \in [n]} v_i \cdot \mathrm{cov}\,[x_i, x_j] \cdot v_j = \mathrm{cov} \left[\sum_{i \in [n]} x_i, \sum_{j \in [n]} x_j \right] = \mathrm{var} \left[\sum_{i \in [n]} x_i \right] \geqslant 0, \tag{153}$$

and thus the covariance matrix is PSD. The correlation matrix $\mathrm{corr}(\mathcal{S})$ is the matrix whose entries are given by relevant correlation of pairs of variables, i.e. $\mathrm{Corr}[\mathcal{S}] = [\mathrm{corr}[x_i, x_j]]_{x_i \in \mathcal{S}, x_j \in \mathcal{S}} \in \mathbb{R}^{\mathcal{S}\times\mathcal{S}}$. Equivalently, the correlation matrix is equal to the covariance matrix of all variables rescaled to have variance 1, $\mathrm{Corr}[\mathcal{S}] = \mathrm{Cov}[\bar{\mathcal{S}}]$, for $\bar{\mathcal{S}} = (x_1/\boldsymbol{\sigma}[x_1], \ldots x_n/\boldsymbol{\sigma}[x_n])$. In consequence, all diagonal values of the correlation matrix are 1 and the matrix is also PSD. If variables are linearly independent, then it is PD. A sample numerical calculation with a correlation matrix is given in appendix B.2. The concept of moment matrices is used in the methods discussed further in this section.

The methods discussed in this work using moment matrices find a major application in optimizing linear functions of probabilities in various quantum scenarios, such as entangled parties with separated sharing or communication of a quantum system with a given dimension. These methods have gained significant popularity due to their effectiveness and applicability in quantum optimization tasks. It is worth noting that earlier studies in 2004 focused on optimizing success probabilities without utilizing moment matrices [94]. Instead, they relied on the formalism of Lagrangian in close connection with the concept of convex optimization. Particularly, researchers explored the problem of determining optimal success probabilities for static linear optics quantum gates, and intriguingly, it was found to be related to convex optimization theory. Through this connection, they successfully derived upper bounds for the success probability of networks implementing single-mode gates. Moreover, the concept of Lagrange duality played a crucial role in providing rigorous proofs for these derived bounds.

4.6.1. The NPA hierarchy. Now, we provide a brief introduction to the optimization over noncommuting variables, with a concentration on the so-called NPA method. The NPA method is based on SDP, as presented in the paper *Bounding the set of quantum correlations* by Navascués *et al* (2007) [226, 227]. Moment matrices are a basic element of the technique. The concept was inspired by the seminal work by Lasserre [184].

The problem at hand is to find a way to characterize, at least approximately, the class of all quantum behaviors $\mathfrak{Q}$ without resorting to the formalism of quantum theory. To this end, the NPA method introduces a hierarchy of SDP problems $\{\mathfrak{Q}_k\}_{k=1}^{\infty}$. Each level of the hierarchy corresponds to a specific SDP problem, where higher levels yield more accurate solutions. In other words, as we increase the level k, the set $\mathfrak{Q}_{k+1}$ becomes a subset of $\mathfrak{Q}_k$, $\mathfrak{Q}_{k+1} \subset \mathfrak{Q}_k$, providing a progressively better approximation of $\mathfrak{Q}$. However, as the level increases, the SDPs become more complex and computationally demanding. It is important to note that the hierarchy of SDP problems converges to the quantum set $\mathfrak{Q}$, meaning that the intersection of all sets in the hierarchy is equal to $\mathfrak{Q}$, *viz.* $\cap_{k=1}^{\infty}\mathfrak{Q}_k = \mathfrak{Q}$. By considering all levels of the hierarchy, we can accurately capture the entire quantum set $\mathfrak{Q}$ without explicit involvement of the formalism of Hilbert spaces, and when we restrict considerations to a particular level, then we can effectively approximate the optimization over the set of all quantum behaviors $\mathfrak{Q}$ using SDP.

A sequence of operators, which is formed by concatenating projective measurement operators, plays a crucial role in the context of quantum systems. Consider an illustrative example of such a sequence, denoted as $E_2^1 E_2^3 F^2{}_1 E_1^1$, consisting of four operators. The operators associated with Alice, denoted as E_x^a, commute with the operators corresponding to Bob, denoted as $F^b{}_y$. This allows us to rearrange the sequence without altering the original action of the operators. Thus, we can rewrite the sequence as $E_2^1 E_2^3 E_1^1 F^2{}_1$ by interchanging the operators while respecting the commutation relationship between Alice's and Bob's measurements. This reordering is made possible by the commutativity property exhibited by the operators belonging to Alice's and Bob's measurements.

In a sequence of operators, we can exploit the orthogonality property $E_x^a E_x^{a'} = 0$ and $F^b{}_y F^{b'}_y = 0$ for $a \neq a'$ and $b \neq b'$. Applying this property and utilizing the commutation property, we can rearrange operators within the sequence. For instance, let us consider the expression $E_1^2 F^3{}_3 E_1^1$. By utilizing the commutation property, we can rearrange the operators as $E_1^2 E_1^1 F^3{}_3$, which equals zero due to the orthogonality between E_1^2 and E_1^1. Additionally, it is worth noting that since E_x^a and $F^b{}_y$ are projectors, we have the property $(E_x^a)^k = E_x^a$ for any $k \geqslant 1$, and the same holds for $F^b{}_y$. This property further aids in simplifying the expressions involving repeated application of the projectors. To characterize the length of a sequence of operators, we define it as the minimum number of projectors required to represent the sequence. In this context, we consider the identity operator $\mathbb{1}$ as the *null sequence*, denoting no projectors, and its length is defined to be zero. This notion of length provides a measure of the complexity or number of steps involved in a sequence of operators.

We will now be considering sets of sequences of operators from the reduced set of operators discussed in section 1.4. Using the NPA method, we can construct a hierarchy of relaxations by choosing different sets of sequences. Specifically, we define a set $\mathcal{S}_k$ to be the set of all sequences of operators $\{E_x^{\tilde{a}}, F_y^{\tilde{b}}\}_{\tilde{a},\tilde{b},x,y}$ of the length at most k (including the sequence of length 0, i.e. $\mathbb{1}$), with the indices $\tilde{a}$ and $\tilde{b}$ covering all values excluding the last one. We also define the so-called *intermediate sets of sequences*, where only specific sequences are included, for instance

$$\mathcal{S}_{1+AB} = \mathcal{S}_1 \cup \left\{E_x^a F^b{}_y\right\}_{\tilde{a},\tilde{b},x,y} = \left\{\mathbb{1}, E_x^{\tilde{a}}, F_y^{\tilde{b}}, E_x^{\tilde{a}} F_y^{\tilde{b}}\right\}_{\tilde{a},\tilde{b},x,y}. \tag{154}$$

The hierarchy of level $\mathfrak{Q}_2$ means that the set $\mathcal{S}$ consists of all sequences of measurement operators of length 2, whereas in level $\mathfrak{Q}_{1+AB}$, $\mathcal{S}$ is a set of all sequences of length 1 and sequences with one operator of Alice and one of Bob. $\mathfrak{Q}_{1+AB}$ revealed to be so efficient that it is called an *almost quantum* set of behaviors [225].

The key idea of the NPA method can be summarized as follows. Consider a behavior $\{P(a,b|x,y)\}$ and suppose that it is quantum. This means that there exists a specific realization involving a quantum state $|\psi\rangle$ and projective measurements $\{E_x^a, F^b{}_y\}$ such that, for all settings x and y, and outcomes $\tilde{a}$ and $\tilde{b}$, the relation (8) holds, and expresses the probability of obtaining outcomes a and b when measurements x and y are performed on the quantum state $|\psi\rangle$. In the context of the NPA method, the notion of moment matrices is used as follows. For any operators O_i and O_j belonging to the set $\mathcal{S}$ of size n, we define the element of the moment matrix as:

$$\Gamma_{O_i,O_j} \equiv \langle\psi|O_i^\dagger O_j|\psi\rangle. \tag{155}$$

This equation establishes a connection between certain elements of the moment matrix and joint probability distributions. Specifically, we have

$$\Gamma_{E_x^{\tilde{a}},F_y^{\tilde{b}}} = P\left(\tilde{a},\tilde{b}|x,y\right), \tag{156}$$

which demonstrates that the elements of the moment matrix correspond to the probabilities of obtaining outcomes $\tilde{a}$ and $\tilde{b}$ for the measurements x and y. Additionally, we have the element $\Gamma_{\mathbb{1},\mathbb{1}} = 1$, which represents the identity operator, indicating that its contribution to the moment matrix is unity, and $|\psi\rangle$ is normalized. This definition results in an $n \times n$ moment matrix, where the rows and columns are indexed by the elements of the set $\mathcal{S}$. Hence, the moment matrix serves as a representation of the moments associated with the considered behavior. For instance, the sequence of operators with a length of

at most 1 in the case of the reduced set (9) is represented by $\mathcal{S}_1 = \{\mathbb{1}, E_1^0, E_2^0, F_1^0, F_2^0\} \equiv \{O_1, O_2, O_3, O_4, O_5\}$. From (155) and (156) we have $P_A(0|0) = \Gamma_{\mathbb{1},E_0^0} \equiv y_1$, $P_A(0|1) = \Gamma_{\mathbb{1},E_1^0} \equiv y_2$, $P_B(0|0) = \Gamma_{\mathbb{1},F^0{}_0} \equiv y_4$, $P_B(0|1) = \Gamma_{\mathbb{1},F_1^0} \equiv y_7$, $P(0,0|0,0) = \Gamma_{E_0^0,F^0{}_0} \equiv y_5$, $P(0,0|1,0) = \Gamma_{E_1^0,F^0{}_0} \equiv y_6$, $P(0,0|0,1) = \Gamma_{E_0^0,F_1^0} \equiv y_8$, and $P(0,0|1,1) = \Gamma_{E_1^0,F_1^0} \equiv y_9$. This leads to the following formula for the Γ matrix, which is in this case a 5×5 real matrix:

$$\begin{bmatrix} 1 & y_1 & y_2 & y_4 & y_7 \\ y_1 & y_1 & y_3 & y_5 & y_8 \\ y_2 & y_3 & y_2 & y_6 & y_9 \\ y_4 & y_5 & y_6 & y_4 & y_{10} \\ y_7 & y_8 & y_9 & y_{10} & y_7 \end{bmatrix}. \tag{157}$$

The probabilities not occurring directly in the matrix can be derived from the non-signaling constraints (7). For instance, from the matrix (157) we have $P(0,1|0,1) = P_A(0|0) - P(0,0|0,1) = y_1 - y_8$.

We can observe that the elements of the moment matrix Γ are subject to the following linear constraints. For any indices i, j, k, and l, the equality $\Gamma_{O_i,O_j} = \Gamma_{O_k,O_l}$ holds whenever the corresponding operators satisfy $O_i^\dagger O_j = O_k^\dagger O_l$, i.e. $O_i^\dagger O_j = O_k^\dagger O_l \implies \Gamma_{O_i,O_j} = \Gamma_{O_k,O_l}$. This constraint ensures that the inner products of identical operator sequences yield equal moments. Similarly, if $O_i^\dagger O_j$ results in the zero operator, then it implies that $\Gamma_{O_i,O_j} = 0$. This condition ensures that the moments associated with operator sequences resulting in the null operator are also zero. These linear constraints provide necessary relations between the elements of the moment matrix, allowing us to impose consistency and capture important properties of the behavior. Positive semi-definiteness of Γ is a direct consequence of (155). Indeed, let $v \in \mathbb{C}^n$. For $V = \sum_j v_j O_j$ we have

$$v^\dagger \Gamma v = \sum_{i,j} v_i^* \Gamma_{O_i,O_j} v_j = \sum_{i,j} v_i^* \langle \psi | O_i^\dagger O_j | \psi \rangle v_j = \langle \psi | V^\dagger V | \psi \rangle = |V|\psi\rangle|^2 \geqslant 0, \tag{158}$$

and thus $\Gamma \succeq 0$. To summarize, we observe the following:

- The sets $\mathcal{S}_1 \subset \mathcal{S}_2 \cdots \subset \mathcal{S}_\infty$ form an increasing sequence, where each set contains longer sequences of measurement operators.
- The hierarchy levels $\mathfrak{Q}_1 \supset \mathfrak{Q}_2 \cdots \supset \mathfrak{Q}$ form a decreasing sequence, indicating a refinement of the approximation to the quantum set.
- The quantum set $\mathfrak{Q}$ is equal to the intersection of all levels Q_k, i.e. $\mathfrak{Q} = \bigcap_{k=1}^\infty Q_k$.

The final equality, which pertains to convergence to the quantum set, has been proven in [227]. The sizes of the sets $\mathcal{S}_k$ and, consequently, the sizes of the Γ matrices, grow exponentially, specifically as $O\left((|A| \cdot |X| + |B| \cdot |Y|)^k\right)$. In practice, sets beyond $\mathfrak{Q}_3$ are rarely utilized, and due to finite precision arithmetics large matrices may cause numerical issues. The set $\mathfrak{Q}_{1+AB}$ is generally sufficient for most purposes and is often referred to as the *almost quantum set*. The set $\mathfrak{Q}_1$ is often called the macroscopic locality set [229]. A comparison of the primal and dual approaches for imposing the aforementioned operator constraints is presented in table 1. A comprehensive discussion on this topic can be found in section 2.3.1 of [211]. It is worth noting that when expressing the constraint of NPA optimizations, the parameter m representing the size of the canonical SDP form in (80) and (81) is significantly smaller if we opt for the latter approach. It can be seen that the above example (157) is written in the form $C - \sum_i y_i A_i$

Table 1. Comparison of sizes of SDP formulations of the levels of the NPA in a scenario with two parties, each with two binary measurements, when the constraints are expressed in terms of primal or dual canonical SDP forms.

Hierarchy level	n	m-dual	Average density (dual)	m-primal
$\mathfrak{Q}_2$	13	31	0.183	137
$\mathfrak{Q}_3$	25	61	0.098	563
$\mathfrak{Q}_4$	41	101	0.060	1579
$\mathfrak{Q}_5$	61	151	0.041	3569
$\mathfrak{Q}_6$	85	211	0.029	7013
$\mathfrak{Q}_7$	113	281	0.022	12 487
$\mathfrak{Q}_8$	145	361	0.017	20 663
$\mathfrak{Q}_9$	181	451	0.014	32 309
$\mathfrak{Q}_{10}$	221	551	0.011	48 289
$\mathfrak{Q}_{11}$	265	661	0.009	69 563
$\mathfrak{Q}_{12}$	313	781	0.008	97 187
$\mathfrak{Q}_{13}$	365	911	0.007	132 313
$\mathfrak{Q}_{14}$	421	1051	0.006	176 189
$\mathfrak{Q}_{15}$	481	1201	0.005	230 159

of the canonical dual formulation (81), and can be easily expressed in other forms discussed in section 3.2.

There are multiple variants and extensions of NPA. In [254] the inventors of NPA showed how to apply their techniques to the general problem of polynomial optimization over non-commuting variables. In [163] it was shown how to use NPA to analyze the so-called extended non-local games. These games involve three parties, *viz.* Alice, Bob, and a referee. Initially, Alice and Bob share a tripartite quantum state with the referee. In these games, the conditions for Alice and Bob to win may depend not only on their answers to randomly selected questions but also on the outcomes of measurements performed by the referee on its portion of the shared quantum state. In a recent work [257] a method for analysis of classical and quantum correlations in networks with causally independent parties was introduced, providing a way to use NPA in complex quantum networks. Another work analyzing generalizations of NPA for characterization of the quantum network correlations, together with convergence results was given in [269], see [301] for an overview.

The almost quantum correlations are applied and discussed in [149, 277]. Their great theoretical importance stems from their role in axiomatics of quantum mechanics [225]. In [225] it was proposed to consider the set of behaviors allowed in the almost quantum relaxation level of NPA as a physical theory and hypothesized that the actual physics of the real world is not the quantum theory but the newly proposed *almost quantum theory*. It was shown that non-trivial communication complexity, no advantage for non-local computation, and local orthogonality are weaker in determining which behaviors are physical than the almost quantum theory. It was also shown that the almost quantum set is closed under various classical operations, including post-selection, composition, grouping of parties, and so-called wiring operation [11]. In [191] a quantitative comparison of several sets of super-quantum behaviors is given.

In [277, appendix A] the NPA method has been modified to express a relaxation of the set of quantum assemblages [62], i.e. sets of unnormalized states toward which a multipartite state can be steered to [331]. In this method, the Γ matrix' entries are not numbers but matrices themselves allowing for introduction in particular the so-called *almost quantum assemblages*, see [276] for a physical definition and discussion for its applications. A hierarchy for analysis

of quantum steering was also given in [177]. The proposed method enables the derivation of steering witnesses for arbitrary families of quantum states. A framework for the analytical derivation of non-linear steering criteria was also presented.

The work [221] introduces the so-called Moroder's hierarchy, where in addition to the NPA constraints, more restrictions of a certain form regarding entanglement can be imposed on the quantum state. From Moroder's hierarchy a further variant, allowing to imposing of constraints on Bob's measuring devices was given in [260].

A prominent application of NPA is in device-independent quantum cryptography, in particular in quantum randomness certification. The initial methods used a single parameter, the Bell inequality violation, as the certificate for this task [101, 252, 253]. In [22, 239] it was shown how to modify NPA so that the full experimental statistics can be imposed as SDP constraints with the method called *more randomness from the same data* or the Nieto–Silleras hierarchy. What is more, the method allows to use of the dual optimization task to derive a new Bell functional suited to provide the most randomness from the particular experimental realization. To this end, the method described in section 3.4 is used, where the variable β provides the coefficients of the Bell functional, and the equations $v_j = q_j \cdot x$ express the behavior constraints.

An alternative approach to optimization over non-commutative polynomials is given in [55]. In that work, the problem of minimization of a trace of a given polynomial function in non-commuting variables using SDP is considered. Next, in [174] a method for constrained trace and eigenvalue optimization of noncommutative polynomials was introduced. The results were used in [298] to characterize the classical and quantum correlations that arise in prepare-and-measure experiments when communication is informationally restricted.

4.6.2. Optimization of von Neumann entropy. In particular, the NPA method can be used to calculate the lower bound of the von Neumann conditional entropy given any kind of knowledge (classical or quantum) that an eavesdropper may possess if is subject to the laws of quantum mechanics. The method uses the SDP representations of the logarithm function discussed in section 4.1 together with the Gauss–Radau quadrature rule for the lower bound. Let w_i and t_i be the nodes and weights defined by this quadrature. Specifically, this method can be employed to compute a lower bound on the von Neumann conditional entropy under the presence of an eavesdropper, considering both classical and quantum knowledge, while adhering to the principles of quantum mechanics. Let $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{E}$ denote the Hilbert spaces corresponding to Alice's, Bob's, and the eavesdropper's devices, respectively. The quadrature rule provides a set of nodes, $\{t_i\}_i$, and weights, $\{w_i\}_i$, that are utilized in the computation. The lower bound formula for the settings selection x^* and y^* is given as [49]:

$$\sum_i c_i \left(\sum_{a,b=0,1} \inf_{\substack{Z_{a,b}\in B(Q_{\mathcal{E}}),\\ \mathrm{cond}(P)}} \left(1 + \phi\left[E^a_{x^*}, F^b{}_{y^*}, Z_{a,b}, t_i\right]\right) \right), \tag{159}$$

where $\phi[E^a_{x^*}, F^b{}_{y^*}, Z_{a,b}, t_i]$ is defined as

$$\mathrm{Tr}\left[\rho_{\mathcal{ABE}} \left(E^a_{x^*} \otimes F^b{}_{y^*} \otimes \left(Z_{a,b} + Z^\dagger_{a,b} + (1-t_i) Z_{a,b} Z^\dagger_{a,b}\right) + t_i \left(\mathbb{1}_{\mathcal{AB}} \otimes Z_{a,b} Z^\dagger_{a,b}\right)\right)\right]. \tag{160}$$

The expression $\mathrm{cond}(P)$ means that the behavior $\{P(a,b|x,y)\}$ satisfies a set of linear constraints defined by the protocol, and c_i are coefficients calculated from Gauss–Radau quadrature as $c_i \equiv w_i/(t_i \log(2))$. The i index in the sum (159) assumes values that index nodes in quadrature, skipping the last one. An example of the implementation of a method is given in [48] and described in detail in [47].

4.6.3. Self-testing with SWAP method. The phenomenon of self-testing is characterized by the ability to assess both states and measurements of certain quantum devices in a black-box setting, relying solely on observed statistics without the need for prior device calibration. However, before the work [340] the existing examples of self-testing are limited in their applicability, as they only provide meaningful assessments for devices that closely resemble the ideal case. In [340] these limitations were overcome by adopting a novel approach to self-testing, utilizing an SDP hierarchy for the characterization of quantum correlations. This approach allows for a more comprehensive and robust assessment of quantum devices, enabling meaningful evaluations even in scenarios where the devices deviate from the ideal case.

We illustrate the method with a bi-partite Bell scenario. Suppose we have gathered an experimental description of the behavior of a device $\{P(a,b|x,y)\}$, and that we expect that these statistics should have been obtained using a particular quantum state $|\bar{\psi}\rangle_{\mathcal{AB}}$ and projective measurements $\{\bar{E}^a_x, \bar{F}^b_y\}$. We would like to have a quantitative way of estimating, how close is the actual state that was prepared in the laboratory, considered as a black box described only by the statistics, to the theoretical one $|\bar{\psi}\rangle_{\mathcal{AB}}$. The work [340] proposed a method to perform such self-testing where the content of the black box is hypothesized to be swapped with a trusted system in a thought experiment; the method itself is called SWAP. Suppose that it is possible to formulate four linear functions $\mathcal{F}_{E,\sigma_x}$, $\mathcal{F}_{E,\sigma_z}$, $\mathcal{F}_{F,\sigma_x}$, and $\mathcal{F}_{F,\sigma_z}$, such that $\mathcal{F}_{E,\sigma_x}[\{\bar{E}^a_x\}] = \sigma_x$, $\mathcal{F}_{E,\sigma_z}[\{\bar{E}^a_x\}] = \sigma_z$, $\mathcal{F}_{F,\sigma_x}[\{\bar{F}^b_y\}] = \sigma_x$, and $\mathcal{F}_{F,\sigma_z}[\{\bar{F}^b_y\}] = \sigma_z$. Recall the SWAP operator given in (131). For $d_S = 2$ it takes the form

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = U[\sigma_x] \cdot V[\sigma_z] \cdot U[\sigma_x], \tag{161}$$

where

$$U[X] \equiv \mathbb{1}_2 \otimes |0\rangle\langle 0| + X \otimes |1\rangle\langle 1|, \text{and} \tag{162a}$$

$$V[X] \equiv \frac{1}{2}\left((\mathbb{1}_2 + X) \otimes \mathbb{1}_2 + (\mathbb{1}_2 - X) \otimes \sigma_x\right). \tag{162b}$$

The black box is possibly using some other quantum state $|\psi\rangle_{\mathcal{AB}}$ and measurements $\{E^a_x, F^b{}_y\}$ to implement $P(a,b|x,y)$, but if the considered scenario possesses the self-testing property, then one can expect that the state and measurements will be not very far from the theoretical ones $|\bar{\psi}\rangle$ and $\{\bar{E}^a_x, \bar{F}^b_y\}$. Now, let us perform a thought experiment with hypothesized swapping of the black-box state $|\psi\rangle_{\mathcal{AB}}$ with trusted ancillary states $|\bar{\phi}_1\rangle_{\mathcal{A}'}$ and $|\bar{\phi}_2\rangle_{\mathcal{B}'}$ using the black-box measurements $\{E^a_x, F^b{}_y\}$ together with trusted operations on the ancillas, where we denote by $\mathcal{A}$ and $\mathcal{B}$ their black-box subsystems, and by $\mathcal{A}'$ and $\mathcal{B}'$ the subsystems of their trusted ancillas. Let us consider the ancillas to be qubits, so that we can apply (161) to each party, Alice and Bob. Since the operations $\{E^a_x\}$ on the black-box subsystem of Alice are expected to approximate to some extent $\{\bar{E}^a_x\}$, we may expect that $\mathcal{F}_{E,\sigma_x}[\{E^a_x\}] \approx \sigma_x$ and $\mathcal{F}_{E,\sigma_z}[\{E^a_x\}] \approx \sigma_z$; and similarly for Bob, *viz.* $\mathcal{F}_{F,\sigma_x}[\{F^b{}_y\}] \approx \sigma_x$ and $\mathcal{F}_{F,\sigma_z}[\{F^b{}_y\}] \approx \sigma_z$. Thus for:

$$\mathcal{S}_{\mathcal{AA}'} \equiv U_{\mathcal{AA}'}\left[\mathcal{F}_{E,\sigma_x}\left[\{E^a_x\}\right]\right] \cdot V_{\mathcal{AA}'}\left[\mathcal{F}_{E,\sigma_z}\left[\{E^a_x\}\right]\right] \cdot U_{\mathcal{AA}'}\left[\mathcal{F}_{E,\sigma_x}\left[\{E^a_x\}\right]\right], \tag{163a}$$

$$\mathcal{S}_{\mathcal{BB}'} \equiv U_{\mathcal{BB}'}\left[\mathcal{F}_{F,\sigma_x}\left[\{F^b{}_y\}\right]\right] \cdot V_{\mathcal{BB}'}\left[\mathcal{F}_{F,\sigma_z}\left[\{F^b{}_y\}\right]\right] \cdot U_{\mathcal{BB}'}\left[\mathcal{F}_{F,\sigma_x}\left[\{F^b{}_y\}\right]\right], \tag{163b}$$

we have $\mathrm{SWAP}(\mathcal{A},\mathcal{A}') \approx \mathcal{S}_{\mathcal{A}\mathcal{A}'}$ and $\mathrm{SWAP}(\mathcal{B},\mathcal{B}') \approx \mathcal{S}_{\mathcal{B}\mathcal{B}'}$. Define $\mathcal{S}_{\mathcal{A}\mathcal{B}\mathcal{A}'\mathcal{B}'} \equiv \mathcal{S}_{\mathcal{A}\mathcal{A}'} \otimes \mathcal{S}_{\mathcal{B}\mathcal{B}'}$ (with proper ordering of the subsystems), and consider the hypothetical state:

$$\rho_{\mathrm{SWAP}} \equiv \mathrm{Tr}_{\mathcal{A}\mathcal{B}}\left[\mathcal{S}_{\mathcal{A}\mathcal{B}\mathcal{A}'\mathcal{B}'} \cdot \left(|\psi\rangle\langle\psi|_{\mathcal{A}\mathcal{B}} \otimes |\bar{\phi}_1\rangle\langle\bar{\phi}_1|_{\mathcal{A}'} \otimes |\bar{\phi}_2\rangle\langle\bar{\phi}_2|_{\mathcal{B}'}\right) \cdot \mathcal{S}^{\dagger}_{\mathcal{A}\mathcal{B}\mathcal{A}'\mathcal{B}'}\right]. \tag{164}$$

Knowing the explicit form of the trusted ancillas $|\bar{\phi}_1\rangle_{\mathcal{A}'}$ and $|\bar{\phi}_2\rangle_{\mathcal{B}'}$ and applying algebraic calculations with (163) on $|\psi\rangle_{\mathcal{A}\mathcal{B}}$ we can derive that the formula for ρ_{SWAP} depending on the elements possible to be retrieved from the NPA moment matrix, see section 4.6. This way e.g. the fidelity of the state or other linear functions of its element can be optimized with the same technique as NPA, as proposed in the seminal paper [340], or NV (see section 4.7.2) as shown in [299].

Without loss of generality, we can assume that the functions $\mathcal{F}_{\cdot,\cdot}$ are acting on the reduced set of operators $\{\mathbb{1}, E^{\tilde{a}}_x, F^{\tilde{b}}_y\}_{\tilde{a},\tilde{b},x,y}$. To illustrate the above concept, we hypothesize that for a particular behavior certain linear combinations of the operators of Alice and Bob express their operators σ_x and σ_z, i.e.:

$$\begin{aligned}
\bar{A}_0 &\equiv \mathcal{F}_{E,\sigma_z}\left[\left\{\bar{E}^{\tilde{a}}_x\right\}\right] = \sigma_z,\\
\bar{A}_1 &\equiv \mathcal{F}_{E,\sigma_x}\left[\left\{\bar{E}^{\tilde{a}}_x\right\}\right] = \sigma_x,\\
\bar{B}_0 &\equiv \mathcal{F}_{F,\sigma_z}\left[\left\{\bar{F}^{\tilde{b}}_y\right\}\right] = \sigma_z,\\
\bar{B}_1 &\equiv \mathcal{F}_{F,\sigma_x}\left[\left\{\bar{F}^{\tilde{b}}_y\right\}\right] = \sigma_x.
\end{aligned} \tag{165}$$

Using the formulae (162) and (163) we get the following expression for Alice:

$$\begin{aligned}
\mathcal{S}_{\mathcal{A}\mathcal{A}'} = U[\bar{A}_1]\cdot V[\bar{A}_0]\cdot U[\bar{A}_1] &= \frac{1}{2}\left(\mathbb{1}\otimes|0\rangle\langle 0| + \bar{A}_1\otimes|1\rangle\langle 1|\right)\cdot\\
&\cdot\left((\mathbb{1}+\bar{A}_0)\otimes\mathbb{1} + (\mathbb{1}-\bar{A}_0)\otimes\sigma_x\right)\cdot\left(\mathbb{1}\otimes|0\rangle\langle 0| + \bar{A}_1\otimes|1\rangle\langle 1|\right)\\
&= \frac{1}{2}\begin{bmatrix}\mathbb{1}+\bar{A}_0 & \bar{A}_1-\bar{A}_1\bar{A}_0\\ \bar{A}_1-\bar{A}_0\bar{A}_1 & \mathbb{1}+\bar{A}_1\bar{A}_0\bar{A}_1\end{bmatrix},
\end{aligned} \tag{166}$$

and similarly for Bob. Let us take the ancillas $|\bar{\phi}_1\rangle_{\mathcal{A}'} = |\bar{\phi}_2\rangle_{\mathcal{B}'} = |0\rangle\langle 0|$. Then, from (164) we have

$$\begin{aligned}
\rho_{\mathrm{SWAP}} &\equiv \mathrm{Tr}_{\mathcal{A}\mathcal{B}}\left((\mathcal{S}_{\mathcal{A}\mathcal{A}'}\otimes\mathcal{S}_{\mathcal{B}\mathcal{B}'})\cdot\begin{bmatrix}|\bar{\psi}\rangle\langle\bar{\psi}|_{\mathcal{A}\mathcal{B}} & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}\cdot\left(\mathcal{S}^{\dagger}_{\mathcal{A}\mathcal{A}'}\otimes\mathcal{S}^{\dagger}_{\mathcal{B}\mathcal{B}'}\right)\right)\\
&= \frac{1}{4}\mathrm{Tr}_{\mathcal{A}\mathcal{B}}\left(\begin{bmatrix}(\mathbb{1}+\bar{A}_0)(\mathbb{1}+\bar{B}_0)|\bar{\psi}\rangle\langle\bar{\psi}|_{\mathcal{A}\mathcal{B}} & 0 & 0 & 0\\ (\mathbb{1}+\bar{A}_0)(\bar{B}_1-\bar{B}_0\bar{B}_1)|\bar{\psi}\rangle\langle\bar{\psi}|_{\mathcal{A}\mathcal{B}} & 0 & 0 & 0\\ (\bar{A}_1-\bar{A}_0\bar{A}_1)(\mathbb{1}+\bar{B}_0)|\bar{\psi}\rangle\langle\bar{\psi}|_{\mathcal{A}\mathcal{B}} & 0 & 0 & 0\\ (\bar{A}_1-\bar{A}_0\bar{A}_1)(\bar{B}_1-\bar{B}_0\bar{B}_1)|\bar{\psi}\rangle\langle\bar{\psi}|_{\mathcal{A}\mathcal{B}} & 0 & 0 & 0\end{bmatrix}\cdot\right.\\
&\left.\cdot\begin{bmatrix}(\mathbb{1}+\bar{A}_0)(\mathbb{1}+\bar{B}_0) & \cdots & \cdots & \cdots\\ (\mathbb{1}+\bar{A}_0)(\bar{B}_1-\bar{B}_1\bar{B}_0) & \cdots & \cdots & \cdots\\ (\bar{A}_1-\bar{A}_1\bar{A}_0)(\mathbb{1}+\bar{B}_0) & \cdots & \cdots & \cdots\\ (\bar{A}_1-\bar{A}_1\bar{A}_0)(\bar{B}_1-\bar{B}_1\bar{B}_0) & \cdots & \cdots & \cdots\end{bmatrix}^{\mathrm{T}}\right),
\end{aligned} \tag{167}$$

where we omitted the unnecessary terms in the last matrix. One can easily see that ρ_{SWAP} is represented by a 4×4 matrix and acts on $\mathcal{A}'\otimes\mathcal{B}'$. Direct calculations show that for instance:

$$(\rho_{\mathrm{SWAP}})_{2,1}=\frac{1}{4}\langle\bar{\psi}|_{\mathcal{AB}}\left(\mathbb{1}+\bar{A}_0\right)^2\left(\mathbb{1}+\bar{B}_0\right)\left(\bar{B}_1-\bar{B}_0\bar{B}_1\right)|\bar{\psi}\rangle_{\mathcal{AB}}. \tag{168}$$

To be more specific, let us focus on the so-called elegant Bell expression [119] which has a property that whenever its Tsirelson bound is attained, then Alice's operators satisfy $\sigma_z=-\mathbb{1}+2\bar{E}_3^0=\bar{A}_0$ and $\sigma_x=-\mathbb{1}+2\bar{E}_1^0=\bar{A}_1$ $(x=1,2,3)$, and another linear functional can express σ_x and σ_z using the operators of Bob from the reduced set. This allows us to calculate e.g. lower bound on the fidelity of ρ_{SWAP} when the value of the Bell expression is slightly less than the Tsirelson bound [286].

The SWAP method has found multiple applications, in particular in the analysis of experimental data. The work [75] used the SWAP method to show that for every bipartite entangled quantum state in arbitrary dimension, there exists a behavior $\{P(a,b|x,y)\}$ (see section 1.4) allowing for self-testing of the state. In [321] the method was applied to self-test arbitrary qutrit states of the form $(2+\gamma^2)^{-\frac{1}{2}}(|00\rangle+\gamma|11\rangle+|22\rangle)$ and used for the analysis of a large scale quantum optical circuitry.

4.7. Non-commuting variables with dimension constraints

In this section, we will explore two distinct approaches that leverage moment matrices for optimizing over states and operators of a fixed dimension. The first approach, discussed in section 4.7.1, builds directly upon the NPA method and incorporates the dimension constraint as additional linear constraints. On the other hand, the second technique, presented in section 4.7.2, shares a similar structure with NPA but adopts a randomized approach to construct the basis of the space of SDP variables.

4.7.1. Dimension constraints imposed on NPA hierarchy. We present our method, which was introduced and developed in our previous works [189, 212], and is referred to as MLP hierarchy. This method enables the analysis of semi-device-independent [248] scenarios using the powerful techniques of SDP by reducing the problem to a device-independent [203] framework modeled in the NPA hierarchy.

To introduce the SDP relaxation, we first consider a device **D0**, consisting of two distinct black boxes assigned to Alice and Bob, respectively. We know the dimension of the messages exchanged between the two parts. The box of Alices generates and emits quantum states from some $\{\rho_x\}_{x\in\bar{X}}$. Bob is provided with a separate device that includes settings corresponding to measurement choices involving measurements denoted as $\left\{\{M_y^b\}_{b\in\bar{B}}\right\}_{y\in\bar{Y}}$. We denote the conditional probability of obtaining an outcome b when settings x and y are selected as $P_{\mathbf{D0}}(b|x,y)$. Suppose we are provided with a dimension witness W in the form presented in (10) earlier. Assume this dimension witness yields an average value of W_0 in experiments conducted on the **D0** device.

Although we do not know the specification of the device **D0**, we can consider an alternative device denoted as **D1**, as follows. The device **D1** comprises two components, each equipped with buttons labeled the same way as in **D0**. In **D1** we assume that both parts share a singlet state of dimension d. Alice's component performs a projective measurement with outcomes 0 (indicating successful projection) or 1 (otherwise), depending on the chosen input x. This measurement projects Alice's part of the singlet onto the state ρ_x, which corresponds to the relevant state in the device **D0**. If the projection succeeds, which occurs with a probability of

$\frac{1}{d}$, the device returns $a=0$ and transforms Alice's side into the state ρ_x. Otherwise, it returns $a=1$. Since the shared state is a singlet, this measurement prepares the same d-dimensional state on Bob's side. Bob subsequently performs the same measurements $\{M_y^b\}_b$ as the device **D0** would perform, and he returns the outcome b. Let us denote the probability that Alice obtains outcome a with setting x, while Bob obtains outcome b with setting y, as $P_{\mathbf{D1}}(a,b|x,y)$. The case where $a=1$ giving the probabilities $P_{\mathbf{D1}}(1,b|x,y)$ are not utilized in our relaxation. Trivially, we have:

$$P_{\mathbf{D0}}(b|x,y) = d \cdot P_{\mathbf{D1}}(0,b|x,y). \tag{169}$$

Let us now turn our attention to a third device, denoted as **D2**, which shares the same interface as **D1**. In contrast to the previous devices, the internal workings of **D2** are unrestricted, meaning that we make no assumptions about the measurements performed. Both Alice's and Bob's parts are now allowed to be in an arbitrary state ρ of any dimension. For this device, we give an additional constraint

$$\forall_x P_{\mathbf{D2}}(0|x) = \frac{1}{d}, \tag{170}$$

where $P_{\mathbf{D2}}(a|x)$ is the probability of getting the outcome a by Alice with the setting x with the device **D2**. It is important to highlight that the description of this device falls under the category of device-independent scenarios. Consequently, we can utilize the NPA method to mathematically represent its behavior, specifically the behavior exhibited by device **D2**. In the case of the third device, we can represent the probability of obtaining outcomes a and b for a given combination of settings x and y for Alice and Bob, respectively, as $P_{\mathbf{D2}}(a,b|x,y)$.

It is evident that all the behaviors achievable by the device **D1**, and equivalently by device **D0**, can also be obtained by device **D2**. Furthermore, as device **D2** is a relaxed variant of the original device **D0**. It is important to note that one of the key characteristics of the set of behaviors is its efficient approximation using the NPA hierarchy. An advantageous aspect of this method is that it provides a bound for any dimension of the communicated system, with the linear bound being the only parameter that needs adjustment. We conclude that using the relation $P(b|x,y) = d \cdot P(0,b|x,y)$, we can impose a dimension constraint to an existing implementation of the NPA as in (170), i.e. $P(0|x) = 1/d$.

4.7.2. NV hierarchy. Navascués and Vertesi (2014) proposed [224, 228] a hierarchy of SDPs aimed at upper bounding quantum correlation and behaviors in scenarios with dimension constraint, where similarly as in NPA, the improved accuracy is obtained when increasing the hierarchy level. The NV hierarchy is particularly useful in dimension-bounded scenarios, where the number of dimensions of the quantum systems involved is limited. In such scenarios, the full characterization of quantum correlations becomes computationally challenging due to the exponential growth of the dimension. NV provides a systematic and tractable approach to approximate and quantify quantum correlations in these scenarios.

Consider a sequence of operators representing states and measurements, denoted as $\mathcal{S} = (O_1, \ldots, O_n)$, where each operator corresponds to a specific quantum state or measurement or their polynomial function, for some N. For instance, we can have a sequence that includes identity operator $\mathbb{1}$, pure states ρ_{00}, ρ_{01}, ρ_{10}, ρ_{11}, and measurement projectors P_0^1, P_1^1, P_0^2, P_1^2. It is important to note that all states in the sequence are pure and all measurements are projectors with fixed rank. In the NV method, a crucial component is a moment matrix denoted as $M = [M]_{\mathcal{S},\mathcal{S}}$. This matrix is indexed by the sequence $\mathcal{S}$ and is defined similarly to the moment

matrix used in the NPA. The entries of the moment matrix are given by the inner products of the operators in the sequence, specifically $M_{O_i,O_j} = \mathrm{Tr}(O_i^\dagger O_j)$. It is worth noting that when the sequence $\mathcal{S}$ contains both states and measurements, the moment matrix will have entries that correspond to the probabilities $P(b|x,y)$ of obtaining outcome b when performing measurement M_b^y on the state ρ_x. This allows the moment matrix to capture the statistical information of the correlations between states and measurements, providing a framework for characterizing and quantifying the quantum correlations present in the system.

The implementation of the NV method involves a randomized approach to construct a set of moment matrices. The first step is to randomize the moment matrices, which will define the optimization problem in the SDP framework. The goal is to optimize a linear combination G of the entries $P(b|x,y)$, $G = \sum_{b,x,y} \beta_{b,x,y} P(b|x,y)$ for some fixed $\{\beta_{b,x,y}\} \subset \mathbb{R}$, which may, for instance, correspond to the average success probability of a quantum random access code. To provide more detailed steps, the implementation begins with an initialization phase where a basis of moment matrices is created:

(i) The set $\mathcal{M}$ is initialized as an empty set to store the moment matrices, $\mathcal{M} = \emptyset$.
(ii) Operators in the sequence, specific to the given hierarchy level, are randomized to form the sequence $\mathcal{S}$.
(iii) The moment matrix $\tilde{\Gamma}$ is constructed by evaluating the inner products of the operators in the randomized sequence $\mathcal{S}$, $\tilde{\Gamma} \equiv \left[\mathrm{Tr}(O_i^\dagger O_j)\right]_{\mathcal{S},\mathcal{S}}$.
(iv) The matrix $\tilde{\Gamma}^\perp$ is obtained by projecting $\tilde{\Gamma}$ onto the subspace orthogonal to the span of the moment matrices in $\mathcal{M}$.
(v) If $\tilde{\Gamma}^\perp$ is a zero matrix, the process of extending the basis of moment matrices $\mathcal{M}$ is terminated.
(vi) Otherwise, the orthogonalized moment matrix $\tilde{\Gamma}^\perp$ is added to the set $\mathcal{M}$, $\mathcal{M} = \mathcal{M} \cup \{\tilde{\Gamma}^\perp\}$. The process returns to step 2 to generate the next randomized sequence.

By iteratively adding orthogonalized moment matrices to $\mathcal{M}$, the NV implementation constructs a basis of moment matrices that captures the possible correlations in the considered system. These moment matrices form the foundation for the subsequent SDP optimization, where the objective is to find the optimal values for the entries $P(b|x,y)$ in order to maximize the value of G. The optimization has the form:

$$\begin{aligned} &\text{maximize } \mathrm{Tr}\left[\hat{\mathcal{B}}\Gamma\right] \\ &\text{subject to } \Gamma \in \mathrm{span}\,(\mathcal{M})\,, \\ &\qquad\qquad (\Gamma)_{\mathbb{1},\mathbb{1}} = 1, \\ &\qquad\qquad \Gamma \succeq 0, \end{aligned} \tag{171}$$

where we call $\hat{\mathcal{B}}$ the *game matrix* of the expression G, and construct it to select from Γ the relevant values $\mathrm{Tr}[\rho_x M_b^y]$ with coefficients defined by probability functional we are considering.

The hierarchy can get a significant boost of performance when symmetries of G are exploited [7, 302].

4.8. The see-saw iterative non-linear optimization

The see-saw method is an iterative optimization technique used in SDP to find approximate solutions for certain optimization problems [242, 328]. It is particularly effective for problems involving quantum states and measurements. The method involves alternating optimization over states and measurements, refining the solutions at each iteration until convergence is achieved. In the see-saw method, the optimization problem is first formulated as an SDP, where the objective function and constraints are expressed in terms of quantum states and measurements. The method starts with an initial guess for either the states or the measurements, which is usually a simple randomization. In each iteration, the method optimizes over the states while keeping the measurements fixed, and then optimizes over the measurements while keeping the states fixed. This alternating optimization continues until a convergence criterion is met, such as a small change in the objective function or constraints. The see-saw method leverages the interplay between states and measurements in quantum systems. By iteratively optimizing over states and measurements, the method explores different combinations that lead to improved solutions. When a set of states is given, the expression (10) can be optimized using a similar approach as in QSD [20, 142, 153] by employing SDPs, see appendix B.3. Similarly, if measurements are provided, SDP can be utilized with states as variables to optimize the expression. In cases where neither states nor measurements are known, SDP can be employed to simultaneously find optimal solutions for both. The see-saw method operates based on the following outline:

(i) Initially, a set of states is chosen randomly as an initial guess.
(ii) Using SDP, the method optimizes over measurements while keeping the states fixed. The objective is to find measurements that maximize the target expression.
(iii) Next, the method optimizes over states while keeping the measurements fixed. It uses SDP to find states that maximize the target expression.
(iv) The process iterates by returning to step 2 if certain stopping criteria are not satisfied. The stopping criteria could be based on the convergence of the objective function or other specified conditions.

It is important to note that the see-saw method does not guarantee finding the global optimum of the optimization problem. Instead, it provides an approximate solution that can be improved iteratively. To enhance the chances of finding better solutions, the method suggests restarting the process multiple times with different initial states. By repeating the see-saw method with various initial states, the hope is to explore different regions of the solution space and potentially find better solutions. Although the method may not guarantee the global optimum, it offers a practical approach for approximating the optimal solution to the optimization problem at hand.

In a related results [259] a toolbox designed to determine the optimal discrimination of optical modes in two distinct scenarios has been introduced. The first scenario, typical of metrology, involves the verifier controlling the light source and establishing a reference frame for the phase. The second scenario, more relevant to cryptography, considers cases where the verifier only observes states diagonal in the photon-number basis. The toolbox utilizes LP and SDP methods to deliver rigorous bounds for the discrimination process, enhancing the understanding and practicality of the methods in both applications.

An example of a see-saw implementation in Matlab is given in appendix B.5.

5. Conclusions

In conclusion, this paper has delved into the realm of SDP within the context of quantum information, offering a comprehensive exploration of their mathematical foundations and practical applications. By elucidating the concepts of convex optimization, duality, and SDP formulations, the study has equipped researchers and practitioners with powerful tools to address optimization challenges in quantum systems. The insights gained from the research of SDP have proven invaluable in advancing the field of quantum information, enabling the characterization and manipulation of quantum correlations, optimization of quantum states, and design of efficient quantum algorithms and protocols. The practical implementation of SDP discussed in the paper, has empowered researchers to effectively formulate and solve optimization problems in quantum systems, fostering the development of more efficient quantum communication protocols, self-testing methods, and a deeper understanding of quantum entanglement.

Data availability statement

No new data were created or analyzed in this study.

Acknowledgments

The work on this review started in 2019. The current support from the Knut and Alice Wallenberg Foundation through the Wallenberg Centre for Quantum Technology, the Swedish Research Council, and NCBiR QUANTERA/2/2020 (www.quantera.eu) an ERA-Net co-fund in Quantum Technologies under the project eDICT is acknowledged. In the period 2019–2022 the work was partially conducted at the Department of Algorithms and System Modeling at Gdańsk University of Technology and was also supported by the Foundation for Polish Science (IRAP project, ICTQT, Contract No. 2018/MAB/5, co-financed by EU within Smart Growth Operational Programme).

Recently we learned about other two independent works about SDP in quantum information [284, 300]. The content of these works and the current work are to a large extent complimentary. The current review concentrates on providing the mathematical and implementation background, by a detailed presentation of the general optimization framework, it covers the discussion of implementations of solvers and is intended primarily for active researchers in quantum information looking for the answer to the question *why* the methods work.

Appendix A. Proof of the decoupling lemma

We will now provide a proof of the decoupling lemma used in section 2.5. We follow the line of [37, 38]. The lemma is used as a constraint qualification condition, i.e. it provides the strong duality sufficient criteria of the Fenchel–Rockafellar scheme. To this end, we first introduce the concept of convex series.

A.1. Convex series

The notion of convex series was introduced in [154]; see [37, p 113nn] for a detailed discussion. *Convex series of C* are series of the form $\sum_{i=1}^{+\infty} \lambda_i x_i$ with $\forall_i x_i \in C, \forall_i \lambda_i \geqslant 0$ and $\sum_{i=1}^{+\infty} \lambda_i = 1$.

C is defined to be *convex series closed* if the sum of every convergent convex series of C is contained in C. One can check that if C is convex series closed, then [37, p 116]:

$$\text{int } C = \text{core } C. \tag{A.1}$$

C is defined to be *convex series compact* if every convex series of C converges to an element of C. It can be shown that C is convex series compact if and only if, it is convex series closed and bounded. Thus, B_X is convex series compact.

A.2. Preliminary comment

F defined as (52) is also lsc, since A is continuous. Without loss of generality, we also assume that $f(0) = g(0) = 0$, since if it is not the case, we can trivially transform the primal problem (53) shifting it by a constant value. From this it follows $F(0,0) = 0$ and

$$\forall \lambda \in [0,1], x \in X, y \in Y F(\lambda x, \lambda y) \leqslant \lambda F(x,y). \tag{A.2}$$

A.3. Step 1: define the convex set S

Let us define [37, p 127]

$$S \equiv \bigcup_{x \in B_X} \{y \in Y : F(x,y) \leqslant 1\}. \tag{A.3}$$

Let $y_0, y_1 \in S$. Then there exist $x_0, x_1 \in B_X$ such that $F(x_i, y_i) \leqslant 1$ for $i = 0, 1$. For any $\lambda \in [0,1]$ from the convexity of F we have $F(\lambda x_0 + (1-\lambda)x_1, \lambda y_0 + (1-\lambda)y_1) \leqslant \lambda F(x_0, y_0) + (1-\lambda)F(x_1, y_1) \leqslant 1$ and thus $\lambda y_0 + (1-\lambda)y_1 \in S$ implying convexity of S.

A.4. Step 2: show that $0 \in$ *core* S

From the definition (15) and the assumption given in (61) we get $\forall_{y \in Y} \exists_{T_y > 0} (T_y y) \in \text{dom}\theta = \text{dom}g - A\text{dom}f$. Consider arbitrary $y \in Y$ and take any $x_y \in \text{dom}f$ such that $T_y y + A x_y \in \text{dom}g$. Then $k_y \equiv F(x_y, T_y y) < +\infty$. We want to ensure that $\exists_{\alpha_y > 0} \alpha_y y \in S$ so that S is absorbing. Indeed, let $\alpha_y = \min(1/k_y, 1/||x_y||)$. This implies that $F(\alpha_y x_y, \alpha_y y) \leqslant 1$ (by (A.2)) and that $\alpha_y x_y \in B_X$, and thus $\alpha_y y \in S$. Since y represents arbitrary direction in Y, so $0 \in \text{core} S$ (and, what is more, S is an absorbing set).

A.5. Step 3: show that *core* S = *int* S

We will use (A.1). To this end, we check that S is convex series closed. Indeed, consider any convergent convex series of S, $\sum_{i=1}^{+\infty} \lambda_i y_i$ with $\forall_i y_i \in S$, summing to some y. It suffices to show that $y \in S$. From (A.3) it follows that $\forall_i \exists_{x_i \in B_X} F(x_i, y_i) \leqslant 1$. The series $\sum_{i=1}^{+\infty} \lambda_i x_i$ converges to some x since B_X is convex series compact. We have $F(x,y) = \sum_{i=1}^{+\infty} F(\lambda_i x_i, \lambda_i y_i) \leqslant \sum_{i=1}^{+\infty} \lambda_i F(x_i, y_i) \leqslant \sum_{i=1}^{+\infty} \lambda_i = 1$, where for the first inequality we used the assumption that F is lsc and (A.2). Thus, $y \in S$, and any convergent convex series of S has sum y contained in S meaning that S is convex series closed. Using (A.1) we conclude that for S we have core S = int S.

A.6. Step 4: show that θ is continuous in the neighborhood of 0

It can be shown [37, p 112] that for a Banach space Z a convex function $f: Z \to \mathbb{R} \cup \{+\infty\}$, locally bounded above at $z \in \mathrm{int}(\mathrm{dom} f)$ is also locally Lipschitz at z. Thus it is also *a fortiori* continuous at z. From (54) and (A.3) it follows that $\forall_{y \in S} \theta(y) \leqslant 1$, so θ is continuous at $0 \in \mathrm{int} S$.

Appendix B. Code samples in Matlab

B.1. Illustration of simple problem formulations using YALMIP

We start code discussions with a trivial example of SDP finding the largest eigenvalue of a matrix. The purpose of this example is to provide a short overview of the syntax characteristic of usage of the YALMIP modeling toolbox [193]. To install it one should follow the instructions from the repository and also install one of the supported solvers, e.g. SDPT3 [309] or SeDuMi [290, 291].

We start with randomizing a Hermitian 3 by 3 matrix `X` using the ordinary Matlab syntax. We see in listing 1 that in this instance the eigenvalues were -0.086472, 0.68428, and 3.227.

Next, we define a 3 by 3 Hermitian variable in listing 2. The function `sdpvar` is used for this purpose. The first two parameters are the number of rows n_r and columns n_c of the matrix. The third parameter specifies the structure of the variable. One of the possibilities includes `'full'` when all entries of the matrix are parameterized independently, meaning $n_\mathrm{r} n_\mathrm{c}$ parameters for real, and $2 n_\mathrm{r} n_\mathrm{c}$ parameters for complex matrices. Another possibility for the parametrization when $n_\mathrm{r} = n_\mathrm{c} = n$ is `'symmetric'` meaning that the element at ith row and jth column is exactly equal to the one at jth row and ith column, using $n(n+1)/2$ parameters for real, and n^2 parameters for complex matrices. A third possibility is `'hermitian'` meaning that the element at ith row and jth column is equal to the complex conjugate of the element at $j$row and ith column. Other possible structures are `'diagonal'` for diagonal matrices, `'toeplitz'` for symmetric Toeplitz matrices, `'hankel'` for unsymmetric Hankel matrices, `'rhankel'` for symmetric Hankel matrices, and `'skew'` for skew-symmetric matrices. When a real square matrix variable is to be created, an abbreviated form `sdpvar(n)` can be used to create n by n real symmetric matrix.

We can notice that the coefficient range is $\{1\}$, meaning that all coefficients in the variable `S` are equal to 1. It is possible to use the parameterized variables of the type `sdpvar` with various coefficient ranges. Usually, if the coefficients are spread by several orders of magnitude, meaning that the program is mixing large and small coefficients, this leads a solver to get into numerical problems. Similar problems may happen if the coefficients are very large or very small. The variables that occur with very small coefficients usually do not influence significantly the value of the solution and can be removed using the `clean` function from the YALMIP, as shown in listing 3.

Now, we will execute the optimization with the command `optimize(F = [S > = 0; trace(S) = = 1], target = -trace(X * S))`. The first argument of this function specifies the constraints of the optimization, and the second is the target of the optimization. We assigned the constraints to the variable `F`. Let us investigate this variable as shown in listing 4. We have specified the positive semi-definiteness constraint `S > = 0` on the complex 3 by 3 matrix `S`, i.e. `S > = 0`. This is described as `Matrix inequality (complex)`

```
>> X = rand(3) + 1i * rand(3); X = X + X';
>> eig(X)

ans =

   -0.086472
    0.68428
      3.227
```

Listing 1. Random matrix for the YALMIP example.

```
>> S = sdpvar(3,3,'hermitian','complex')

Linear matrix variable 3x3 (hermitian, complex, 9 variables)
Coeffiecient range: 1 to 1
```

Listing 2. Creation of a 3 by 3 Hermitian variable in YALMIP.

```
>> % large and small coefficients example
>> x = 1e9 * sdpvar(1) + 1e-9 * sdpvar(1)
Linear scalar (real, 2 variables)
Coefficients range: 1e-09 to 1000000000
>> clean(x,1e-9)
Linear scalar (real, 1 variable)
Coefficients range: 1000000000 to 1000000000
```

Listing 3. `sdpvar` with large range of coefficients, and removing small coefficients with `clean` function.

```
>> % constraints of a sample SDP
>> F
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|   ID|                          Constraint|    Coefficient range|
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|   #1|   Matrix inequality (complex) 3x3|              1 to 1|
|   #2|            Equality constraint 1x1|              1 to 1|
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Listing 4. Investigation of a sample constraint in SDP: positive semi-definiteness of `S`, i.e. `S >= 0`, and trace normalization, i.e. `trace(S) == 1`.

`3x3`. The second imposed constraint of trace normalization, `trace(S) == 1`, is described as `Equality constraint 1x1`.

An optional third argument to `optimize` can specify additional optimization parameters, like the selection of the solver, specification of how much information during the execution of the optimization should be printed to the screen, the maximal number of iterations (if the solver allows for it). For instance, to specify that from all available solvers, YALMIP should use SDPT3, not print any information, and limit the number of iterations to 20 one can provide a setting `sdpsettings('solver', 'sdpt3', 'verbose', 0, 'sdpt3.maxit',`

```
+ Solver chosen : SDPT3-4
+ Converting to real constraints
+ Processing objective function
+ Processing constraints
+ Calling SDPT3-4
```

Listing 5. Stages of YALMIP processing in the discussed example.

```
num. of constraints =  9
dim. of sdp    var  =  6,   num. of sdp  blk  =  1
dim. of free   var  =  1
```

Listing 6. Sample information about size of the problem to be passed to the solver as printed by SDPT3.

20). Another useful option `'showprogress'` allows us to see the progress of YALMIP, which is useful for debugging purposes for very large problems. Recall that with a modeling tool, before the optimization the problem is being converted to the form suitable for the solver, usually the canonical form discussed section 3.2.1, or SDPA form discussed in section 3.2.2. This formulation in some cases may take more time than the actual solver time. The option `'removeequalities'` specifies how constraints of the equalities should be preprocessed before passing to the solver, as discussed in section 3.6. The option `'dualize'` tells YALMIP to fit the problem formulation to the primal instead of the dual form. If multiple optimizations are to be executed we recommend storing the settings in a separate variable and providing it as the third optional argument to `optimize`, especially for smaller problems. The reason for this is the fact that if this argument is not provided explicitly, then YALMIP will re-create its content for each execution of `optimize`, which requires additional computational time. Turning on the option `'showprogress'` shows that the stages with the default settings (with YALMIP version 20210331) are as shown in listing 5.

At this stage, the control is passed to the specified solver. Recall from the discussion in section 1.3 that YALMIP provides to the solver the problem framed in the dual form (81). The solver usually prints the values describing the size parameters of the problem passed to the solver, as shown in listing 6 for the case of the solver SDPT3. The listing 6 shows that the dimension of the SDP variable is 6. This stems from the fact, that in the stage `Converting to real constraints` YALMIP has reformulated the n by n complex variable to $2n$ by $2n$ real variable, as discussed in section 3.5; in the considered example $n = 3$. This reformulation as a real variables problem, requires stating a requirement that the dual SDP variable Z has the form (95). It is easy to see that this requires $\frac{n\cdot(n+1)}{2}$ matrices A_i to express that the two blocks containing B^{R}, and $\frac{n\cdot(n-1)}{2}$ matrices A_i to express the relation $B^{\mathrm{I}} = -B^{\mathrm{I}}$ in the off-diagonal block. This gives in total n^2 matrices A_i, what yields 9 in this case, displayed as `number of constraints = 9`. Each of the problem's constraints framed in the dual form corresponds to an additional free variable in the primal problem, as discussed in section 3.6. The fact that there is only one equality `trace(S) = = 1` in the case, is expressed as `dim. of free var = 1` in listing 6.

We will briefly analyze the two of the settings, *viz.* `'removeequalities'` and `'dualize'`. Their default values are both 0, and this is the case analyzed in the previous paragraph. If we set `'removeequalities'` to 1, then the discussed stages will result in output given in

```
>> optimize(F = [S >= 0; trace(S) == 1], target = -trace(X * S), settings
   = sdpsettings('solver', 'sdpt3', 'showprogress', 1, 'removeequalities',
    1, 'dualize', 0))
+ Solver chosen : SDPT3-4
+ Converting to real constraints
+ Processing objective function
+ Processing constraints
+ Solving equalities
+ Converting problem to new basis
+ Calling SDPT3-4

 num. of constraints =  8
 dim. of sdp    var  =  6,   num. of sdp  blk  =  1
```

Listing 7. Size of the sample problem modelled with YALMIP with `'removeequalities'` set to 1 framed in the canonical dual form.

```
>> optimize(F = [S >= 0; trace(S) == 1], target = -trace(X * S), settings
   = sdpsettings('solver', 'sdpt3', 'dualize', 1))

 num. of constraints =  1
 dim. of sdp    var  =  6,   num. of sdp  blk  =  1
```

Listing 8. Size of the sample problem modeled with YALMIP with `'dualize'` set to 1 in order to frame the optimization problem in the canonical primal form.

listing 7. We notice that before calling the solver, two additional stages of YALMIP processing are taken, *viz.* `Solving equalities` and `Converting problem to new basis`, both responsible for the reduction of the number of variables and reformulation of the dual form, as discussed in section 3.6. Comparing with listing 6 we see that the effect is the reduction of the number of matrices A_i from $m = 9$ to $8 = m - n_{\mathrm{F}}$ and, at the same time, removal of the $n_{\mathrm{F}} = 1$ free variables. On the test platform, the problem size reduction resulted in a drop of the solver time from 0.48s to 0.24s, at the cost of additional YALMIP processing time increased to 0.14 s from 0.10 s. If we set `'dualize'` to 1, then the problem will be framed in the primal form (80) instead. Again, the complex SDP of size n will be expressed as a real SDP of size $2n$, as explained above and in section 3.5. From listing 8 we see that indeed the size of the SDP variable is 6, and there is only one constraint $\{A_i\}$ to express the requirement `trace(S) = = 1`.

The SDPT3 solver will provide also information about the chosen algorithm, as shown in listing 9. In this case, the algorithm uses the HKM search direction, see (117). Other input parameters of SDPT3 are `gam` and `expon`, and they are used to calculate the value of the step-length α_{P} and expon_used in (120) for the predictor–corrector mechanism, as discussed in section 3.9.

The core part of solving an SDP is an iterative procedure of gradual improvement of the solution $(X^{(i)}, y^{(i)}, Z^{(i)})$, with a sample progress report shown in listing 10. The first column is the iteration number, here the solver finished after the 14th iteration. The second and the third are primal and dual step-lengths, see (119), taken in each iteration. When the step-lengths are close to 1, it means that the search direction allowed for a large change in the values of $(X^{(i)}, y^{(i)}, Z^{(i)})$, what usually indicates a significant improvement of the solution in the iteration.

```
*** convert ublk to linear blk
*****************************************************************************

SDPT3: homogeneous self-dual path-following algorithms
*****************************************************************************

 version   predcorr   gam   expon
   HKM        1       0.000    1
```

Listing 9. Sample information about size of the parameters of the algorithm used by SDPT3.

```
it pstep dstep pinfeas dinfeas  gap      mean(obj)     cputime     kap    tau     theta
-----------------------------------------------------------------------------------------
 0|0.000|0.000|1.1e+01|1.2e+01|6.1e+02| 4.323372e-02| 0:0:00|6.1e+02|1.0e+00|1.0e+00| chol 1  1
 1|1.000|1.000|9.5e+00|1.0e+01|7.3e+02| 9.283979e-01| 0:0:00|4.2e+02|1.0e+00|8.1e-01| chol 1  1
 2|1.000|1.000|1.5e+00|1.7e+00|7.5e+01| 5.926998e+00| 0:0:00|8.7e+01|1.1e+00|1.5e-01| chol 1  1
 3|0.881|0.881|2.5e-01|2.8e-01|9.0e+00| 3.744834e+00| 0:0:00|8.7e-01|1.4e+00|3.2e-02| chol 1  1
 4|0.885|0.885|4.2e-02|4.7e-02|1.1e+00| 2.992843e+00| 0:0:00|1.3e+00|1.6e+00|6.2e-03| chol 1  1
 5|1.000|1.000|5.0e-03|7.5e-03|1.3e-01| 3.231258e+00| 0:0:00|2.7e-01|1.7e+00|7.7e-04| chol 1  1
 6|0.862|0.862|1.2e-03|4.6e-03|2.6e-02| 3.233302e+00| 0:0:00|8.1e-02|1.8e+00|1.9e-04| chol 1  1
 7|0.994|0.994|1.4e-04|1.6e-03|4.6e-04| 3.230829e+00| 0:0:00|1.4e-02|1.8e+00|2.2e-05| chol 1  1
 8|1.000|1.000|1.6e-05|6.1e-04|2.8e-05| 3.228448e+00| 0:0:00|1.7e-03|1.8e+00|2.5e-06| chol 1  1
 9|1.000|1.000|1.8e-06|2.4e-04|3.7e-06| 3.227543e+00| 0:0:00|1.9e-04|1.8e+00|2.9e-07| chol 1  1
10|1.000|1.000|2.1e-07|9.6e-05|5.0e-07| 3.227187e+00| 0:0:00|2.2e-05|1.8e+00|3.3e-08| chol 1  1
11|1.000|1.000|2.5e-08|1.9e-05|6.1e-08| 3.226998e+00| 0:0:00|2.6e-06|1.8e+00|3.8e-09| chol 1  1
12|1.000|1.000|5.2e-09|3.9e-06|1.6e-07| 3.226960e+00| 0:0:00|3.0e-07|1.8e+00|8.2e-10| chol 1  1
13|1.000|1.000|1.1e-09|7.7e-07|3.7e-08| 3.226953e+00| 0:0:00|6.5e-08|1.8e+00|1.8e-10| chol 1  1
14|1.000|1.000|2.5e-10|7.7e-08|8.1e-09| 3.226951e+00| 0:0:00|1.4e-08|1.8e+00|4.0e-11|
  Stop: max(relative gap,infeasibilities) < 1.00e-07
```

Listing 10. Iterations of interior-point method in SDPT3.

The fourth and fifth columns are residuals norms of the primal and dual solutions, as discussed in section 3.8. The residual norms are expected to be close to 0 when the solution is feasible, i.e. satisfies all the imposed constraints. The sixth column is proportional to the gap (112) and also should be close to 0 when the iterates approach the solution. Due to numerical inaccuracy, one usually considers values of the gap and the residual norms close to 10^{-7} as satisfactory. The 7th columns `mean(obj)` is the mean value of the primal $\mathrm{Tr}(CX^{(i)})$ and dual $b^{\mathrm{T}} \cdot y^{(i)}$ solutions of the current iterate. The 8th column provides the time passed til the current iteration (in the example it passed less than one second). The following three columns, `kap`, `tau`, `theta` provide information about specific parameters used in the determination of the step-length, and the last column provides information about the Cholesky factorization taken in calculation of the search direction; these information are beyond the scope of this work.

The SDPT3 solver provides a brief summary of its execution, as shown in listing 11. The data include the number of iterations, the value of the primal solution $\mathrm{Tr}(CX^{(i)})$, and the value of the dual solution $b^{\mathrm{T}} \cdot y^{(i)}$, the value of the gap (112), and the relative gap (i.e. the gap divided by 1 plus the mean value of the primal and dual solutions), the parameters measuring infeasibility (which should be close to 0), the norms of the solution in the last iteration $(X^{(i)}, y^{(i)}, Z^{(i)})$, the norms of the matrices defining the problem $\{A_i\}_i$, b, and C, the total CPU time, and the CPU time per iteration, the termination code. The successful termination code is 0. The values -1,-5, and -9 indicate a lack of progress, when the improvements are too slow; 1 and 2 indicate dual or primal infeasibility of the solution, -6 indicates that the maximal number of iteration has been reached before the desired quality of the solution was obtained; there is

```
number of iterations   = 14
primal objective value =  3.22695066e+00
dual   objective value =  3.22695103e+00
gap := trace(XZ)       = 8.06e-09
relative gap           = 1.91e-09
actual relative gap    = -4.99e-08
rel. primal infeas     = 2.45e-10
rel. dual   infeas     = 7.73e-08
norm(X), norm(y), norm(Z) = 4.4e+00, 8.3e-01, 1.4e+00
norm(A), norm(b), norm(C) = 2.5e+01, 4.1e+00, 1.4e+00
Total CPU time (secs)  = 0.40
CPU time per iteration = 0.03
termination code       =  0
DIMACS: 2.5e-10  0.0e+00  7.7e-08  0.0e+00  -5.0e-08  1.1e-09
```

Listing 11. Summary of SDPT3 execution.

```
corr = sdpvar(3,3,'symmetric','real')
F = [corr(1,1)==1; corr(2,2)==1; corr(3,3)==1; corr(1,2) >= 0.67; corr
    (1,2) <= 0.73; corr(1,3) >= 0.79; corr(1,3) <= 0.81; corr >= 0];
optimize(F, -corr(2,3)); corrMax = double(corr);
optimize(F, corr(2,3)); corrMin = double(corr);
```

Listing 12. Correlation matrix example in YALMIP.

also a couple of value indicating various numerical problems. The last information contains the so-called DIMACS statistics for standardized benchmarking purposes [213].

B.2. Correlation matrix

Now, we provide an illustration of the concept of SDP optimization over correlation matrices, as discussed in section 4.6. Let us consider a set $\mathcal{S} = \{x_1, x_2, x_3\}$. Suppose we have obtained information, possibly from experimental data, that $\mathrm{corr}(x_1, x_2) = 0.7 \pm 0.03$ and $\mathrm{corr}(x_1, x_3) = 0.8 \pm 0.01$. The question of interest is: what is the range of possible values for $\mathrm{corr}(x_2, x_3)$? To answer this, we construct a real 3 by 3 correlation matrix with diagonal elements equal to 1 and impose constraints on its entries based on the specified ranges. We then perform maximization and minimization of the entry corresponding to $\mathrm{corr}(x_2, x_3)$ to determine its possible range in listing 12. The results, as shown in listing 13, indicate that $\mathrm{corr}(x_2, x_3)$ lies within the interval $[0.074\,153, 0.995\,73]$.

B.3. Quantum state discrimination

In the task of discriminating N non-orthogonal quantum states, the goal is to employ a measurement strategy using operators $M_1, M_2, \ldots, M_N$ in order to maximize the average success of the discrimination. This average success is quantified by the expression $\frac{1}{N}\sum_{i\in[N]} \mathrm{Tr}(\rho_i M_i)$, where ρ_i represents the quantum state and M_i corresponds to the measurement operator for the ith state. By optimizing this expression, one can effectively differentiate between the given non-orthogonal states.

```
>> corrMax
corrMax =

1        0.73        0.79
0.73     1           0.99573
0.79     0.99573     1
>> corrMin
corrMin =

1        0.67        0.79
0.67     1           0.074153
0.79     0.074153    1
```

Listing 13. Results of the correlation matrix example calculations.

```
>> state1 = RandomState(3);
>> state2 = RandomState(3);
>> state3 = RandomState(3);
>> meas1 = sdpvar(3,3,'hermitian','complex');
>> meas2 = sdpvar(3,3,'hermitian','complex');
>> meas3 = sdpvar(3,3,'hermitian','complex');
>> optimize([meas1 >= 0; meas2 >= 0; meas3 >= 0; meas1+meas2+meas3 == eye
    (3)], -trace(state1*meas1 + state2*meas2 + state3*meas3)/3)
```

Listing 14. Quantum state discrimination in YALMIP

The Matlab code provided in listing 14 demonstrates quantum state discrimination using the YALMIP optimization toolbox. The code begins by generating three random quantum states of dimension 3: `state1`, `state2`, and `state3`. These states represent the quantum systems to be discriminated. Next, the code declares three measurement variables, `meas1`, `meas2`, and `meas3`, using the `sdpvar` function from YALMIP. These variables are Hermitian matrices of size 3 by 3, representing the measurement operators corresponding to each state. The optimization problem is formulated using the `optimize` function, which takes an objective function and a set of constraints as inputs. The objective function aims to maximize the average success rate of discrimination, given by the expression `-trace(state1*meas1 + state2*meas2 + state3*meas3)/3`. This expression calculates the average trace of the product of the state and measurement operators. The division by three accounts for the number of states being discriminated. The constraints include ensuring that each measurement operator is PSD (`meas1 >= 0, meas2 >= 0, meas3 >= 0`) and that the sum of all measurement operators equals the identity matrix (`meas1 + meas2 + meas3 == eye(3)`). These constraints guarantee that the measurement operators are valid and form a valid measurement scheme. The output of the optimization will provide the optimal measurement operators that achieve the highest discrimination performance.

We note that in the discrimination process, it is sometimes useful to consider scenarios where the weights of the different states are specified. By assigning specific weights to each state, the discrimination problem can be formulated in a more structured manner. This

```
function [rho, F, rhoSymExt] = GetDPS(dims, nCopies, PPTs, bNormalize)
% [rho, F, rhoSymExt] = GetDPS(dims, nCopies, PPTs, bNormalize = true)
%   This function creates a variable rho of the type sdpvar for YALMIP.
%   rho will satisfy the relaxed separability criteria of the Doherty-
    Parillo-Spedalieri method.
%   The variable can be later used in optimization problems involving
    separable states.
%
%   Inputs:
%     dims specifies dimensions of subsystems
%     nCopies specifies how many copies of each subsystem use
%     PPTs specifies over which copies perform the PPT (number of columns
    should be equal to total number of copies of subsystems)
%     If bNormalize == true then trace is 1; else trace is <= 1.
%   Outputs:
%     rho is the approximately system with subsystems given by dims (
    YALMIP variable)
%     F are constraints for YALMIP
%     rhoSymExt is the symmetric extension (YALMIP variable)
%
%   Example of execution: [rho, F, rhoSymExt] = GetDPS([8 2], [1 2], [1 0
    0; 0 1 0; 0 0 1]);
```

Listing 15. The header of the function creating variable with DPS constraints.

approach allows for a more targeted optimization of the average success metric, leading to enhanced discrimination capabilities. By leveraging knowledge about the weights of the states, researchers can design measurement schemes and strategies that are tailored to maximize the overall success rate in discriminating the non-orthogonal quantum states.

B.4. Implementation of the Doherty–Parillo–Spedalieri method

The code from listing 15 defines a header of a function called `GetDPS`, which is designed to create YALMIP variables for quantum states, along with associated constraints, specifically tailored for applying the relaxed separability criteria of the DPS method discussed in section 4.2. These variables and constraints can be used for conducting optimizations involving separable quantum states. The function takes inputs such as the dimensions of subsystems, the number of copies, and the configuration of PPT tests. The `bNormalize` parameter allows to choose whether the trace of the resulting state should be exactly 1 or not greater than 1. The function outputs the YALMIP variable `rho`, the corresponding YALMIP constraints `F`, and the symmetric extension variable `rhoSymExt`.

The part of the code from listing 16 performs some initial calculations and checks based on the provided input. It ensures that the boolean variable `bNormalize` is set to `true` if not explicitly specified, confirming that trace normalization is the default behavior. The code then asserts that the dimensions of subsystems and the number of copies match and that the total number of copies aligns with the number of columns in the provided PPTs matrix. The number of subspaces is determined, considering the input dimensions and the number of copies. The dimensions of the symmetric extension are computed, including those for subsystem copies, which are identified. Finally, the total dimension of the symmetric extension is calculated by

```
if nargin < 4
  bNormalize = true;
end

assert(size(dims) == size(nCopies))
assert(sum(nCopies) == size(PPTs, 2))

nSubspaces = length(dims); % number of subspaces of the resulting
 product state
dimSymExt = []; % dimensions of subsystems (including copies)
subsystemsPos = [1 cumsum(nCopies)+1](1:nSubspaces); % where each
 subsystem has its first copy
subsystemsCopiesPos = setdiff(1:sum(nCopies), subsystemsPos); % which
 subsystems are copies
for ii = 1:nSubspaces % dimensions of copies of each subspace
  dimSymExt = [dimSymExt dims(ii) * ones(1, nCopies(ii))];
end
totalDim = prod(dimSymExt); % total dimension of symmetric extension
```

Listing 16. Initialization of variables and input validation for the function creating variable with DPS constraints.

```
% create basis of symmetric extension matrices
basis = [1]; % this will contain all vectors spanning symmetric
 extension subspace
for ii = 1:nSubspaces
  [vv, dd] = eig(full(SymmetricProjection(dims(ii), nCopies(ii))));
  basis = kron(basis, vv(:, find(diag(dd) > 0.5)) ); % find those
 eigenvalues that are equal 1 (knowing they are either 0 or 1)
end
nBasis = size(basis, 2);
```

Listing 17. Calculation of the basis of symmetric extension matrices in the function creating variable with DPS constraints.

taking the product of the individual dimensions. These preliminary steps ensure the consistency and validity of the input.

The code given in listing 17 focuses on creating a basis for the symmetric extension matrices. It begins with an initial basis vector `[1]`, which will accumulate the vectors spanning the symmetric extension subspace. For each subspace specified in the input, it computes the eigenvalues and eigenvectors of a symmetric projection matrix constructed based on the dimensions and the number of copies using a relevant subroutine `SymmetricProjection` from the package QETLAB [162]. Note that the eigenvalues of the symmetric projection are either 0 or 1, and thus the symmetric subspace can be determined by selecting the eigenvectors corresponding to eigenvalues greater than, for instance, 0.5. The basis of all vectors spanning symmetric extension subspace stored in the variable `basis` is updated accordingly using the Kronecker product. This process continues for all subspaces, ultimately generating a basis with vectors that span the symmetric extension space. The variable `nBasis` records the number of basis vectors.

```
% create rhoSymExt variable
rhoSymBasis = sdpvar(nBasis, nBasis, 'hermitian', 'complex');
rhoSymExt = basis * rhoSymBasis * basis';
rhoSymExt = clean(rhoSymExt, 1e-9); % avoid small terms occuring in
 numerical eigen-decomposition
rhoSymExt = 0.5 * (rhoSymExt + rhoSymExt'); % correct small numerical
 inaccuracies to restore the Hermicity of the matrix
F = [(rhoSymBasis >= 0):'GetDPS, rhoSymBasis >= 0']; % F is the variable
  for storing the constraints
if bNormalize
  F = [F; (trace(rhoSymExt) == 1):'GetDPS, trace(rhoSymExt) == 1'];
else
  F = [F; (trace(rhoSymExt) <= 1):'GetDPS, trace(rhoSymExt) <= 1'];
end
```

Listing 18. Creation of a variable containing a symmetric extension of the function creating variable with DPS constraints.

```
% create rho variable
rho = TrX(rhoSymExt, subsystemsCopiesPos, dimSymExt);

% apply PPTs to subsystems
for PPT = PPTs'
  F = [F; (Tx(rhoSymExt, find(PPT), dimSymExt) >= 0):['GetDPS, one of
 PPTs: ' num2str(PPT')]]; % add all selected PPT constraints
end
```

Listing 19. Application of the PPT constraints on the approximately separable quantum state in the function creating variable with DPS constraints.

The code from listing 18 is responsible for creating the `rhoSymExt` variable, which represents the symmetric extension of a quantum state. It starts by defining `rhoSymBasis`, which is an `nBasis` by `nBasis` Hermitian complex matrix. The symmetric extension matrix `rhoSymExt` is then constructed by performing matrix operations involving the basis vectors. After constructing `rhoSymExt`, a cleaning operation is applied to remove small numerical artifacts that could affect the numerical calculations during SDP optimizations. The variable `F` is used to store constraints associated with `rhoSymBasis`. If the `bNormalize` flag is set, the trace constraint enforces that the trace of `rhoSymExt` equals 1, indicating a normalized quantum state.

The part of the code from listing 19 is responsible for creating the `rho` variable, which represents the quantum state constrained to satisfy the selected DPS criteria. It is derived from its previously created symmetric extension `rhoSymExt` by tracing out the copies of subsystems (as specified by the variable `subsystemsCopiesPos`). Following that, the code iterates over the specified on the input PPT checks (passed to the function in the variable `PPTs`). Each PPT check corresponds to a constraint that enforces the partial transpose of the symmetric extension of the state to be PSD. The partial transposition is performed using the function `Tx` from the package Quantinf [77]. The constraints are added to the variable `F`, which collects all the constraints to be used in later optimization problems.

Let us now use the defined function to the example from section VII.B from the paper by Doherty *et al* [87]. We define auxiliary lambda functions `ket` and `bra` that are creating vectors

```
ket = @(ii,jj) VersorD(ii+4*jj+1, 16);
bra = @(ii,jj) ket(ii,jj)';
W = (ket(2,2)-ket(0,0))*(bra(2,2)-bra(0,0)) + (ket(2,2)-ket(1,1))*(bra
    (2,2)-bra(1,1)) + (ket(3,3)-ket(0,1))*(bra(3,3)-bra(0,1)) + (ket(3,3)-
    ket(1,0))*(bra(3,3)-bra(1,0)) + ket(2,3)*bra(2,3) + ket(3,2)*bra(3,2) -
     ket(2,2)*bra(2,2) - ket(3,3)*bra(3,3);
[rhoAB, F, rhoABB] = GetDPS([4 4], [1 2], [1 0 0; 0 1 0]); % two copies of
    the second system
target = trace(W * rhoAB);
optimize(F, target, sdpsettings('solver', 'sdpt3', 'verbose', 0));
assert(norm(value(target)) < 1e-8)
```

Listing 20. Illustration of working of the function creating variable with DPS constraints applied on a separability witness .

$|i\rangle \otimes |j\rangle$ and $\langle i| \otimes \langle j|$ in two ququart spaces. Next, we use them to express the witness `W` which is non-negative on all product states, as derived in [87]. Next, we use the function `GetDPS` to create a variable under DPS constraints, where the second subsystem has two copies and two PPT constraints are imposed: for partial transposition of the second and third subsystems. The optimization shows that the maximal value of the witness possible to be obtained by states satisfying these constraints is 0, as expected. A unit testing example is shown in listing 20.

The second example involves the NTV method discussed in section 4.2, as shown in the unit test from listing 21. We use the Bell functional $P(0,0|1,1)+P(0,0|1,2)+P(0,0|2,1)-P(0,0|2,2)-P_A(0|1)-P_B(0|1)$ with the Tsirelson bound $\frac{1}{\sqrt{2}}-\frac{1}{2}$ [52]. This Bell functional is a form of the CHSH functional [73]. The `GetDPS` function is used to prepare a variable `W` that satisfies the relaxed separability criteria DPS method. The dimensions of the four subsystems are specified as `[4 2 2]`. The first two subsystems are passed jointly as a four dimensional subsystem to reflect that they represent an entangled two-qubit state. The remaining two subsystems represent single qubit measurements. This code creates operators for performing measurements on the entangled state in the following way. The `PA1` operator represents SWAP operators between Alice's qubit and Alice's first measurement stored in the third subsystem. Similarly `PB1` is the SWAP between Bob's qubit and Bob's measurement contained in the fourth subsystem of `W`. They are created using the `sysexchange` function from the mentioned Quantinf package. `PA2` and `PB2` are the computational basis measurements for Alice and Bob, respectively. The target variable is calculated as the trace of the operator `W` times the `CHSH` operator. The code uses YALMIP's optimize function to solve the SDP problem, with constraints defined in F, and the goal is to maximize the target value. Finally, it asserts that the computed value of the target is close to the expected value.

B.5. Implementation of the see-saw method

We will demonstrate a simple implementation of the see-saw technique discussed in section 4.8.

To this end, let us now delve into the concept of a random access code (RAC), which aims to compress a uniformly randomized n-digit string into a single digit, allowing Bob to recover any of the n digits with a high probability [12, 13]. In this scenario, Alice receives a uniformly distributed random input string $\mathbf{x}$ comprising n digits. She computes the value a using the function $f[\mathbf{x}]$ and transmits it to Bob. Upon receiving a and another uniformly distributed input y,

```
[W, F] = GetDPS([4 2 2], [1 1 1], [1 0 0; 0 1 0; 0 0 1]); % the first two
    subsystems of W contain the two qubit state (dimension 4), the
    remaining two are storing representations of measurements

% select measurement results
swapOp = kron(SwapOperator(2), eye(4));
PA1 = sysexchange(swapOp, [2 3], [2 2 2 2]); % SWAP operator between 1st
    and 3rd subsystems
PA2 = kron([1 0; 0 0], eye(8)); % computational basis measurement of Alice
PB1 = sysexchange(swapOp, [1 4], [2 2 2 2]); % SWAP operator between 2nd
    and 4th subsystems
PB2 = kron(eye(2), [1 0; 0 0], eye(4)); % computational basis measurement
    of Bob

% solve SDP
CHSH = PA1*PB1 + PA1*PB2 + PA2*PB1 - PA2*PB2 - PA1 - PB1; % arXiv:0702130
    v2 eq. (1): P(0,0|1,1) + P(0,0|1,2) + P(0,0|2,1) - P(0,0|2,2) - P_A
    (0|1) - P_B(0|1)
target = trace(W * CHSH);
optimize(F, -target, sdpsettings('solver', 'sdpt3', 'verbose', 0));

% check
assert(norm(value(target) - (sqrt(2)-1)/2) < 1e-8)
```

Listing 21. Illustration of working of the function creating variable with DPS constraints for implementation of the Navascués-de la Torre-Vértesi (NTV) method to find the Tsirelson bound of a variant of the CHSH Bell functional.

Bob employs the function $g(a,y)$ to compute the value b. The protocol is considered successful if b is equal to the yth digit of $\mathbf{x}$.

Let us explore a specific case of a $2^2 \to 1$ RAC. Here, Alice aims to encode two bits into a single bit. The mappings providing the optimal success probability of the task are as follows: $f(00)=0, f(01)=0, f(10)=1, f(11)=1$. On Bob's side, the decoding is defined as follows: $g(a,1)=a$ and $g(a,2)=0$. The success probability of this classical RAC is $\frac{3}{4}$.

In the quantum scenario, Alice tries to encode two bits into a qubit system. Here an encoding allowing to get the optimal quantum value is as follows. Take

$$f(00)=\begin{bmatrix}\cos\frac{\pi}{8}\\ \sin\frac{\pi}{8}\end{bmatrix}, f(01)=\begin{bmatrix}\cos\frac{\pi}{8}\\ -\sin\frac{\pi}{8}\end{bmatrix}, f(10)=\begin{bmatrix}\sin\frac{\pi}{8}\\ \cos\frac{\pi}{8}\end{bmatrix}, f(11)=\begin{bmatrix}\sin\frac{\pi}{8}\\ -\cos\frac{\pi}{8}\end{bmatrix}. \tag{B.1}$$

Bob performs his decoding using the projectors

$$P_0^1=\begin{bmatrix}1&0\\0&0\end{bmatrix}, P_1^1=\begin{bmatrix}0&0\\0&1\end{bmatrix}, P_0^2=\begin{bmatrix}\frac{1}{2}&\frac{1}{2}\\ \frac{1}{2}&\frac{1}{2}\end{bmatrix}, P_1^2=\begin{bmatrix}\frac{1}{2}&-\frac{1}{2}\\ -\frac{1}{2}&\frac{1}{2}\end{bmatrix}. \tag{B.2}$$

The optimal success probability is $\frac{1}{2}\left(1+\frac{\sqrt{2}}{2}\right)\approx 0.853\,55$.

The first step in the implementation of see-saw in Matlab with the YALMIP toolbox is to provide a proper definition of the variables used to express the quantum states for the encoding, see (B.1). We provide such code in listing 22. `Frho` is a critical component in the formulation of constraints that guarantee the validity of the operators stored within the cell-type data structure `rhoCellVar` as quantum qubit states. These constraints collectively ensure that the stored

```
% variables for states
Frho = []; % we will place constraints for quantum states here
rhoCellVar = cell(2,2); % we will store quantum states here
for x1 = 1:2
  for x2 = 1:2
    rhoCellVar{x1,x2} = sdpvar(2,2,'hermitian','complex');
    Frho = [Frho; trace(rhoCellVar{x1,x2}) == 1; rhoCellVar{x1,x2} >= 0];
  end
end
```

Listing 22. Preparation of SDP variables to store the optimized quantum states in the $2^2 \to 1$ quantum RAC for see-saw implementation with YALMIP.

```
% variables for measurements
mCellVar = cell(2,2); % row indicates which measurement, column which
    output
Fmeas = [];
for y = 1:2
  for b = 1:2
    mCellVar{y,b} = sdpvar(2,2,'hermitian','complex');
    Fmeas = [Fmeas; mCellVar{y,b} >= 0];
  end
end
Fmeas = [Fmeas; mCellVar{1,1}+mCellVar{1,2} == eye(2); mCellVar{2,1}+
    mCellVar{2,2} == eye(2)]; % measurements sum to identity
```

Listing 23. Preparation of SDP variables to store the optimized quantum measurements in the $2^2 \to 1$ quantum RAC for see-saw implementation with YALMIP.

density matrices meet the essential criteria for a valid quantum state: they are Hermitian, PSD, and normalized. Note that even though the optimal states in (B.1) are pure, the form of SDP constraints allows for expressing them only as density matrices.

The next code segment from listing 23 focuses on defining variables and constraints related to measurements. The `mCellVar` cell array is employed to store measurement operators, and the `Fmeas` set of constraints is used to ensure the validity of these operators. The constraints stipulate that the measurement operators must be Hermitian, PSD, and sum to the identity matrix, i.e. form a POVM. The nested loops over `y` and `b` create these measurement operators and add the corresponding constraints to the `Fmeas` list variable. The constraint of summation to the identity in this example is given outside the loops.

The code snippet from listing 24 is setting up the remaining initialization for the see-saw optimization. The `Succ` variable is defined as a success probability function for the QRAC. This function is defined as the so-called lambda expression taking two parameters, and it computes the success probability based on input density matrices `rhoCellIn` and measurement operators `mCellIn` for the given quantum states and measurements. The nested loops initialize quantum states, i.e. `rhoCell` with random density matrices using a `RandomState` function, ensuring that the dimension is set to 2. Additionally, an empty cell array `mCell` is created to store the current measurement operators in the 'saw' step. Finally, we choose the solver settings, with SDPT3 solver in this example.

The major element of the see-saw is shown in listing 25 which represents the process involving three iterations. The objective of every step is to maximize the success probability of

```
% other initialization
Succ = @(rhoCellIn,mCellIn) trace(rhoCellIn{1,1}*(mCellIn{1,1}+mCellIn
    {2,1}) + rhoCellIn{1,2}*(mCellIn{1,1}+mCellIn{2,2}) + rhoCellIn{2,1}*(
    mCellIn{1,2}+mCellIn{2,1}) + rhoCellIn{2,2}*(mCellIn{1,2}+mCellIn{2,2})
    ) / 8; % success function

% seed with random states
rhoCell = cell(2,2);
for x1 = 1:2
  for x2 = 1:2
    rhoCell{x1,x2} = RandomState(2);
  end
end

mCell = cell(2,2); % to store measurements

settings = sdpsettings('solver', 'sdpt3'); % choose SDPT3 as the solver
```

Listing 24. Initialization of values of the encoding states with random values, and definition of the success probability in the $2^2 \to 1$ quantum RAC for see-saw implementation with YALMIP.

```
for iter = 1:3 % stopping criterion: just three iterations

  % the "SEE" step
  optimize(Fmeas, -Succ(rhoCell, mCellVar), settings); % optimize
    measurements for given states
  for x1 = 1:2
    for x2 = 1:2
      mCell{x1,x2} = double(mCellVar{x1,x2}); % store measurements as
    constants
    end
  end
  progressTab(1, iter) = Succ(rhoCell, mCell); % "see" progress

  % the "SAW" step
  optimize(Frho, -Succ(rhoCellVar, mCell), settings); % optimize states
    for given measurements
  for x1 = 1:2
    for x2 = 1:2
      rhoCell{x1,x2} = double(rhoCellVar{x1,x2}); % store states as
    constants
    end
  end
  progressTab(2, iter) = Succ(rhoCell, mCell); % "saw" progress

end
```

Listing 25. Intertwined 'see' and 'saw' steps in the $2^2 \to 1$ quantum RAC in the see-saw implementation example with YALMIP.

the quantum RAC by alternately optimizing measurement operators and quantum states. The loop consists of two main steps: the 'see' step and the 'saw' step. In the 'see' steps, the code optimizes the measurement operators `mCellVar` while keeping the quantum states fixed in `rhoCell`. After each optimization, the optimized measurements are stored as constants in the `mCell` variable. Then, the success probability with these updated measurements is computed and recorded in the `progressTab` array. In the 'saw' step, the code optimizes the quantum states `rhoCellVar` while keeping the measurements fixed in `mCell`. Similar to the 'see' step, the optimized states are stored as constants in the `rhoCell` variable. The success probability with these updated states is again computed and recorded in the `progressTab` array. The process continues for just three iterations as specified. The stopping criteria for the process can be customized based e.g. on progress in optimizing the target value. Already the second iteration provides the optimal result in this simple case.

ORCID iD

Piotr Mironowicz https://orcid.org/0000-0003-4122-5372

References

[1] 2021 *MOSEK Modeling Cookbook* 3.2.3 edn (MOSEK ApS) (available at: https://docs.mosek.com/modeling-cookbook/intro.html)
[2] 2022 *MATLAB 9.13.0.2126072 (R2022b) Update 3* (The MathWorks Inc.) (available at: https://www.mathworks.com/products/matlab.html)
[3] Acín A, Fritz T, Leverrier A and Sainz A B 2015 A combinatorial approach to nonlocality and contextuality *Commun. Math. Phys.* **334** 533–628
[4] Agarwal H and Garg I 2022 A brief review of operator monotone and operator convex functions *J. Phys.: Conf. Ser.* **2267** 012087
[5] Agrawal A, Verschueren R, Diamond S and Boyd S 2018 A rewriting system for convex optimization problems *J. Control Decis.* **5** 42–60
[6] Agresti I, Polacchi B, Poderini D, Polino E, Suprano A, Šupić I, Bowles J, Carvacho G, Cavalcanti D and Sciarrino F 2021 Experimental robust self-testing of the state generated by a quantum network *PRX Quantum* **2** 020346
[7] Aguilar E A, Borkała J J, Mironowicz P and Pawłowski M 2018 Connections between mutually unbiased bases and quantum random access codes *Phys. Rev. Lett.* **121** 050501
[8] Alizadeh F 1991 Combinatorial optimization with interior point methods and semi-definite matrices *PhD Thesis* University of Minnesota
[9] Alizadeh F 1995 Interior point methods in semidefinite programming with applications to combinatorial optimization *SIAM J. Optim.* **5** 13–51
[10] Alizadeh F, Haeberly J-P A and Overton M L 1998 Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results *SIAM J. Optim.* **8** 746–68
[11] Allcock J, Brunner N, Linden N, Popescu S, Skrzypczyk P and Vértesi T 2009 Closed sets of nonlocal correlations *Phys. Rev.* A **80** 062107
[12] Ambainis A, Leung D, Mancinska L and Ozols M 2008 Quantum random access codes with shared randomness (arXiv:0810.2937)
[13] Ambainis A, Nayak A, Ta-Shma A and Vazirani U 1999 Dense quantum coding and a lower bound for 1-way quantum automata *Proc. 31st Annual ACM Symposium on Theory of Computing* pp 376–83
[14] Ando T 1978 *Topics on Operator Inequalities (Lecture Note Series)* (Ryukyu University)
[15] Anjos M F and Lasserre J B 2011 *Handbook on Semidefinite, Conic and Polynomial Optimization* vol 166 (Springer Science & Business Media)
[16] Aspect A, Dalibard J and Roger G 1982 Experimental test of Bell's inequalities using time-varying analyzers *Phys. Rev. Lett.* **49** 1804

[17] Aspect A, Grangier P and Roger G 1981 Experimental tests of realistic local theories via Bell's theorem *Phys. Rev. Lett.* **47** 460
[18] Aspect A, Grangier P and Roger G 1982 Experimental realization of Einstein-Podolsky-Rosen-Bohm Gedankenexperiment: a new violation of Bell's inequalities *Phys. Rev. Lett.* **49** 91
[19] Baccari F, Cavalcanti D, Wittek P and Acín A 2017 Efficient device-independent entanglement detection for multipartite systems *Phys. Rev.* X **7** 021042
[20] Bae J and Kwek L-C 2015 Quantum state discrimination and its applications *J. Phys. A: Math. Theor.* **48** 083001
[21] Bamps C and Pironio S 2015 Sum-of-squares decompositions for a family of Clauser-Horne-Shimony-Holt-like inequalities and their application to self-testing *Phys. Rev.* A **91** 052111
[22] Bancal J-D, Sheridan L and Scarani V 2014 More randomness from the same data *New J. Phys.* **16** 033011
[23] Beavis B and Dobbs I 1990 *Optimisation and Stability Theory for Economic Analysis* (Cambridge University Press)
[24] Beck A 2014 *Introduction to Nonlinear Optimization: Theory, Algorithms and Applications With Matlab* (SIAM)
[25] Bell J S 1964 On the Einstein Podolsky Rosen paradox *Physics* **1** 195
[26] Bell J S 2004 *Speakable and Unspeakable in Quantum Mechanics: Collected Papers on Quantum Philosophy* (Cambridge University Press)
[27] Ben-Tal A and Nemirovski A 2011–2012 *Lectures on Modern Convex Optimization* (SIAM)
[28] Bengtsson I and Życzkowski K 2017 *Geometry of Quantum States: An Introduction to Quantum Entanglement* (Cambridge University Press)
[29] Benson S J and Ye Y 2006 DSDP5 user guide-software for semidefinite programming, *Technical Report* ANL/MCS-P1289-0905 (Argonne National Lab. (ANL))
[30] Benson S J, Ye Y and Zhang X 2000 Solving large-scale sparse semidefinite programs for combinatorial optimization *SIAM J. Optim.* **10** 443–61
[31] Bernards F and Gühne O 2020 Generalizing optimal Bell inequalities *Phys. Rev. Lett.* **125** 200401
[32] Bezanson J, Edelman A, Karpinski S and Shah V B 2017 Julia: a fresh approach to numerical computing *SIAM Rev.* **59** 65–98
[33] Bhatia R 2009 *Positive Definite Matrices, in Positive Definite Matrices* (Princeton University Press)
[34] Bochnak J, Coste M and Roy M-F 2013 *Real Algebraic Geometry* vol 36 (Springer)
[35] Borchers B 1999 CSDP, a C library for semidefinite programming *Opt. Methods Softw.* **11** 613–23
[36] Borwein J M and Lewis A S 2006 *Convex Analysis and Nonlinear Optimization (CMS Books in Mathematics)* (Springer)
[37] Borwein J M and Zhu Q J 2005 *Techniques of Variational Analysis (CMS Books in Mathematics)* (Springer)
[38] Borwein J M and Zhu Q J 2006 Variational methods in convex analysis *J. Glob. Opt.* **35** 197–213
[39] Bot R I, Grad S-M and Wanka G 2009 *Duality in Vector Optimization (Vector Optimization)* (Springer)
[40] Boyd S, El Ghaoui L, Feron E and Balakrishnan V 1994 *Linear Matrix Inequalities in System and Control Theory* (SIAM)
[41] Boyd S P and Vandenberghe L 2004 *Convex Optimization* (Cambridge University Press)
[42] Brandao F G S L and Vianna R O 2004 Robust semidefinite programming approach to the separability problem *Phys. Rev.* A **70** 062309
[43] Brassard G, Buhrman H, Linden N, Méthot A A, Tapp A and Unger F 2006 Limit on nonlocality in any world in which communication complexity is not trivial *Phys. Rev. Lett.* **96** 250401
[44] Briët J and Zuiddam J 2016 On the orthogonal rank of Cayley graphs and impossibility of quantum round elimination *Quantum Inf. Comput.* **17** 0106
[45] Brown P 2019 On constructions of quantum-secure device-independent randomness expansion protocols *PhD Thesis* University of York
[46] Brown P, Fawzi H and Fawzi O 2021 Computing conditional entropies for quantum correlations *Nat. Commun.* **12** 1–12
[47] Brown P, Fawzi H and Fawzi O 2021 Device-independent lower bounds on the conditional von Neumann entropy (arXiv:2106.13692)
[48] Brown P J 2023 Examples of scripts implementing the method (arXiv:2106.13692)
[49] Brown P J 2022 Example scripts for computing rates of device-independent protocols (available at: https://github.com/peterjbrown519/DI-rates)

[50] Brown P J, Ragy S and Colbeck R 2019 A framework for quantum-secure device-independent randomness expansion *IEEE Trans. Inf. Theory* **66** 2964–87
[51] Brukner Č 2011 Questioning the rules of the game *Physics* **4** 55
[52] Brunner N, Gisin N, Scarani V and Simon C 2007 Detection loophole in asymmetric Bell experiments *Phys. Rev. Lett.* **98** 220403
[53] Brunner N, Pironio S, Acin A, Gisin N, Méthot A A and Scarani V 2008 Testing the dimension of Hilbert spaces *Phys. Rev. Lett.* **100** 210503
[54] Bu K, Singh U, Fei S-M, Pati A K and Wu J 2017 Maximum relative entropy of coherence: an operational coherence measure *Phys. Rev. Lett.* **119** 150405
[55] Burgdorf S, Cafuta K, Klep I and Povh J 2013 The tracial moment problem and trace-optimization of polynomials *Math. Program.* **137** 557–78
[56] Cabello A 2021 Converting contextuality into nonlocality *Phys. Rev. Lett.* **127** 070401
[57] Cabello A, Severini S and Winter A 2010 (Non-)contextuality of physical theories as an axiom (arXiv:1010.2163)
[58] Cabello A, Severini S and Winter A 2014 Graph-theoretic approach to quantum correlations *Phys. Rev. Lett.* **112** 040401
[59] Cameron P J, Montanaro A, Newman M W, Severini S and Winter A 2007 On the quantum chromatic number of a graph *Electron. J. Comb.* **14** R81
[60] Carlen E 2010 Trace inequalities and quantum entropy: an introductory course *Entropy and the Quantum* vol 529 (American Mathematical Society) pp 73–140
[61] Cavalcanti D and Skrzypczyk P 2016 Quantum steering: a review with focus on semidefinite programming *Rep. Prog. Phys.* **80** 024001
[62] Cavalcanti D, Skrzypczyk P, Aguilar G H, Nery R V, Ribeiro P H S and Walborn S P 2015 Detection of entanglement in asymmetric quantum networks and multipartite quantum steering *Nat. Commun.* **6** 7941
[63] Chaturvedi A, Farkas M and Wright V J 2021 Characterising and bounding the set of quantum behaviours in contextuality scenarios *Quantum* **5** 484
[64] Chaturvedi A and Saha D 2020 Quantum prescriptions are more ontologically distinct than they are operationally distinguishable *Quantum* **4** 345
[65] Chaturvedi A, Viola G and Pawłowski M 2022 Extending loophole-free nonlocal correlations to arbitrarily large distances (arXiv:2211.14231)
[66] Cheng J T W and Zhang S 2006 On implementation of a self-dual embedding method for convex programming *Opt. Methods Softw.* **21** 75–103
[67] Chernyshenko S I, Goulart P, Huang D and Papachristodoulou A 2014 Polynomial sum of squares in fluid dynamics: a review with a look ahead *Phil. Trans. R. Soc.* A **372** 20130350
[68] Chesi G 2010 LMI techniques for optimization over polynomials in control: a survey *IEEE Trans. Autom. Control* **55** 2500–10
[69] Chiribella G, D'Ariano G M and Perinotti P 2011 Informational derivation of quantum theory *Phys. Rev.* A **84** 012311
[70] Choi M-D 1975 Completely positive linear maps on complex matrices *Linear Algebr. Appl.* **10** 285–90
[71] Chung F R 1997 *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics)* (University of Pennsylvania)
[72] Cirel'son B S 1980 Quantum generalizations of Bell's inequality *Lett. Math. Phys.* **4** 93–100
[73] Clauser J F, Horne M A, Shimony A and Holt R A 1969 Proposed experiment to test local hidden-variable theories *Phys. Rev. Lett.* **23** 880
[74] Clifton R, Bub J and Halvorson H 2003 Characterizing quantum theory in terms of information-theoretic constraints *Found. Phys.* **33** 1561–91
[75] Coladangelo A, Goh K T and Scarani V 2017 All pure bipartite entangled states can be self-tested *Nat. Commun.* **8** 15485
[76] Cover T M and Thomas J A 2012 *Elements of Information Theory* (Wiley)
[77] Cubitt T 2013 Quantum information package (available at: www.dr-qubit.org/matlab.html)
[78] Cubitt T S, Leung D, Matthews W and Winter A 2011 Zero-error channel capacity and simulation assisted by non-local correlations *IEEE Trans. Inf. Theory* **57** 5509–23
[79] Czekaj L, Horodecki M, Horodecki P and Horodecki R 2017 Information content of systems as a physical principle *Phys. Rev.* A **95** 022119
[80] Dantzig G B 1990 Origins of the simplex method *A History of Scientific Computing* ed S G Nash (Association for Computing Machinery) pp 141–51

[81] Dattorro J 2010 Convex optimization & Euclidean distance geometry (available at: Lulu.com)
[82] de Gois C, Hansenne K and Gühne O 2023 Uncertainty relations from graph theory *Phys. Rev.* A **107** 062211
[83] De Wolf R 2001 *Quantum Computing and Communication Complexity* (University of Amsterdam)
[84] Diamond S and Boyd S 2016 CVXPY: a Python-embedded modeling language for convex optimization *J. Mach. Learn. Res.* **17** 1–5
[85] Doherty A C, Liang Y-C, Toner B and Wehner S 2008 The quantum moment problem and bounds on entangled multi-prover games *2008 23rd Annual IEEE Conf. on Computational Complexity* (IEEE) pp 199–210
[86] Doherty A C, Parrilo P A and Spedalieri F M 2002 Distinguishing separable and entangled states *Phys. Rev. Lett.* **88** 187904
[87] Doherty A C, Parrilo P A and Spedalieri F M 2004 Complete family of separability criteria *Phys. Rev.* A **69** 022308
[88] Duan R, Severini S and Winter A 2012 Zero-error communication via quantum channels, noncommutative graphs and a quantum Lovász number *IEEE Trans. Inf. Theory* **59** 1164–74
[89] Dunning I, Huchette J and Lubin M 2017 JuMP: a modeling language for mathematical optimization *SIAM Rev.* **59** 295–320
[90] Eaton J W, Bateman D, Hauberg S and Wehbring R 2020 GNU Octave version 6.1.0 manual: a high-level interactive language for numerical computations
[91] Ebadian A, Nikoufar I and Gordji M E 2011 Perspectives of matrix convex functions *Proc. Natl Acad. Sci.* **108** 7313–4
[92] Effros E and Hansen F 2014 Non-commutative perspectives *Ann. Funct. Anal.* **5** 74–79
[93] Effros E G 2009 A matrix convexity approach to some celebrated quantum inequalities *Proc. Natl Acad. Sci.* **106** 1006–8
[94] Eisert J 2005 Optimizing linear optics quantum gates *Phys. Rev. Lett.* **95** 040502
[95] Eldar Y C 2003 A semidefinite programming approach to optimal unambiguous discrimination of quantum states *IEEE Trans. Inf. Theory* **49** 446–56
[96] Fang K and Fawzi H 2021 Geometric Rényi divergence and its applications in quantum channel capacities *Commun. Math. Phys.* **384** 1615–77
[97] Fannes M, Lewis J T and Verbeure A 1988 Symmetric states of composite systems *Lett. Math. Phys.* **15** 255–60
[98] Fawzi H and Fawzi O 2018 Efficient optimization of the quantum relative entropy *J. Phys. A: Math. Theor.* **51** 154003
[99] Fawzi H and Saunderson J 2017 Lieb's concavity theorem, matrix geometric means and semidefinite optimization *Linear Algebr. Appl.* **513** 240–63
[100] Fawzi H, Saunderson J and Parrilo P A 2019 Semidefinite approximations of the matrix logarithm *Found. Comput. Math.* **19** 259–96
[101] Fehr S, Gelles R and Schaffner C 2013 Security and composability of randomness expansion from Bell inequalities *Phys. Rev.* A **87** 012335
[102] Fenchel W 1949 On conjugate convex functions *Can. J. Math.* **1** 73–77
[103] Fiacco A V and McCormick G P 1990 *Nonlinear Programming: Sequential Unconstrained Minimization Techniques* vol 4 (SIAM)
[104] Frérot I, Baccari F and Acín A 2022 Unveiling quantum entanglement in many-body systems from partial information *PRX Quantum* **3** 010342
[105] Freund R M and Mizuno S 2000 Interior point methods: current status and future directions *High Performance Optimization* (Springer) pp 441–66
[106] Fritz T, Sainz A B, Augusiak R, Brask J B, Chaves R, Leverrier A and Acín A 2013 Local orthogonality as a multipartite principle for quantum correlations *Nat. Commun.* **4** 2263
[107] Fujii J I 1992 Operator means and the relative operator entropy *Operator Theory and Complex Analysis: Workshop on Operator Theory and Complex Analysis Sapporo (Japan, June 1991)* (Springer) pp 161–72
[108] Fujii J I and Kamei E 1989 Relative operator entropy in noncommutative information theory *Math. Japon* **34** 341–8
[109] Fujii J I and Seo Y 2018 The relative operator entropy and the Karcher mean *Linear Algebr. Appl.* **542** 4–34
[110] Fujii J I and Seo Y 2022 Relative operator entropy *Operator and Norm Inequalities and Related Topics* (Springer) pp 69–95

[111] Fujisawa K, Kojima M and Nakata K 1997 Exploiting sparsity in primal-dual interior-point methods for semidefinite programming *Math. Program.* **79** 235–53
[112] Fujisawa K, Kojima M, Nakata K, and Yamashita M 1995 SDPA (SemiDefinite Programming Algorithm) user's manual-version 6.2 *Research Report* B-308 (Department of Mathematical and Computing Sciences, Tokyo Institute of Technology) p 2-12-1
[113] Fujisawa K, Kojima M, Nakata K and Yamashita M 2002 SDPA (semidefinite programming algorithm) user's manual-version 6.2. 0 *Research Reports on Mathematical and Computing Sciences Series B: Operations Research* (Department of Mathematical and Computing Sciences, Tokyo Institute of Technology)
[114] Gallego R, Brunner N, Hadley C and Acín A 2010 Device-independent tests of classical and quantum dimensions *Phys. Rev. Lett.* **105** 230501
[115] Gallier J 2019 The Schur complement and symmetric positive semidefinite (and definite) matrices (available at: www.cis.upenn.edu/~jean/, University of Pennsylvania)
[116] Gärtner B and Matousek J 2012 *Approximation Algorithms and Semidefinite Programming* (Springer Science & Business Media)
[117] Gilbert G T 1991 Positive definite matrices and Sylvester's criterion *Am. Math. Mon.* **98** 44–46
[118] Gill P E, Murray W, Saunders M A, Tomlin J A and Wright M H 1986 On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method *Math. Program.* **36** 183–209
[119] Gisin N 2009 *Bell Inequalities: Many Questions, a Few Answers* (Springer) pp 125–38
[120] Gleason A M 1975 Measures on the closed subspaces of a Hilbert space *The Logico-Algebraic Approach to Quantum Mechanics* (*Historical Evolution* vol I) (Springer) pp 123–33
[121] Goemans M X 1997 Semidefinite programming in combinatorial optimization *Math. Program.* **79** 143–61
[122] Goemans M X and Williamson D 2001 Approximation algorithms for MAX-3-CUT and other problems via complex semidefinite programming *Proc. 33rd Annual ACM Symposium on Theory of Computing* pp 443–52
[123] Goemans M X and Williamson D P 1995 Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming *J. ACM* **42** 1115–45
[124] Goldfarb D and Scheinberg K 1998 Interior point trajectories in semidefinite programming *SIAM J. Optim.* **8** 871–86
[125] Gondzio J 2012 Interior point methods 25 years later *Eur. J. Oper. Res.* **218** 587–601
[126] Gonzaga C C and Todd M J 1992 An O(nL)-iteration large-step primal-dual affine algorithm for linear programming *SIAM J. Optim.* **2** 349–59
[127] Grant M and Boyd S 2014 CVX: Matlab software for disciplined convex programming version 2.1 (available at: https://cvxr.com/cvx/citing/)
[128] Grant M, Boyd S and Ye Y 2011 CVX: Matlab software for disciplined convex programming
[129] Grötschel M, Lovász L and Schrijver A 1981 The ellipsoid method and its consequences in combinatorial optimization *Combinatorica* **1** 169–97
[130] Grötschel M, Lovász L and Schrijver A 1984 Geometric methods in combinatorial optimization *Progress in Combinatorial Optimization* (Elsevier) pp 167–83
[131] Grötschel M, Lovász L and Schrijver A 1986 Relaxations of vertex packing *J. Comb. Theory* B **40** 330–43
[132] Guimaraes D A, Floriano G H F and Chaves L S 2015 A tutorial on the CVX system for modeling and solving convex optimization problems *IEEE Latin Am. Trans.* **13** 1228–57
[133] Gupta S, Saha D, Xu Z-P, Cabello A and Majumdar A S 2023 Quantum contextuality provides communication complexity advantage *Phys. Rev. Lett.* **130** 080802
[134] Hansen F and Pedersen G K 2003 Jensen's operator inequality *Bull. London Math. Soc.* **35** 553–64
[135] Hansson A and Vandenberghe L 2014 Sampling method for semidefinite programmes with non-negative Popov function constraints *Int. J. Control* **87** 330–45
[136] Hardy L 2001 Quantum theory from five reasonable axioms (arXiv:quant-ph/0101012)
[137] Hardy L 2011 Reformulating and reconstructing quantum theory (arXiv:1104.2066)
[138] Bauschke H H and Combettes P L 2010 *Convex Analysis and Monotone Operator Theory in Hilbert Spaces (CMS Books in Mathematics)* (Springer)
[139] Heisenberg W 1927 Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik *Z. Phys.* **43** 172–98
[140] Helmberg C 2000 Semidefinite programming for combinatorial optimization *PhD Thesis* Konrad-Zuse-Zentrum fur Informationstechnik

[141] Helmberg C, Rendl F, Vanderbei R J and Wolkowicz H 1996 An interior-point method for semidefinite programming *SIAM J. Optim.* **6** 342–61
[142] Helstrom C W 1969 Quantum detection and estimation theory *J. Stat. Phys.* **1** 231–52
[143] Helton J and McCullough S 2004 A Positivstellensatz for non-commutative polynomials *Trans. Am. Math. Soc.* **356** 3721–37
[144] Helton J W 2002 "Positive" noncommutative polynomials are sums of squares *Ann. Math.* **156** 675–94
[145] Helton J W and Nie J 2009 Sufficient and necessary conditions for semidefinite representability of convex hulls and sets *SIAM J. Optim.* **20** 759–91
[146] Helton J W and Nie J 2010 Semidefinite representation of convex sets *Math. Program.* **122** 21–64
[147] Helton J W and Vinnikov V 2007 Linear matrix inequality representation of sets *Commun. Pure Appl. Math.* A **60** 654–74
[148] Henderson L 2020 Quantum reaxiomatisations and information-theoretic interpretations of quantum theory *Stud. Hist. Phil. Sci.* B **72** 292–300
[149] Hoban M J and Sainz A B 2018 A channel-based framework for steering, non-locality and beyond *New J. Phys.* **20** 053048
[150] Horn R A and Johnson C R 2012 *Matrix Analysis* (Cambridge University Press)
[151] Horodecki M, Horodecki P and Horodecki R 1996 Separability of mixed quantum states: necessary and sufficient conditions *Phys. Lett.* A **223** 1–8
[152] Howard M, Wallman J, Veitch V and Emerson J 2014 Contextuality supplies the 'magic' for quantum computation *Nature* **510** 351–5
[153] Ivanovic I D 1987 How to differentiate between non-orthogonal states *Phys. Lett.* A **123** 257–9
[154] Jameson G J O 1972 Convex series *Math. Proc. Camb. Phil. Soc.* **72** 37–47
[155] Jamiołkowski A 1972 Linear transformations which preserve trace and positive semidefiniteness of operators *Rep. Math. Phys.* **3** 275–8
[156] Janotta P and Hinrichsen H 2014 Generalized probability theories: what determines the structure of quantum theory? *J. Phys. A: Math. Theor.* **47** 323001
[157] Jarvis-Wloszek Z, Feeley R, Tan W, Sun K and Packard A 2005 Control applications of sum of squares programming *Positive Polynomials in Control* (Springer) pp 3–22
[158] Jbilou K, Messaoudi A and Tabaâ K 2004 Some Schur complement identities and applications to matrix extrapolation methods *Linear Algebr. Appl.* **392** 195–210
[159] Jeyakumar V, Lasserre J B, Li G and Pham T S 2016 Convergent semidefinite programming relaxations for global bilevel polynomial optimization problems *SIAM J. Optim.* **26** 753–80
[160] Ježek M, Řeháček J and Fiurášek J 2002 Finding optimal strategies for minimum-error quantum-state discrimination *Phys. Rev.* A **65** 060301
[161] Jiang M, Luo S and Fu S 2013 Channel-state duality *Phys. Rev.* A **87** 022310
[162] Johnston N 2016 QETLAB: A MATLAB toolbox for quantum entanglement, version 0.9 (available at: https://qetlab.com)
[163] Johnston N, Mittal R, Russo V and Watrous J 2016 Extended non-local games and monogamy-of-entanglement games *Proc. R. Soc.* A **472** 20160003
[164] Karger D, Motwani R and Sudan M 1998 Approximate graph coloring by semidefinite programming *J. ACM* **45** 246–65
[165] Karmarkar N 1984 A new polynomial-time algorithm for linear programming *Proc. 16th Annual ACM Symposium on Theory of Computing* pp 302–11
[166] Kempe J, Kobayashi H, Matsumoto K, Toner B and Vidick T 2011 Entangled games are hard to approximate *SIAM J. Comput.* **40** 848–77
[167] Kennard E H 1927 Zur quantenmechanik einfacher bewegungstypen *Z. Phys.* **44** 326–52
[168] Khachian L G 1979 A polynomial time algorithm for linear programing *Sov. Math. Dokl.* **244** 1093–6
[169] Kheirfam B 2015 An adaptive infeasible interior-point algorithm with full Nesterov-Todd step for semidefinite optimization *J. Math. Model. Algor.* **14** 55–66
[170] Klee V and Minty G J 1972 How good is the simplex algorithm *Inequalities* **3** 159–75
[171] Kleinberg J and Goemans M X 1998 The Lovász theta function and a semidefinite programming relaxation of vertex cover *SIAM J. Discrete Math.* **11** 196–204
[172] Klep I, Magron V and Povh J 2022 Sparse noncommutative polynomial optimization *Math. Program.* A+B **193** 789–829
[173] Klep I and Povh J 2010 Semidefinite programming and sums of hermitian squares of noncommutative polynomials *J. Pure Appl. Algebra* **214** 740–9

[174] Klep I and Povh J 2016 Constrained trace-optimization of polynomials in freely noncommuting variables *J. Glob. Opt.* **64** 325–48
[175] Kobayashi K, Nakata K and Kojima M 2007 A conversion of an SDP having free variables into the standard form SDP *Comput. Optim. Appl.* **36** 289–307
[176] Kochen S and Specker E P 1990 The problem of hidden variables in quantum mechanics *Ernst Specker* (Birkhäuser) pp 235–63
[177] Kogias I, Skrzypczyk P, Cavalcanti D, Acín A and Adesso G 2015 Hierarchy of steering criteria based on moments for all bipartite quantum systems *Phys. Rev. Lett.* **115** 210401
[178] Kojima M, Megiddo N and Mizuno S 1993 A primal-dual infeasible-interior-point algorithm for linear programming *Math. Program.* **61** 263–80
[179] Kojima M, Mizuno S and Yoshise A 1989 *A Primal-Dual Interior Point Algorithm for Linear Programming* (Springer)
[180] Kojima M, Shindoh S and Hara S 1997 Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices *SIAM J. Optim.* **7** 86–125
[181] Kraus F 1936 Über konvexe matrixfunktionen *Math. Z.* **41** 18–42
[182] Kubo F and Ando T 1980 Means of positive linear operators *Math. Ann.* **246** 205–24
[183] Kueng R, Long D M, Doherty A C and Flammia S T 2016 Comparing experiments to the fault-tolerance threshold *Phys. Rev. Lett.* **117** 170502
[184] Lasserre J B 2001 Global optimization with polynomials and the problem of moments *SIAM J. Optim.* **11** 796–817
[185] Lasserre J B 2007 A sum of squares approximation of nonnegative polynomials *SIAM Rev.* **49** 651–69
[186] Leifer M S 2014 Is the quantum state real? An extended review of psi-ontology theorems *Quanta* **3** 67
[187] Leung D and Matthews W 2015 On the power of PPT-preserving and non-signalling codes *IEEE Trans. Inf. Theory* **61** 4486–99
[188] Lewenstein M and Sanpera A 1998 Separability and entanglement of composite quantum systems *Phys. Rev. Lett.* **80** 2261
[189] Li H-W, Mironowicz P, Pawłowski M, Yin Z-Q, Wu Y-C, Wang S, Chen W, Hu H-G, Guo G-C and Han Z-F 2013 Relationship between semi-and fully-device-independent protocols *Phys. Rev.* A **87** 020302
[190] Li H-W, Yin Z-Q, Wu Y-C, Zou X-B, Wang S, Chen W, Guo G-C and Han Z-F 2011 Semi-device-independent random-number expansion without entanglement *Phys. Rev.* A **84** 034301
[191] Lin P-S, Vértesi T and Liang Y-C 2022 Naturally restricted subsets of nonsignaling correlations: typicality and convergence *Quantum* **6** 765
[192] Linden N, Popescu S, Short A J and Winter A 2007 Quantum nonlocality and beyond: limits from nonlocal computation *Phys. Rev. Lett.* **99** 180502
[193] Löfberg J 2004 YALMIP : a toolbox for modeling and optimization in MATLAB *Proc. CACSD Conf. (Taipei, Taiwan)*
[194] Löfberg J 2009 Dualize it: software for automatic primal and dual conversions of conic programs *Opt. Methods Softw.* **24** 313–25
[195] Lofberg J 2009 Pre-and post-processing sum-of-squares programs in practice *IEEE Trans. Autom. Control* **54** 1007–11
[196] Lovász L 1979 On the Shannon capacity of a graph *IEEE Trans. Inf. Theory* **25** 1–7
[197] Löwner K 1934 Über monotone matrixfunktionen *Math. Z.* **38** 177–216
[198] Lucchetti R 2006 *Convexity and Well-Posed Problems (CMS Books in Mathematics)* (Springer)
[199] Magesan E, Gambetta J M and Emerson J 2012 Characterizing quantum gates via randomized benchmarking *Phys. Rev.* A **85** 042311
[200] Magnus J R and Neudecker H 2019 *Matrix Differential Calculus With Applications in Statistics and Econometrics* (Wiley)
[201] Masanes L and Müller M P 2011 A derivation of quantum theory from physical requirements *New J. Phys.* **13** 063001
[202] Masanes L, Pironio S and Acín A 2011 Secure device-independent quantum key distribution with causally independent measurement devices *Nat. Commun.* **2** 238
[203] Mayers D and Yao A 1998 Quantum cryptography with imperfect apparatus *Proc. 39th Annual Symp. on Foundations of Computer Science (Cat. No. 98CB36280)* (IEEE) pp 503–9
[204] McCullough S and Putinar M 2005 Noncommutative sums of squares *Pac. J. Math.* **218** 167–71

[205] Mehrotra S 1992 On the implementation of a primal-dual interior point method *SIAM J. Optim.* **2** 575–601
[206] Mercer J 1909 Functions of positive and negative type and their connection with the theory of integral equations *Phil. Trans. R. Soc.* A **209** 415–46
[207] Merkel S T, Gambetta J M, Smolin J A, Poletto S, Córcoles A D, Johnson B R, Ryan C A and Steffen M 2013 Self-consistent quantum process tomography *Phys. Rev.* A **87** 062119
[208] Mészáros C 1998 On free variables in interior point methods *Opt. Methods Softw.* **9** 121–39
[209] Meyer C D 2000 *Matrix Analysis and Applied Linear Algebra* vol 71 (SIAM)
[210] Miltenberger M 2021 Mittelmann-plots—interactive visualizations of Mittelmann benchmarks (available at: https://mattmilten.github.io/mittelmann-plots/)
[211] Mironowicz P 2015 Applications of semi-definite optimization in quantum information protocols *PhD Thesis* Gdańsk University of Technology
[212] Mironowicz P, Li H-W and Pawłowski M 2014 Properties of dimension witnesses and their semi-definite programming relaxations *Phys. Rev.* A **90** 022322
[213] Mittelmann H 2023 Decision tree for optimization software (available at: https://plato.asu.edu/guide.html)
[214] Mittelmann H D 2012 The state-of-the-art in conic optimization software *Handbook on Semidefinite, Conic and Polynomial Optimization* ed M Anjos and J Lasserre (Springer) pp 671–86
[215] Mohar B and Poljak S 1993 *Eigenvalues in Combinatorial Optimization, in Combinatorial and Graph-Theoretical Problems in Linear Algebra* (Springer) pp 107–51
[216] Monteiro R D C 1997 Primal–dual path-following algorithms for semidefinite programming *SIAM J. Optim.* **7** 663–78
[217] Monteiro R D C 1998 Polynomial convergence of primal-dual algorithms for semidefinite programming based on the Monteiro and Zhang family of directions *SIAM J. Optim.* **8** 797–812
[218] Monteiro R D C and Adler I 1989 Interior path following primal-dual algorithms. Part I: Linear programming *Math. Program.* **44** 27–41
[219] Monteiro R D C and Adler I 1989 Interior path following primal-dual algorithms. Part II: convex quadratic programming *Math. Program.* **44** 43–66
[220] Monteiro R D C and Zhang Y 1998 A unified analysis for a class of long-step primal-dual path-following interior-point algorithms for semidefinite programming *Math. Program.* **81** 281–99
[221] Moroder T, Bancal J-D, Liang Y-C, Hofmann M and Gühne O 2013 Device-independent entanglement quantification and related applications *Phys. Rev. Lett.* **111** 030501
[222] Nakata M 2010 A numerical evaluation of highly accurate multiple-precision arithmetic version of semidefinite programming solver: SDPA-GMP,-QD and-DD *2010 IEEE Int. Symp. on Computer-Aided Control System Design* (IEEE) pp 29–34
[223] Navascués M, de la Torre G and Vértesi T 2014 Characterization of quantum correlations with local dimension constraints and its device-independent applications *Phys. Rev.* X **4** 011011
[224] Navascués M, Feix A, Araújo M and Vértesi T 2015 Characterizing finite-dimensional quantum behavior *Phys. Rev.* A **92** 042117
[225] Navascués M, Guryanova Y, Hoban M J and Acín A 2015 Almost quantum correlations *Nat. Commun.* **6** 6288
[226] Navascués M, Pironio S and Acín A 2007 Bounding the set of quantum correlations *Phys. Rev. Lett.* **98** 010401
[227] Navascués M, Pironio S and Acín A 2008 A convergent hierarchy of semidefinite programs characterizing the set of quantum correlations *New J. Phys.* **10** 073013
[228] Navascués M and Vértesi T 2015 Bounding the set of finite dimensional quantum correlations *Phys. Rev. Lett.* **115** 020501
[229] Navascués M and Wunderlich H 2010 A glance beyond the quantum model *Proc. R. Soc.* A **466** 881–90
[230] Nemirovski A 2004 Interior point polynomial time methods in convex programming *Lecture Notes* vol 42 pp 3215–24 (available at: https://www2.isye.gatech.edu/~nemirovs/)
[231] Nemirovski A 2007 Advances in convex optimization: conic programming *Int. Congress of Mathematicians* vol 1 pp 413–44
[232] Nesterov Y 2018 Lectures on convex optimization *Springer Optimization and Its Applications* vol 137, 2nd edn (Springer)
[233] Nesterov Y and Nemirovskii A 1994 *Interior-Point Polynomial Algorithms in Convex Programming* (SIAM)

[234] Nesterov Y and Nemirovsky A 1992 Conic formulation of a convex programming problem and duality *Opt. Methods Softw.* **1** 95–115
[235] Nesterov Y E and Todd M J 1997 Self-scaled barriers and interior-point methods for convex programming *Math. Oper. Res.* **22** 1–42
[236] Nesterov Y E and Todd M J 1998 Primal-dual interior-point methods for self-scaled cones *SIAM J. Optim.* **8** 324–64
[237] Netzer T 2010 On semidefinite representations of non-closed sets *Linear Algebr. Appl.* **432** 3072–8
[238] Netzer T and Sinn R 2009 A note on the convex hull of finitely many projections of spectrahedra (arXiv:0908.3386)
[239] Nieto-Silleras O, Pironio S and Silman J 2014 Using complete measurement statistics for optimal device-independent randomness evaluation *New J. Phys.* **16** 013035
[240] Overton M L 1992 Large-scale optimization of eigenvalues *SIAM J. Optim.* **2** 88–120
[241] O'donoghue B, Chu E, Parikh N and Boyd S 2016 Conic optimization via operator splitting and homogeneous self-dual embedding *J. Optim. Theory Appl.* **169** 1042–68
[242] Pál K F and Vértesi T 2010 Maximal violation of a bipartite three-setting, two-outcome Bell inequality using infinite-dimensional quantum systems *Phys. Rev.* A **82** 022116
[243] Papachristodoulou A, Anderson J, Valmorbida G, Prajna S, Seiler P Parrilo P, Peet M M and Jagt D 2013 SOSTOOLS version 4.00 sum of squares optimization toolbox for MATLAB (arXiv:1310.4716)
[244] Papachristodoulou A and Prajna S 2005 A tutorial on sum of squares techniques for systems analysis *Proc. 2005, American Control Conf., 2005* (IEEE) pp 2686–700
[245] Parrilo P A 2003 Semidefinite programming relaxations for semialgebraic problems *Math. Program.* **96** 293–320
[246] Parrilo P A and Jadbabaie A 2008 Approximation of the joint spectral radius using sum of squares *Linear Algebr. Appl.* **428** 2385–402
[247] Parrilo P A and Sturmfels B 2003 Minimizing polynomial functions *Algorithmic and Quantitative Real Algebraic Geometry (DIMACS Series in Discrete Mathematics and Theoretical Computer Science vol 60) (American Mathematical Society)* pp 83–99 (available at: https://bookstore.ams.org/dimacs-60)
[248] Pawłowski M and Brunner N 2011 Semi-device-independent security of one-way quantum key distribution *Phys. Rev.* A **84** 010302
[249] Pawłowski M, Paterek T, Kaszlikowski D, Scarani V, Winter A and Żukowski M 2009 Information causality as a physical principle *Nature* **461** 1101–4
[250] Peres A 1996 Separability criterion for density matrices *Phys. Rev. Lett.* **77** 1413
[251] Piani M, Cianciaruso M, Bromley T R, Napoli C, Johnston N and Adesso G 2016 Robustness of asymmetry and coherence of quantum states *Phys. Rev.* A **93** 042107
[252] Pironio S, Acín A, Massar S, de La Giroday A B, Matsukevich D N, Maunz P, Olmschenk S, Hayes D, Luo L and Manning T A 2010 Random numbers certified by Bell's theorem *Nature* **464** 1021–4
[253] Pironio S and Massar S 2013 Security of practical private randomness generation *Phys. Rev.* A **87** 012336
[254] Pironio S, Navascués M and Acin A 2010 Convergent relaxations of polynomial optimization problems with noncommuting variables *SIAM J. Optim.* **20** 2157–80
[255] Popescu S and Rohrlich D 1994 Quantum nonlocality as an axiom *Found. Phys.* **24** 379–85
[256] Potra F A and Wright S J 2000 Interior-point methods *J. Comput. Appl. Math.* **124** 281–302
[257] Pozas-Kerstjens A, Rabelo R, Rudnicki Ł, Chaves R, Cavalcanti D, Navascués M and Acín A 2019 Bounding the sets of classical and quantum correlations in networks *Phys. Rev. Lett.* **123** 140503
[258] Prajna S, Papachristodoulou A and Parrilo P A 2002 Introducing SOSTOOLS: a general purpose sum of squares programming solver *Proc. 41st IEEE Conf. on Decision and Control, 2002* vol 1 (IEEE) pp 741–6
[259] Primaatmaja I W, Ho A and Scarani V 2021 Optimal single-shot discrimination of optical modes *Phys. Rev.* A **103** 052410
[260] Pusey M F 2013 Negativity and steering: a stronger Peres conjecture *Phys. Rev.* A **88** 032313
[261] Pusz W and Woronowicz S L 1975 Functional calculus for sesquilinear forms and the purification map *Rep. Math. Phys.* **8** 159–70
[262] Pyatnitskiy Y S and Skorodinskiy V 1982 Numerical methods of Lyapunov function construction and their application to the absolute stability problem *Syst. Control Lett.* **2** 130–5

[263] Quarteroni A, Sacco R and Saleri F 2007 Foundations of matrix analysis *Numerical Mathematics* (Springer) pp 1–32
[264] Raggio G A and Werner R F 1988 Quantum statistical mechanics of general mean field systems *Helv. Phys. Acta* **62** 980
[265] Rains E M 1999 Bound on distillable entanglement *Phys. Rev.* A **60** 179
[266] Rains E M 2001 A semidefinite program for distillable entanglement *IEEE Trans. Inf. Theory* **47** 2921–33
[267] Ramana M and Goldman A J 1995 Some geometric results in semidefinite programming *J. Glob. Opt.* **7** 33–50
[268] Regula B and Takagi R 2021 Fundamental limitations on distillation of quantum channel resources *Nat. Commun.* **12** 4411
[269] Renou M-O and Xu X 2022 Two convergent NPA-like hierarchies for the quantum bilocal scenario (arXiv:2210.09065)
[270] Robertson H P 1929 The uncertainty principle *Phys. Rev.* **34** 163
[271] Rockafellar R T and Wets R J-B R 2009 *Variational Analysis (Grundlehren der Mathematischen Wissenschaften)* (Springer)
[272] Rosset D QDimSum: Symmetric SDP relaxations for qudits systems (available at: https://github.com/denisrosset/qdimsum)
[273] Sadiq M, Badziąg P, Bourennane M and Cabello A 2013 Bell inequalities for the simplest exclusivity graph *Phys. Rev.* A **87** 012128
[274] Sagnol G 2013 On the semidefinite representation of real functions applied to symmetric matrices *Linear Algebr. Appl.* **439** 2829–43
[275] Sagnol G and Stahlberg M 2022 PICOS: a Python interface to conic optimization solvers *J. Open Source Softw.* **7** 3915
[276] Sainz A B, Aolita L, Piani M, Hoban M J and Skrzypczyk P 2018 A formalism for steering with local quantum measurements *New J. Phys.* **20** 083040
[277] Sainz A B, Brunner N, Cavalcanti D, Skrzypczyk P and Vértesi T 2015 Postquantum steering *Phys. Rev. Lett.* **115** 190403
[278] Scheiderer C 2009 Positivity and sums of squares: a guide to recent results *Emerging Applications of Algebraic Geometry* (Springer) pp 271–324
[279] Scheiderer C 2018 Semidefinite representation for convex hulls of real algebraic curves *SIAM J. Appl. Algebra Geom.* **2** 1–25
[280] Scheiderer C 2018 Spectrahedral shadows *SIAM J. Appl. Algebra Geom.* **2** 26–44
[281] Schur J 1917 Über Potenzreihen, die im Innern des Einheitskreises beschränkt sind *J. Angew. Math.* **147** 205–32
[282] Seidenberg A 1954 A new decision method for elementary algebra *Ann. Math.* **60** 365–74
[283] Shannon C 1956 The zero error capacity of a noisy channel *IRE Trans. Inf. Theory* **2** 8–19
[284] Skrzypczyk P and Cavalcanti D 2023 *Semidefinite Programming in Quantum Information Science* (IOP Publishing) (available at: https://doi.org/10.1088/978-0-7503-3343-6)
[285] Slater M 2013 Lagrange multipliers revisited *Traces and Emergence of Nonlinear Programming* (Springer) pp 293–306
[286] Smania M, Mironowicz P, Nawareg M, Pawłowski M, Cabello A and Bourennane M 2020 Experimental certification of an informationally complete quantum measurement in a device-independent protocol *Optica* **7** 123–8
[287] Spekkens R W 2005 Contextuality for preparations, transformations and unsharp measurements *Phys. Rev.* A **71** 052108
[288] Stein E M and Shakarchi R 2009 *Real Analysis: Measure Theory, Integration and Hilbert Spaces* (Princeton University Press)
[289] Stinespring W F 1955 Positive functions on C star-algebras *Proc. Am. Math. Soc.* **6** 211–6
[290] Sturm J F SQLP/SeDuMi: SeDuMi: a linear/quadratic/semidefinite solver for MATLAB and octave (available at: https://github.com/sqlp/sedumi)
[291] Sturm J F 1999 Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones *Opt. Methods Softw.* **11** 625–53
[292] Sturm J F 2002 Implementation of interior point methods for mixed semidefinite and second order cone optimization problems *Opt. Methods Softw.* **17** 1105–54
[293] Sturm J F and Zhang S 1999 Symmetric primal-dual path-following algorithms for semidefinite programming *Appl. Numer. Math.* **29** 301–15

[294] Sun D, Toh K-C, Yuan Y and Zhao X-Y 2020 SDPNAL+: a matlab software for semidefinite programming with bound constraints (version 1.0) *Opt. Methods Softw.* **35** 87–115
[295] Šupić I and Bowles J 2020 Self-testing of quantum systems: a review *Quantum* **4** 337
[296] Sutter D, Scholz V B, Winter A and Renner R 2017 Approximate degradable quantum channels *IEEE Trans. Inf. Theory* **63** 7832–44
[297] Tarski A 1949 A decision method for elementary algebra and geometry *J. Symb. Log.* **14** 188–188
[298] Tavakoli A, Cruzeiro E Z, Woodhead E and Pironio S 2022 Informationally restricted correlations: a general framework for classical and quantum systems *Quantum* **6** 620
[299] Tavakoli A, Kaniewski J, Vértesi T, Rosset D and Brunner N 2018 Self-testing quantum states and measurements in the prepare-and-measure scenario *Phys. Rev.* A **98** 062307
[300] Tavakoli A, Pozas-Kerstjens A, Brown P and Araújo M 2023 Semidefinite programming relaxations for quantum correlations (arXiv:2307.02551)
[301] Tavakoli A, Pozas-Kerstjens A, Luo M-X and Renou M-O 2022 Bell nonlocality in networks *Rep. Prog. Phys.* **85** 056001
[302] Tavakoli A, Rosset D and Renou M-O 2019 Enabling computation of correlation bounds for finite-dimensional quantum systems via symmetrization *Phys. Rev. Lett.* **122** 070501
[303] Terlaky T 2013 *Interior Point Methods of Mathematical Programming* vol 5 (Springer Science & Business Media)
[304] Tilly J *et al* 2022 The variational quantum eigensolver: a review of methods and best practices *Phys. Rep.* **986** 1–128
[305] Todd M J 1999 A study of search directions in primal-dual interior-point methods for semidefinite programming *Opt. Methods Softw.* **11** 1–46
[306] Todd M J 2001 Semidefinite optimization *Acta Numer.* **10** 515–60
[307] Todd M J, Toh K-C and Tütüncü R H 1998 On the Nesterov–Todd direction in semidefinite programming *SIAM J. Optim.* **8** 769–96
[308] Toh K-C 2002 A note on the calculation of step-lengths in interior-point methods for semidefinite programming *Comput. Optim. Appl.* **21** 301–10
[309] Toh K-C, Todd M J and Tütüncü R H 1999 SDPT3-a MATLAB software package for semidefinite programming, version 1.3 *Opt. Methods Softw.* **11** 545–81
[310] Toh K-C, Todd M J and Tütüncü R H 2012 On the implementation and usage of SDPT3–a Matlab software package for semidefinite-quadratic-linear programming, version 4.0 *Handbook on Semidefinite, Conic and Polynomial Optimization* (Springer) pp 715–54
[311] Tomamichel M 2015 *Quantum Information Processing With Finite Resources: Mathematical Foundations* vol 5 (Springer)
[312] Tütüncü R H, Toh K C and Todd M J 2003 Solving semidefinite-quadratic-linear programs using SDPT3 *Math. Program.* **95** 189–217
[313] v. Neumann J 1928 Zur theorie der gesellschaftsspiele *Math. Ann.* **100** 295–320
[314] Vandenberghe L and Boyd S 1995 A primal-dual potential reduction method for problems involving matrix inequalities *Math. Program.* **69** 205–36
[315] Vandenberghe L and Boyd S 1996 *SIAM Rev.* **38** 49–95
[316] Vazirani V V 2013 *Approximation Algorithms* (Springer Science & Business Media)
[317] Vinnikov V 2012 LMI representations of convex semialgebraic sets and determinantal representations of algebraic hypersurfaces: past, present and future *Mathematical Methods in Systems, Optimization, and Control* (Festschrift in Honor of J William Helton) pp 325–49
[318] Waki H, Kim S, Kojima M and Muramatsu M 2006 Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity *SIAM J. Optim.* **17** 218–42
[319] Wang G Q and Bai Y Q 2009 A new primal-dual path-following interior-point algorithm for semidefinite optimization *J. Math. Anal. Appl.* **353** 339–49
[320] Wang J and Magron V 2021 Exploiting term sparsity in noncommutative polynomial optimization *Comput. Optim. Appl.* **80** 483–521
[321] Wang J *et al* 2018 Multidimensional quantum entanglement with large-scale integrated optics *Science* **360** 285–91
[322] Wang X and Duan R 2016 A semidefinite programming upper bound of quantum capacity *2016 IEEE Int. Symp. on Information Theory (ISIT)* (IEEE) pp 1690–4
[323] Watrous J 2009 Semidefinite programs for completely bounded norms *Theory of Computing* **5** 217–38

[324] Watrous J 2011 CS 867/QIC 890 semidefinite programming in quantum information *Lecture Notes* (available at: https://johnwatrous.com/lecture-notes/)
[325] Watrous J 2013 Simpler semidefinite programs for completely bounded norms *Chicago J. Theor. Comput. Sci.* **2013** 1–19 (available at: http://cjtcs.cs.uchicago.edu/articles/2013/contents.html)
[326] Watrous J 2018 *The Theory of Quantum Information* (Cambridge University Press)
[327] Werner R F 1989 Quantum states with Einstein-Podolsky-Rosen correlations admitting a hidden-variable model *Phys. Rev.* A **40** 4277
[328] Werner R F and Wolf M M 2001 Bell inequalities and entanglement *Quantum Inf. Comput.* **1** 1–25
[329] Wilde M M 2017 *Quantum Information Theory* 2nd edn (Cambridge University Press)
[330] Wilde M M 2020 Coherent quantum channel discrimination *2020 IEEE Int. Symp. on Information Theory (ISIT)* (IEEE) pp 1915–20
[331] Wiseman H M, Jones S J and Doherty A C 2007 Steering, entanglement, nonlocality and the Einstein-Podolsky-Rosen paradox *Phys. Rev. Lett.* **98** 140402
[332] Wittek P 2015 Algorithm 950: Ncpol2sdpa-sparse semidefinite programming relaxations for polynomial optimization problems of noncommuting variables *ACM Trans. Math. Softw.* **41** 1–12
[333] Wittek P and Brown P J Updating ncpol2sdpa after Peter Wittek (available at: https://github.com/peterjbrown519/ncpol2sdpa)
[334] Wright M 2005 The interior-point revolution in optimization: history, recent developments and lasting consequences *Bull. Am. Math. Soc* **42** 39–56
[335] Yamashita M, Fujisawa K, Fukuda M, Kobayashi K, Nakata K and Nakata M 2012 Latest developments in the SDPA family for solving large-scale SDPs *Handbook on Semidefinite, Conic and Polynomial Optimization* pp 687–713
[336] Yamashita M, Fujisawa K and Kojima M 2003 Implementation and evaluation of SDPA 6.0 (semidefinite programming algorithm 6.0 *Opt. Methods Softw.* **18** 491–505
[337] Yamashita M, Fujisawa K, Nakata K, Nakata M, Fukuda M, Kobayashi K and Goto K 2010 *A High-Performance Software Package for Semidefinite Programs: SDPA 7* (Tokyo, Japan)
[338] Yang L, Sun D and Toh K-C 2015 SDPNAL++: a majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints *Math. Program. Comput.* **7** 331–66
[339] Yang T H and Navascués M 2013 Robust self-testing of unknown quantum systems into any entangled two-qubit states *Phys. Rev.* A **87** 050102
[340] Yang T H, Vértesi T, Bancal J-D, Scarani V and Navascués M 2014 Robust and versatile black-box certification of quantum devices *Phys. Rev. Lett.* **113** 040401
[341] Yu X-D, Simnacher T, Wyderka N, Nguyen H C and Gühne O 2021 A complete hierarchy for the pure state marginal problem in quantum mechanics *Nat. Commun.* **12** 1–7
[342] Zalinescu C 2002 *Convex Analysis in General Vector Spaces* (World Scientific)
[343] Zhang F 2006 *The Schur Complement and its Applications* vol 4 (Springer Science & Business Media)
[344] Zhang Y 1998 On extending some primal–dual interior-point algorithms from linear programming to semidefinite programming *SIAM J. Optim.* **8** 365–86
[345] Zhang Y, Tapia R A and Dennis Jr J E 1992 On the superlinear and quadratic convergence of primal-dual interior point linear programming algorithms *SIAM J. Optim.* **2** 304–24
[346] Zhao X-Y, Sun D and Toh K-C 2010 A Newton-CG augmented Lagrangian method for semidefinite programming *SIAM J. Optim.* **20** 1737–65