Conic Optimization for Control, Energy Systems, and Machine Learning: Applications and Algorithms[☆,☆☆]

Richard Y. Zhang[a,1], Cédric Josz[b,1], Somayeh Sojoudi[b,*]

[a] *Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA 94720, United States of America*
[b] *Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720, United States of America*

Abstract

Optimization is at the core of control theory and appears in several areas of this field, such as optimal control, distributed control, system identification, robust control, state estimation, model predictive control and dynamic programming. The recent advances in various topics of modern optimization have also been revamping the area of machine learning. Motivated by the crucial role of optimization theory in the design, analysis, control and operation of real-world systems, this tutorial paper offers a detailed overview of some major advances in this area, namely conic optimization and its emerging applications. First, we discuss the importance of conic optimization in different areas. Then, we explain seminal results on the design of hierarchies of convex relaxations for a wide range of nonconvex problems. Finally, we study different numerical algorithms for large-scale conic optimization problems.

1. Introduction

Optimization is an important tool in the design, analysis, control and operation of real-world systems. In its purest form, optimization is the mathematical problem of minimizing (or maximizing) an *objective function* by selecting the best choice of *decision variables*, possibly subject to *constraints* on their specific values. In the wider engineering context, optimization also encompasses the process of identifying the most suitable objective, variables, and constraints. The goal is to select a mathematical model that gives useful insight to the practical problem at hand, and to design a robust and scalable algorithm that finds a provably optimal (or near-optimal) solution in a reasonable amount of time.

The theory of *convex* optimization had a profound influence on the development of modern control theory, giving rise to the ideas of robust and optimal control [1–3], distributed control [4], system identification [5], model predictive control [6] and dynamic programming [7]. The mathematical study of convex optimization dates by more than a century, but its usefulness for practical applications only came to light during the 1990s, when it was discovered that many important engineering problems are actually convex (or can be well-approximated as being so) [8]. Convexity is crucial in this regard, because it allows the corresponding optimization problem to be solved—very reliably and efficiently—to global optimality, using only local search algorithms. Moreover, this global optimality can be certified by solving a dual convex optimization problem.

With the recent advances in computing and sensor technologies, convex optimization has also become the backbone of data science and machine learning, where it supplies us the techniques to extract useful information from data [9–11]. This tutorial paper is, in part, inspired by the crucial role of optimization theory in both the long-standing area of control systems and the newer area of machine learning, as well as its multi-billion applications in large-scale real-world systems such as power grids.

Nevertheless, most interesting optimization problems are nonconvex: structured analysis and synthesis and output feedback control in control theory [2, 12]; deep learning, Bayesian inference, and clustering in machine learning [4, 9, 13, 14]; and integer and mixed integer programs in operations research [15–17]. Nonconvex problems are difficult to solve, both in theory and in practice. While candidate solutions are easy to find, locally optimal solutions are not necessarily globally optimal. Even if a candidate solution is suspected of being globally optimal, there is often no effective way of validating the hypothesis.

Convex optimization can rigorously solve nonconvex problems to global optimality, using techniques collectively known as *convexification*. The essential idea is to relax a nonconvex problem into a hierarchy of convex problems that monotonously converge towards the global solution [18–21]. These resulting convex problems come in a standard form known as *conic optimization*, which generalizes semidefinite programs (SDP) and linear matrix inequalities (LMI) from control theory, as well as quad-

[☆] The first two authors contributed equally to this work.
[☆☆] This work was supported by the ONR Award N00014-18-1-2526 and an NSF EPCN Grant.
[*] Corresponding author
Email addresses: ryz@berkeley.edu (Richard Y. Zhang), cedric.josz@berkeley.edu (Cédric Josz), sojoudi@berkeley.edu (Somayeh Sojoudi)

dratically-constrained quadratic programs (QCQPs) from statistics, and linear programs (LPs) from operations research. Conic optimization can be solved with reliability using interior-point methods [22, 23], and also with great efficiency using large-scale numerical algorithms designed to exploit problem structure [24–32].

The remainder of this paper is organized as follows. We begin in Section 2 by reviewing well-known conic optimization problems in control theory. Case studies are provided in Section 3 to show the importance of conic optimization in emerging applications. Different convexification techniques for polynomial optimization are studied in Section 4, followed by a detailed investigation of numerical algorithms for conic optimization in Section 5. Concluding remarks are drawn in Section 6.

Notations: The symbols $\mathbb{R}$ and $\mathbb{S}^n$ denote the sets of real numbers and $n \times n$ real symmetric matrices, respectively. The symbols $\mathbb{R}_+$ and $\mathbb{S}^n_+$ denote the sets of nonnegative real numbers and $n \times n$ symmetric and positive-semidefinite matrices, respectively. $\text{rank}\{\cdot\}$ and $\text{trace}\{\cdot\}$ denote the rank and trace of a matrix. The notation $X \succeq 0$ means that $X \in \mathbb{S}^n_+$. X^{opt} shows a global solution of a given conic optimization problem with the variable X. The vectorization of a matrix is the column-stacking operation

$$\text{vec}\, X = [X_{1,1}, \ldots, X_{n,1}, X_{1,2}, \ldots, X_{n,2}, \ldots, X_{n,n}]^T,$$

and the Kronecker product

$$A \otimes B = \begin{bmatrix} A_{1,1}B & \cdots & A_{1,n}B \\ \vdots & \ddots & \vdots \\ A_{n,1}B & \cdots & A_{n,n}B \end{bmatrix},$$

is defined to satisfy the Kronecker identity

$$\text{vec}\,(AXB) = (B^T \otimes A)\text{vec}\, X.$$

2. Conic Optimization in Control Theory

Before the 1990s, much of control theory was focused around developing *analytical* solutions to specific control problems. The development of conic optimization brought about a paradigm shift in the field of control theory. Modern control theory is centered around reformulating a given control problem into a conic optimization problem, with the goal of deriving a *numerical* solution. The advantage of the latter approach is that it is much more general, and is better able to accommodate constraints that appear in practical problems.

Below, we review two classical control problems where the numerical approach provides a natural generalization to the analytical approach. For a more detailed and extensive review of conic optimization in control applications, the reader is referred to classic texts [1, 2].

2.1. Stability Analysis

The discrete-time linear state-space model

$$x_{k+1} = Ax_k$$

with fixed $A \in \mathbb{R}^{n\times n}$ matrix is said to be (asymptotically) stable if, starting from any initial condition x_0, the state x_k converges to zero as time goes to infinity $k \to \infty$. Stability is readily established by examining the eigenvalues of A and verifying that each of them has modulus strictly less than one

$$|\lambda_i(A)| < 1 \qquad \forall i \in \{1, 2, \ldots, n\}.$$

Equivalently via the classical results of Lypunov, the model is stable if and only if there exists a positive definite matrix $P \succ 0$ satisfying

$$APA^T - P \prec 0.$$

The matrix P is known as a *quadratic Lyapunov function*. In this simple case, an explicit choice can be computed by solving the discrete Lyapunov equation.

Now, suppose that the state-space model is allowed to vary with time, as in

$$x_{k+1} = A_k x_k.$$

With a time-varying A_k matrix, asymptotical stability can no longer be established via an analytical approach like examining the eigenvalues of individual matrices. However, we can numerically verify stability, by attempting to find a quadratic Lyapunov function P satisfying

$$\begin{aligned} P &\succ 0 \\ A_k P A_k^T - P &\prec 0 \qquad \forall k \in \{1, 2, 3, \ldots, \}. \end{aligned}$$

If a valid choice of P can be found, then the time-varying model is stable. Indeed, the search for P is a conic feasibility problem, and can be solved to arbitrary accuracy in polynomial time using an interior-point method. However, in the time-varying context, the quadratic Lypunov test is conservative, meaning that the model can still be stable even if valid choice of P does not exist.

More sophisticated stability tests, ranging from parameter-dependent Lyapunov functions to matrix sum-of-squares, are less conservative than the stability test based on quadratic Lyapunov functions described above [33–35]. On the other hand, the reduction in conservatism usually comes with increases to computational cost.

2.2. Optimal Control

Given a discrete-time linear state-space model

$$x_{k+1} = Ax_k + Bu_k$$

with fixed $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ matrices, the *linear quadratic* (LQ) optimal control problem seeks the optimal

control signal $\{u_0, u_1, \ldots, u_T\}$ that minimizes the quadratic objective functional

$$J = \frac{1}{2}\sum_{k=0}^{T}(x_k^T Q x_k + u_k^T R u_k)$$

starting from a fixed initial condition $x_0 = c$, where $Q, R \succ 0$. The problem has a well-known analytical solution known as the *linear quadratic regulator* (LQR). This is a time-varying linear feedback law

$$u_k = -K_k x_k \qquad k \in \{1, 2, \ldots, T\}$$

with the gain matrix

$$K_k = (R + B^T P_k B)^{-1}(B^T P_{k+1} A)$$

determined by the Riccati difference equation

$$\begin{aligned} P_T &= Q, \\ P_{k-1} &= A^T P_k A - (A^T P_k B)(R + B^T P_k B)^{-1}(B^T P_k A). \end{aligned}$$

As the control horizon approaches infinity $T \to \infty$, the Riccati difference equation reduces to the *algebraic* Riccati equation, and the optimal control signal reduces to a static linear feedback law.

In practice, the basic LQ problem is often too simple to adequately capture the practical considerations faced by the controls engineer. For example, it is usually necessary to impose bound constraints on the state variables and the control signal, as in

$$u_{\mathrm{lb}} \le u_k \le u_{\mathrm{ub}}, \qquad x_{\mathrm{lb}} \le x_k \le x_{\mathrm{ub}},$$

in order to keep the physical plant and controller within their "linear range". Also, we may wish to minimize a time-varying quadratic functional of the form

$$J = \frac{1}{2}\sum_{k=0}^{T}(x_k^T Q_k x_k + u_k^T R_k u_k),$$

in order to emphasize certain time periods over others, or to trade off the transient response against asymptotic behavior. This "realistic" version of the LQ problem does not have an analytic solution. However, it is a convex quadratic program, and can be converted into a second-order cone program

$$\begin{array}{ll} \text{minimize} & \beta \\ \text{subject to} & x_0 = c \\ & x_{k+1} = Ax_k + Bu_k, \qquad \forall k \in \{0, 1, \ldots, T\} \\ & x_{\mathrm{lb}} \le x_k \le x_{\mathrm{ub}}, \qquad \forall k \in \{0, 1, \ldots, T\} \\ & u_{\mathrm{lb}} \le u_k \le u_{\mathrm{ub}}, \qquad \forall k \in \{0, 1, \ldots, T\} \\ & \beta \ge \left\| \begin{bmatrix} Q_0^{1/2} x_0 & Q_1^{1/2} x_1 & \cdots & Q_T^{1/2} x_T \\ R_0^{1/2} u_0 & R_1^{1/2} u_1 & \cdots & R_T^{1/2} u_T \end{bmatrix} \right\|_F \end{array}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, and solved to arbitrary accuracy in polynomial time using an interior-point method. (Modern interior-point methods solve convex quadratic programs by converting them into second-order cone programs, in order to exploit the self-scaled property of second-order cones [23, 36]; see also [37].) The resulting control signal can then be realized in a model-predictive controller.

3. Emerging Applications of Conic Optimization

Although conic optimization has appeared in many subareas of control theory since early 1990s, it has found new applications in many other problems in the last decade. Some of these applications will be discussed below.

3.1. Machine Learning

In machine learning, kernel methods are used to study relations, such as principle components, clusters, rankings, and correlations, in large datasets [9]. They are used for pattern recognition and analysis. Support vector machines (SVMs) are kernel-based supervised learning techniques. SVMs are one of the most popular classifiers that are currently used. A kernel matrix is a symmetric and positive semidefinite matrix that plays a key role in kernel-based learning problems. Specifying this matrix is a classical model selection problem in machine learning. A great deal of effort has been made over the past two decades to elucidate the role of semidefinite programing as an efficient convex optimization framework for machine learning, kernel-machines, SVMs, and learning kernel matrices from data in particular [9, 13, 14, 38–40].

In many applications, such as social networks, neuroscience, and financial markets, there exists a massive amount of multivariate timestamped observations. Such data can often be modeled as a network of interacting components, in which every component in the network is a node associated with some time series data. The goal is to infer the relationships between the network entities using observational data. Learning and inference are important topics in machine learning. Learning a hidden structure in data is usually formulated as an optimization problem augmented with sparsity-promoting techniques [41–43]. These techniques have become essential to the tractability of big-data analyses in many applications, such as data mining [44–46], pattern recognition [47, 48], human brain functional connectivity [49], and compressive sensing [11, 50]. Similar approaches have been used to arrive at a parsimonious estimation of high-dimensional data. However, most of the existing statistical learning techniques in data analytics need a large amount of data (compared to the number of parameters), which limit their applications in practice [10, 51]. To address this issue, a special attention has been paid to the augmentation of these learning problems with sparsity-inducing penalty functions to obtain sparse and easy-to-analyze solutions.

Graphical lasso (GL) is a popular method for estimating the inverse covariance matrix [52–54]. GL is an optimization problem that shrinks the elements of the inverse covariance matrix towards zero compared to the maximum likelihood estimates, using an l_1 regularization. There is a large body of literature suggesting that the solution of GL is a good estimate for the unknown graphical model, under a suitable choice of the regularization parameter [52–57]. To explain the GL problem, consider a random vector $x = (x_1, x_2, ..., x_d)$ with a multivariate normal distribution. Let $\Sigma_* \in \mathbb{S}^d$ denote the correlation matrix associated with the vector x. The inverse of the correlation matrix can be used to determine the conditional independence between the random variables $x_1, x_2, ..., x_d$. In particular, if the $(i,j)^{\text{th}}$ entry of Σ_*^{-1} is zero for two disparate indices i and j, then x_i and x_j are conditionally independent given the rest of the variables. The graph $\text{supp}(\Sigma_*^{-1})$ (i.e., the sparsity graph of Σ_*^{-1}) represents a graphical model capturing the conditional independence between the elements of $\mathbf{x}$.

Assume that Σ_* is nonsingular and that $\text{supp}(\Sigma_*^{-1})$ is a sparse graph. Finding this graph is cumbersome in practice because the exact correlation matrix Σ_* is rarely known. More precisely, $\text{supp}(\Sigma_*^{-1})$ should be constructed from a given sample correlation matrix (constructed from n samples), as opposed to Σ_*. Let Σ denote an arbitrary $d \times d$ positive-semidefinite matrix, which is provided as an estimate of Σ_*. Consider the convex optimization problem:

$$\begin{aligned} \min_{S \in \mathbb{S}^d} \quad & -\log\det(S) + \text{trace}(\Sigma S) \\ \text{s.t.} \quad & S \succeq 0 \end{aligned} \tag{1}$$

It is easy to verify that the optimal solution of the above problem is equal to $S^{\text{opt}} = \Sigma^{-1}$. However, there are two issues with this solution. First, since the number of samples available in many applications is modest compared to the dimension of Σ, the matrix Σ could be ill-conditioned or even singular. In that case, the equation $S^{\text{opt}} = \Sigma^{-1}$ leads to large or unbounded entries for the optimal solution of (1). Second, although Σ_*^{-1} is assumed to be sparse, a small random difference between Σ_* and Σ would make S^{opt} highly dense. In order to address the aforementioned issues, consider the problem

$$\begin{aligned} \min_{S \in \mathbb{S}^d} \quad & -\log\det(S) + \text{trace}(\Sigma S) + \lambda \|S\|_* \\ \text{s.t.} \quad & S \succeq 0 \end{aligned} \tag{2}$$

where $\lambda \in \mathbb{R}_+$ is a regularization parameter and $\|S\|_*$ denotes the sum of the absolute values of the off-diagonal entries of S. This problem is referred to as *Graphical Lasso* (GL). Intuitively, the term $\|S\|_*$ in the objective function serves as a surrogate for promoting sparsity among the off-diagonal entries of S, while ensuring that the problem is well-defined even with a singular input Σ.

There have been major interests in studying the properties of GL as a conic optimization problem, in addition to the design of numerical algorithms for this problem [52, 53, 58]. For example, [59] and [60] have shown that the conic optimization problem (2) is highly related to a simple thresholding technique. This result is leveraged in [61] to obtain an explicit formula that serves as an exact solution of GL for acyclic graphs and as an approximate solution of GL for arbitrary sparse graphs. Another line of work has been devoted to studying the connectivity structure of the optimal solution of the GL problem. In particular, [62] and [63] have proved that the connected components induced by thresholding the sample correlation matrix and the connected components in the support graph of the optimal solution of the GL problem lead to the same vertex partitioning.

3.2. Optimization for Power Systems

The real-time operation of an electric power network depends heavily on several large-scale optimization problems solved from every few minutes to every several months. State estimation, optimal power flow (OPF), unit commitment, and network reconfiguration are some fundamental optimization problems solved for transmission and distribution networks. These different problems have all been built upon the power flow equations. Regardless of their large-scale nature, it is a daunting challenge to solve these problems efficiently. This is a consequence of the nonlinearity/non-convexity created by two different sources: (i) discrete variables such as the ratio of a tap-changing transformer, the on/off status of a line switch, or the commitment parameter of a generator, and (ii) the laws of physics. Issue (i) is more or less universal and researchers in many fields of study have proposed various sophisticated methods to handle integer variables. In contrast, Issue (ii) is pertinent to power systems, and it demands new specialized techniques and approaches. More precisely, complex power being a quadratic function of complex bus voltages imposes quadratic constraints on OPF-based optimization problems, and makes them NP-hard [64].

OPF is at the heart of Independent System Operator (ISO) power markets and vertically integrated utility dispatch [65]. This problem needs to be solved annually for system planning, daily for day-ahead commitment markets, and every 5-15 minutes for real-time market balancing. The existing solvers for OPF-based optimization either make potentially very conservative approximations or deploy general-purpose local-search algorithms. For example, a linearized version of OPF, named DC OPF, is normally solved in practice, whose solution may not be physically meaningful due to approximating the laws of physics. Although OPF has been studied for 50 years, the algorithms deployed by ISOs suffer from several issues, which may incur tens of billions of dollars annually [65].

The power flow equations for a power network are quadratic in the complex voltage vector. It can be verified that the constraints of the above-mentioned OPF-based

Table 1: Performance of penalized SDP for OPF.

Test cases	Near-opt. cost	Global opt. guarantee	Run time (s)
Polish 2383wp	1874322.65	99.316%	529
Polish 2736sp	1308270.20	99.970%	701
Polish 2737sop	777664.02	99.995%	675
Polish 2746wop	1208453.93	99.985%	801
Polish 2746wp	1632384.87	99.962%	699
Polish 3012wp	2608918.45	99.188%	814
Polish 3120sp	2160800.42	99.073%	910

problems can often be cast as quadratic constraints after introducing certain auxiliary parameters. This has inspired many researchers to study the benefits of conic optimization for power optimization problems. In particular, it has been shown in a series of papers that a basic SDP relaxation of OPF is exact and finds global minima for benchmarks examples [66–69]. By leveraging the physics of power grids, it is also theoretically proven that the SDP relaxation is always exact for every distribution network and every transmission network containing a sufficient number of transformers (under some technical assumptions) [70, 71].

The papers [72] and [68] show that if the SDP relaxation is not exact due to the violation of certain assumptions, a penalized SDP relaxation would work for a carefully chosen penalty term, which leads to recovering a near-global solution. This technique is tested on several real-world grids and the outcome is partially reported in Table 1 [73]. The number in the name of the test cases in the left column corresponds to the number of nodes in the network. It can be observed that the SDP relaxation has found operating points for the nationwide grid of Poland in different times of the year, where the global optimality guarantee of each solution is at least 99%, implying that the unknown global minima are at most 1% away from the obtained feasible solutions. In some cases, finding a suitable penalization parameter can be challenging, but this can be remedied by using [74, Algorithm 1] based on the notion of Laplacian matrix. Different techniques have been proposed in the literature to obtain tighter relaxations of OPF [75–79]. Several papers have developed SDP-based approximations or reformulations for a wide range of power problems, such as state estimation [80, 81], unit commitment [17], and transmission switching [16, 82]. The recent findings in this area show the significant potential of conic relaxation for power optimization problems.

3.3. Matrix Completion

Consider a symmetric matrix $\widehat{X} \in \mathbb{S}^n$, where certain off-diagonal entries are missing. The low-rank positive-semidefinite matrix completion problem is concerned with the design of the unknown entries of this matrix in such a way that the matrix becomes positive semidefinite with the lowest rank possible. This problem has applications in signal processing and machine learning, where the goal is to learn a structured dataset (or signal) from limited observations (or samples). It is also related to the complexity reduction for semidefinite programs [24, 83, 84]. Let the known entries of $\widehat{X}$ be represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the vertex set $\mathcal{V}$ and the edge set $\mathcal{E}$, where each edge of the graph corresponds to a known off-diagonal entry of $\widehat{X}$. The matrix completion problem can be expressed as:

$$\begin{aligned}
\min_{X \in \mathbb{S}^n} \quad & \operatorname{rank}\{X\} && && \text{(3a)}\\
\text{s.t.} \quad & X_{ij} = \widehat{X}_{ij}, && \forall (i,j) \in \mathcal{E} && \text{(3b)}\\
& X_{kk} = \widehat{X}_{kk}, && \forall k \in \mathcal{V} && \text{(3c)}\\
& X \succeq 0 && && \text{(3d)}
\end{aligned}$$

The missing entries of $\widehat{X}$ can be obtained from an optimal solution of the above problem. Since (3) is nonconvex, it is desirable to find a convex formulation/approximation of this problem. To this end, consider a number $\bar{n}$ that is greater than or equal to n. Let $\bar{\mathcal{G}} = (\bar{\mathcal{V}}, \bar{\mathcal{E}})$ be a graph with $\bar{n}$ vertices such that $\mathcal{G}$ is a subgraph of $\bar{\mathcal{G}}$. Consider the optimization problem

$$\begin{aligned}
\min_{\bar{X} \in \mathbb{S}^{\bar{n}}} \quad & \sum_{(i,j) \in \bar{\mathcal{E}} \setminus \mathcal{E}} t_{ij}\ \bar{X}_{ij} && && \text{(4a)}\\
\text{s.t.} \quad & \bar{X}_{ij} = \widehat{X}_{ij}, && \forall (i,j) \in \mathcal{E} && \text{(4b)}\\
& \bar{X}_{kk} = \widehat{X}_{kk}, && \forall k \in \mathcal{V} && \text{(4c)}\\
& \bar{X}_{kk} = 1, && \forall k \in \bar{\mathcal{V}} \setminus \mathcal{V} && \text{(4d)}\\
& \bar{X} \succeq 0 && && \text{(4e)}
\end{aligned}$$

Define $\bar{X}^{\text{opt}}(n)$ as the $n \times n$ principal submatrix of an optimal solution of (4). It is shown in [85] that:

- $\bar{X}^{\text{opt}}(n)$ is a positive semidefinite filling of $\widehat{X}$ whose rank is upper bounded by certain parameters of the graph $\bar{\mathcal{G}}$, no mater what the coefficients t_{ij}'s are as long as they are all nonzero.
- $\bar{X}^{\text{opt}}(n)$ is low rank if the graph $\mathcal{G}$ is sparse.
- $\bar{X}^{\text{opt}}(n)$ is guaranteed to be a solution of (3) for certain types of graphs $\mathcal{G}$.

Since (4) is a semidefinite program, it points to the role of conic optimization in solving the matrix completion problem.

3.4. Affine Rank Minimization Problem

Consider the problem

$$\begin{aligned}
\min_{Y \in \mathbb{R}^{m \times r}} \quad & \operatorname{rank}\{Y\} && && \text{(5a)}\\
\text{s.t.} \quad & \operatorname{trace}\{N_k Y\} \le a_k, && k = 1, \ldots, p && \text{(5b)}
\end{aligned}$$

where $N_1, \ldots, N_p \in \mathbb{R}^{r \times m}$ are constant sparse matrices. This is a general affine rank minimization problem without any positive semidefinite constraint. A special case of

the above problem is the regular matrix completion problem defined as finding the missing the entries of a rectangular matrix with partially known entries to minimize its rank [86–90]. A popular convexification method for (5) is to replace its objective function with the nuclear norm of Y [88]. This is motivated by the fact that the nuclear norm function is the convex envelop of rank$\{Y\}$ on the set $\{Y \in \mathbb{R}^{m\times r} \,|\, \|Y\| \leq 1\}$ [91]. Optimization (5) can be reformulated as a matrix optimization problem whose matrix variable is symmetric and positive semidefinite. To explain this reformulation, consider a new matrix variable X defined as

$$X \triangleq \begin{bmatrix} Z_1 & Y \\ Y^{\mathrm{T}} & Z_2 \end{bmatrix}, \tag{6}$$

where Z_1 and Z_2 are auxiliary matrices, and Y acts as a submatrix of X corresponding to its first m rows and last r columns. Now, consider the problem

$$\begin{aligned} \min_{X\in\mathbb{S}^{m+r}} \quad & \text{rank}\{X\} && \text{(7a)}\\ \text{s.t.} \quad & \text{trace}\{M_k X\} \leq a_k, \quad k = 1,\ldots,p && \text{(7b)}\\ & X \succeq 0 && \text{(7c)} \end{aligned}$$

where

$$M_k \triangleq \begin{bmatrix} 0_{m\times m} & \frac{1}{2}N_k^{\mathrm{T}} \\ \frac{1}{2}N_k & 0_{r\times r} \end{bmatrix} \tag{8}$$

For every feasible solution X of the above problem, its associated submatrix Y is feasible for (5) and satisfies the inequality

$$\text{rank}\{Y\} \leq \text{rank}\{X\} \tag{9}$$

The above inequality turns into an equality at optimality and, moreover, the problems (5) and (7) are equivalent [91, 92]. One may replace the rank objective in (7) with the linear term trace$\{X\}$ based on the nuclear norm technique, or use the more general idea delineated in (4) to find an SDP approximation of the problem with a guarantee on the rank of the solution.

3.5. Conic Relaxation of Quadratic Optimization

Consider the standard non-convex quaddratically-constrained quadratic program (QCQP):

$$\begin{aligned} \min_{x\in\mathbb{R}^{n-1}} \quad & x^T A_0 x + 2b_0^T x + c_0 && \text{(10a)}\\ \text{s.t.} \quad & x^T A_k x + 2b_k^T x + c_k \leq 0, \quad k = 1,\ldots,m && \text{(10b)} \end{aligned}$$

where $A_k \in \mathbb{S}^{n-1}$, $b_k \in \mathbb{R}^{n-1}$ and $c_k \in \mathbb{R}$ for $k = 0,\ldots,m$. This problem can be reformulated as

$$\begin{aligned} \min_{X\in\mathbb{S}^{n}} \quad & \text{trace}\{M_0 X\} \\ \text{s.t.} \quad & \text{trace}\{M_k X\} \leq 0, \quad k = 1,\ldots,m \\ & X_{11} = 1, \\ & X \succeq 0, \\ & \text{rank}\{X\} = 1 \end{aligned} \tag{11}$$

where X plays the role of

$$\begin{bmatrix} 1 \\ x \end{bmatrix}[1 \quad x^T]. \tag{12}$$

and

$$M_k = \begin{bmatrix} c_k & b_k^T \\ b_k & A_k \end{bmatrix}, \qquad k = 0,\ldots,m \tag{13}$$

Dropping the rank constraint from (11) leads to an SDP relaxation of the QCQP problem (7). If the matrices M_k's are sparse, then the SDP relaxation is guaranteed to have a low-rank solution. To enforce the low-rank solution to be rank-1, one can use the idea described in (4) and find a penalized SDP approximation of (7) by first dropping the rank constraint from (11) and then adding a penalty term similar to $\sum_{(i,j)\in\bar{\mathcal{E}}\setminus\mathcal{E}} t_{ij}\, X_{ij}$ to its objective function.

In an effort to study low-rank solutions of the SDP relaxation of (7), let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with n vertices such that $(i, j) \in \mathcal{G}$ if the (i, j) entry of at least one of the matrices $M_0, M_1, ..., M_m$ is nonzero. The graph $\mathcal{G}$ captures the sparsity of the optimization problem (7). Notice that those off-diagonal entries of X that correspond to non-existent edges of $\mathcal{G}$ play no direct role in the SDP relaxation. Let $\widehat{X}$ denote an arbitrary solution of the SDP relaxation of (7). It can be observed that every solution to the low-rank positive-semidefinite matrix completion problem (3) is a solution of the SDP relaxation as well. Now, one can use the SDP problem (4) to find low-rank feasible solutions of the SDP relaxation of the QCQP problem. By carefully picking the graph $\bar{\mathcal{G}}$, one can obtain a feasible solution of the SDP relaxation whose rank is less than or equal to the treewidth of $\mathcal{G}$ plus 1 (this number is expected to be small for sparse graphs) [85, 93].

3.6. State Estimation

Consider the problem of recovering the state of a nonlinear system from noisy data. Without loss of generality, we only focus on the quadratic case, where the goal is to find an unknown state/solution $x \in \mathbb{R}^n$ for a system of quadratic equations of the form

$$z_r = x^T M_r x + \omega_r, \qquad \forall r \in \{1,\ldots,m\} \tag{14}$$

where

- $z_1,\ldots,z_m \in \mathbb{R}$ are given measurements/specifications.
- Each of the parameters $\omega_1,\ldots,\omega_m$ is an unknown measurement noise with some known statistical information.
- $M_1,\ldots,M_m$ are constant $n \times n$ matrices.

One example is power systems state estimation. There, the vector x corresponds to a vector of voltage phasors, each corresponding to a different bus in the electricity transmission network. The measurements are made by a control system called supervisory control and data acquisition (SCADA). The classical SCADA measurements

of nodal and branch powers, and voltage magnitudes, can all be posed in the quadratic form shown above.

Several algorithms in different contexts, such as signal processing, have been proposed in the literature for solving special cases of the above system of quadratic equations [94–99]. These methods are often related to semidefinite programming. As an example, consider the conic program

$$\min_{\substack{X\in\mathbb{S}^n\\ \nu\in\mathbb{R}^m}} \quad \text{trace}\{MX\} + \mu\left(\frac{|\nu_1|}{\sigma_1}+\cdots+\frac{|\nu_m|}{\sigma_m}\right) \tag{15a}$$

$$\text{s.t.} \quad \text{trace}\{M_rX\} + \nu_r = z_r, \qquad r = 1,\ldots,m \tag{15b}$$

$$X \succeq 0 \tag{15c}$$

where $\mu > 0$ is a sufficiently large fixed parameter. The objective function of this problem has two terms: one taking care of non-convexity and another one for estimating the noise values. The matrix M can be designed such that the solution X^{opt} of the above conic program and the unknown state x be related through the inequality:

$$\|X^{\text{opt}} - \alpha xx^T\| \leq 2\sqrt{\frac{\mu \times \|\omega\|_1 \times \text{trace}\{X^{\text{opt}}\}}{\eta}} \tag{16}$$

where $\omega := [\omega_1 \ \cdots \ \omega_m]$, and α and η are constants [100]. This implies that the distance between the solutions of the conic program and the unknown state depends on the power of the noise. A slightly different version of the above result holds even in the case where a modest number of the values $\omega_1,\ldots,\omega_m$ are arbitrarily large corresponding to highly corrupted/manipulated data [81]. In that case, as long as the number of such measurements is not relatively large, the solution of the conic program will not be affected. The proposed technique has been successfully tested on the European Grid, where more than 18,000 parameters were successfully estimated in [100, 101].

4. Convexification Techniques

In this section, we present convex relaxations for solving difficult non-convex optimization problems. These non-convex problems include integer programs, quaddratically-constrained quadratic programs, and more generally polynomial optimization. There are countless examples: MAX CUT, MAX SAT, traveling salesman problem, pooling problem, angular synchronization, optimal power flow, etc. Convex relaxations are crucial when searching for integer solutions (e.g. via branch-and-bound); when optimizing over real variables, they provide a way to find globally optimal solutions, as opposed to local solutions. Our focus is on hierarchies of relaxations that grow tighter and tighter towards the original non-convex problem of interest. That is, hierarchies are a sequence of (generally) convex optimization problems whose optimal values become closer and closer to the global optimum of the non-convex problem. Generally, each convex optimization problem is guaranteed to provide a bound on the global optimum. The common framework we consider is that of optimizing a polynomial f of n variables constrained by m polynomial inequalities, i.e.

$$\inf_{x\in\mathbb{R}^n} \ f(x) \quad \text{s.t.} \quad g_i(x) \geqslant 0, \qquad i = 1,\ldots,m.$$

We consider linear programming (LP) hierarchies, second order-conic programming (SOCP) hierarchies, and semidefinite positive (SDP) hierarchies. These constitute three alternative ways of solving polynomial optimization problems, each with their pros and cons. After briefly recalling some of the historical contributions, we refer to recent developments that have taken place over the last three years. They are aimed at making the hierarchies more tractable. In particular, the recently proposed *multi-ordered Lasserre hierarchy* can solve a key industrial problem with thousands of variables and constraints to global optimality. The Lasserre hierarchy was previously limited to small-scale problems since it was introduced 17 years ago.

4.1. LP hierarchies

When solving integer programs, it is quite natural to consider convex relaxations in the form of linear programs. The idea is to come up with a polyhedral representation of the convex hull of the feasible set so that all the vertices are integral. In that case, optimizing over the representation can yield the desired integral solutions. For an excellent reference on the various approaches, including those of Gomori and Chvátal, see the recent book [15]. It is out of this desire to obtain nice representations that the Sherali-Adams [102] and Lovász-Schrijver [103] hierarchies arose in 1990; they both provide tighter and tighter outer approximations of the convex hull of the feasible set. In fact, after a finite number of steps (which is known *a priori*), their polyhedral representations coincide with the convex hull of the integral feasible set. At each iteration of the hierarchy, the representation of Sherali-Adams is contained in the one of Lovász-Schrijver [20]. One way to view these hierarchies is via lift-and-project. For instance, the Sherali-Adams hierarchy can be viewed as taking products of the constraints, which are redundant for the original problem, but strengthen the linear relaxation. Interestingly, the well-foundedness of this approach, and indeed the convergence of the Sherali-Adams hierarchy, can be justified by the works of Krivine [104, 105] in 1964, as well as those of Cassier [106] (in 1984) and Handelman [107] (in 1988). It was Lasserre [21, Section 5.4] who recognized that, thanks to Krivine, one can generalize the approach of Sherali-Adams to polynomial optimization. Global convergence is ensured under some mild assumptions which can always be met when the feasible set is compact. In order to meet these assumptions, one needs to make some adjustements to the modeling of the feasible set. These include normalizing the constraints to be between zero and one.

Example 1. *Consider the following polynomial optimization problem taken from* [21, Example 5.5]:

$$\inf_{x\in\mathbb{R}} \; x(x-1) \quad \text{s.t.} \quad x \geqslant 0 \;\text{ and }\; 1-x \geqslant 0$$

Its optimal value is $-1/4$*. To obtain the second-order LP relaxation, one can add the following redundant constraints:*

$$x^2 \geqslant 0, \quad x(1-x) \geqslant 0, \quad (1-x)^2 \geqslant 0$$

The lifted problem then reads:

$$\inf_{y_1,y_2\in\mathbb{R}} \; y_2 - y_1 \quad \text{s.t.} \quad \left\{ \begin{array}{r} y_1 \geqslant 0 \\ 1-y_1 \geqslant 0 \\ y_2 \geqslant 0 \\ y_1 - y_2 \geqslant 0 \\ 1-2y_1+y_2 \geqslant 0 \end{array} \right.$$

where y_k *corresponds to* x^k *for a positive integer* k*. An optimal solution to this problem is* $(y_1, y_2) = (1/2, 0)$*. We then obtain a lower bound on the original problem equal to* $-1/2$*. The third-order LP relaxation is obtained by adding yet more redundant constraints:*

$$x^3 \geqslant 0, \quad x^2(1-x) \geqslant 0, \quad x(1-x)^2 \geqslant 0 \quad (1-x)^3 \geqslant 0$$

Now, the lifted problem reads:

$$\inf_{y_1,y_2\in\mathbb{R}} \; y_2 - y_1 \quad \text{s.t.} \quad \left\{ \begin{array}{r} y_1 \geqslant 0 \\ 1-y_1 \geqslant 0 \\ y_2 \geqslant 0 \\ y_1 - y_2 \geqslant 0 \\ 1-2y_1+y_2 \geqslant 0 \\ y_3 \geqslant 0 \\ y_2 - y_3 \geqslant 0 \\ y_1 - 2y_2 + y_3 \geqslant 0 \\ 1-3y_1+3y_2-y_3 \geqslant 0 \end{array} \right.$$

An optimal solution is given by $(y_1, y_2, y_3) = (1/3, 0, 0)$*, yielding the lower bound of* $-1/3$*. And so on and so forth.*

While convergence of the Sherali-Adams hierarchy is preserved when generalizing their approach to polynomial optimization, *finite* convergence is not preserved. In the words of Lasserre [21, Section 5.4.2]: "*Unfortunately, we next show that in general the LP-relaxations cannot be exact, that is, the convergence is only asymptotic, not finite.*" This means that, while the global value can be approached to abritrary accuracy, it may never be reached. As a consequence, one cannot hope to extract global minimizers, i.e. those points that satisfy all the constraints and whose evaluations are equal to the global value.

We now turn our attention to some recent work on designing LP hierarchies for polynomial optimization on the positive orthant, i.e. with the constraints $x_1 \geqslant 0, \dots x_n \geqslant 0$. Invoking a result of Póyla in 1928 [108], it was recently proposed in [109] to multiply the objective function by $(1 + g_1(x) + \ldots + g_m(x))^r$ for some positive integer r, in addition to taking products of the constraints (as in the aforementioned LP hierarchy). This work comes as a result of generalizing the notion of copositivity from quadratic forms to polynomial functions [110].

We next discuss some numerical aspects of LP hierarchies for polynomial optimization. It has been shown that multiplying constraints by one another when applying lift-and-project to integer programs is very efficient in practice. However, it can lead to numerical issues for polynomial optimization. The reason for this is that the coefficients in the polynomial constraints are not necessarily all of the same order of magnitude. For example, in the univariate case, multiplying $0.1x - 2 \geqslant 0$ by itself yields $0.01x^2 - 0.2x + 4 \geqslant 0$, leading to coefficients ranging two orders of magnitude. This can be challenging, even for state-of-the-art software in linear programming. To date, there exists no way of scaling the coefficients of a polynomial optimization problem so as to make them more or less homogenous. In contrast, in integer programs, the constraints $x_i^2 - x_i = 0$ naturally have all coefficients equal to zero or one. As can be read in [111, page 7]: "*We remark that while there have been other approaches to produce LP hierarchies for polynomial optimization problems (e.g., based on the Krivine-Stengle certificates of positivity* [21, 105, 112]*), these LPs, though theoretically significant, are typically quite weak in practice and often numerically ill-conditioned* [113]." This leads us to discuss the LP hierarchy proposed in [114] in 2014.

Viewed through the lenses of lift-and-project, the approach in [114] avoids products of constraints and instead adds the redundant constraints $x^{2\alpha} g_i(x) \geqslant 0$ and $(x^\alpha \pm x^\beta)^2 g_i(x) \geqslant 0$ where $x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ and $\alpha, \beta \in \mathbb{N}^n$. By doing this for monomials of higher and higher degree $\alpha_1 + \ldots + \alpha_n$, one obtains an LP hierarchy. The nice property of this hierarchy is that it avoids the conditioning issues associated with the previously discussed hierarchies. The authors also propose to multiply the objective by $(x_1^2 + \ldots + x_n^2)^r$ for some integer r as a means to strenghen the hierarchy. Global convergence can then be guaranteed (upon reformulation) when optimizing a homogenous polynomial whose individual variables are raised only to even degrees and are constrained to lie in the unit sphere. This follows from [111, Theorem 13], a consequence of Pólya's previously mentioned result [108]. We conclude by noting that the distinct LP hierarchies presented above can be combined. For the interested reader, numerical experiments can be found in [115, Table 4.4].

4.2. SOCP hierarchies

A natural way to provide hierarchies that are stronger than LP hierarchies is to resort to conic optimization whose feasible set is not a polyhedra. In fact, in their original paper, Lovász and Schrijver proposed strengthening their LP hierarchy by adding a positive semidefinite constraint. We will deal with SDP in the next section, and we now focus on SOCP for which very efficient solvers

exist. It was this practical consideration which led the authors of [114] to restrain the cone of sum-of-squares arising in the Lasserre hierarchy. By restricting the number of terms inside the squares to two at most, i.e. by avoiding squares such as the one crossed out in $\sigma(x_1, x_2) = x_1^2 + (2x_1 - x_2^3)^2 + \cancel{(1 - x_1 + x_2)^2}$, one obtains SOCP constraints instead of SDP constraints. Numerical experiments can found in [116]. A dual perspective to this approach is to relax the semidefinite constraints in the Lasserre hierarchy to all necessary SOCP constraints. This idea was independently proposed in [117], except that the authors of that work do not relax the moment matrix. This ensures that the relaxation obtained is at least as tight as the first-order Lasserre hiearchy. When applied to find global minimizers to the optimal power flow problem [118], this provides reduced runtime in some instances. On other instances, the hierarchy does not seem to globally converge, at least with the limited computional power used in the experiments. This was elucidated in [119] where it was shown that restricting sum-of-squares to two terms at most does not preserve global convergence, even if the polynomial optimization problem is convex. Interestingly, the restriction on the sum-of-squares can be used to strenghten the LP hierarchies described in the previous section [115]. However, this does not affect the ill-conditioning associated with the LP hierarchies, nor their asymptotic convergence (as opposed to finite convergence).

We note that SOCP hierarchies have been successfully applied for solving control problems in robotics. Below we borrow an example from [120] to illustrate the use of sums-of-squares and SOCP hierarchies to control.

Example 2. *Consider the following dynamical system which represents a jet engine model:*

$$\begin{array}{rcl} \dot{x} & = & -y - \frac{3}{2}x^2 - \frac{1}{2}x^3 \\ \dot{y} & = & 3x - y \end{array}$$

In order to prove the stability of the equilibrium $(x, y) = (0, 0)$, one may look for a Lyapunov function using sums-of-squares, i.e. a function $V : \mathbb{R}^2 \longrightarrow \mathbb{R}$ such that $V(x, y)$ and $-\dot{V}(x, y)$ are sums-of-squares. In this case, it suffices to look for sums-of-squares of degree 4, which yields the following Lyapunov function [121]:

$$\begin{array}{rcl} V(x,y) & = & 4.5819x^2 - 1.5786xy + 1.7834y^2 - 0.12739x^3 \\ & & +2.5189x^2y - 0.34069xy^2 + 0.61188y^3 \\ & & +0.47537x^4 - 0.052424x^3y + 0.44289x^2y^2 \\ & & +0.0000018868xy^3 + 0.090723y^4 \end{array}$$

In order to reduce the computational burden, one could require that $V(x, y)$ and *$-\dot{V}(x, y)$ are sums-of-squares where the number of terms inside the squares are limited to two at most. This yields an SOCP feasibility problem. This technique has been used to find regions of attraction to problems arising in robotics* [114].

4.3. SDP hierarchies

SDP hierarchies revolve around the notion of sum-of-squares, which were first introduced in the context of optimization by Shor [122] in 1987. Shor showed that globally minimizing a univariate polynomial on the real line breaks down to a convex problem. It relies on the fact that a univariate polynomial is nonnegative if and only if it is a sum-of-squares of other polynomials. This is also true for bivariate polynomials of degree four as well as multivariate quadratic polynomials. But it is generally not true for other polynomials, as was shown by Hilbert in 1888 [123]. This led Shor [124] (see also [125]) to tackle the minimization of multivariate polynomials by reformulating them using quadratic polynomials. Later, Nesterov [126] provided a self-concordant barrier that allows one to use efficient interior point algorithms to minimize a univariate polynomial via sum-of-squares.

We turn our attention to the use of sum-of-squares in a more general context, i.e. constrained optimization. Working on Markov chains where one seeks invariant measures, Lasserre [19] realized that minimizing a polynomial function under polynomial constraints can also be viewed as a problem where one seeks a measure. He showed that a dual perspective to this approach consists in optimizing over sum-of-squares. In order to justify the global convergence of his approach, Lasserre used Putinar's Positivstellensatz [127]. This result was discovered in 1993 and provided a crucial refinement of Schmüdgen's Positivstellensatz [128] proven a few years earlier. It was crucial because it enabled numerical computations, leading to what is known today as the Lasserre hierarchy. Schmüdgen's Positivstellensatz essentially says that a polynomial that is positive on a set defined by polynomial inequalities can be decomposed a sum of products of the the polynomi-als multiplied by sums of squares; Putinar's removes the product from the decomposition.

Example 3. *Consider the following polynomial optimization problem taken from* [119]:

$$\inf_{x_1, x_2 \in \mathbb{R}} \; x_1^2 + x_2^2 + 2x_1x_2 - 4x_1 - 4x_2 \;\; \text{s.t.} \;\; x_1^2 + x_2^2 = 1$$

Its optimal value is $2 - 4\sqrt{2}$, which can be found using sums-of-squares since:

$$\begin{array}{c} x_1^2 + x_2^2 + 2x_1x_2 - 4x_1 - 4x_2 \quad - \quad (2 - 4\sqrt{2}) \quad = \\ (\sqrt{2} - 1)(x_1 - x_2)^2 \quad + \quad \sqrt{2}(-\sqrt{2} + x_1 + x_2)^2 \\ + \;\; 2(\sqrt{2} - 1)(1 - x_1^2 - x_2^2) \end{array}$$

It can be seen from the above equation that when (x_1, x_2) is feasible, the first line must be nonnegative, proving that $2 - 4\sqrt{2}$ is a lower bound. This corresponds to the first-order Lasserre hierarchy since the polynomials inside the squares are of degree one at most. To make the link with the previous section, note that one can ask to restrict the number of terms to two at most inside the squares. This allows one to use second-order conic programming instead of semidefinite programming, but does not preserve global

convergence. The best bound that can be obtained (i.e. $-4\sqrt{2}$) is given by the following decomposition:

$$x_1^2 + x_2^2 + 2x_1x_2 - 4x_1 - 4x_2 \quad - \quad (-4\sqrt{2}) \quad =$$
$$\frac{\sqrt{2}}{2}(-\sqrt{2} + 2x_1)^2 \quad + \quad \frac{\sqrt{2}}{2}(-\sqrt{2} + 2x_2)^2 + (x_1 + x_2)^2$$
$$+ \quad 2\sqrt{2}(1 - x_1^2 - x_2^2)$$

Parallel to Lasserre's contribution, Parrilo [18] pioneered the use of sum-of-squares for obtaining strong bounds on the optimal solution of nonconvex problems (e.g. MAX CUT). He also showed how they can be used for many important problems in systems and control. These include Lyapunov analysis for control systems [129]. We do not dwell on these as they are outside the scope of this tutorial, but we also mention a different view of optimal control via occupation measures [130]. In contrast to Lasserre, Parrilo's work [131] panders to Stengle's Positivstellensatz [112], which is used for proving infeasibility of systems of polynomial equations. This result can be seen as a generalization of Farkas' Lemma which certifies the emptiness of a polyhedral set.

As discussed above, the Lasserre hierarchy provides a sequence of semidefinite programs whose optimal values converge (monotonically) towards the global value of a polynomial optimization problem. This is true provided that the feasible set is compact, and that a bound R on the radius of the set is known, so that one can include a redundant ball constraint $x_1^2 + \cdots + x_n^2 \leqslant R^2$. When modeling the feasible set in this matter, there is also zero duality gap in each semidefinite program [132]. This is a crucial property when using path following primal-dual interior point methods, which are some of the most efficient approaches for solving semidefinite programs.

In contrast to LP hierarchies which have only asymptotic convergence in general, the Lasserre hierarchy has finite convergence generically. This means that for a given abritary polynomial optimization problem, finite convergence will almost surely hold. It was Nie [133] who proved this result, which had been observed in practice ever since the Lasserre hierarchy was introduced. He relied on theorems of Marshall [134, 135] which attempted to answer the question: when can a nonnegative polynomial have a sum-of-squares decomposition? In the Positivstellensätze discussed above, the assumption of positivity is made, which only guarantees asymptotic convergence. The result of Nie marks a crucial difference with the LP hierarchies because it means that in practice, one can solve non-convex problems exactly via a convex relaxation, whereas with LP hierarchies one may only approximate them. In fact, when finite convergence is reached, the Lasserre hierarchy not only provides the global value, but also finds global minimizers, i.e. points that satisfy all the constraints and whose evaluations are the global value. This last feature illustrates a nice synergy between advances in optimization and advances on the theory of moments, which we next discuss.

When the Lasserre hierarchy was introduced, the theory of moments lacked a result to guarantee when global solutions could be extracted. At the time of Lasserre's original paper, the theory only applied to bivariate polynomial optimization [136, Theorem 1.6]. With the success of the Lasserre hierarchy, there was a growing need for more theory to be developed. This theory was developed a few years later by Curto and Fialkow [137, Theorem 1.1]. They showed that it is sufficient to check a rank condition in the Lasserre hierarchy in order to extract global minimizers (and in fact, the number of minimizers is equal to the rank).

Interestingly, the same situation occured when the complex Lasserre hierarchy was recently introduced in [138]. The Lasserre hierarchy was generalized to complex numbers in order to enhance its tractability when dealing with polynomial optimization in complex numbers, i.e.

$$\inf_{z \in \mathbb{C}^n} \; f(z, \bar{z}) \quad \text{s.t.} \quad g_i(z, \bar{z}) \geqslant 0, \; i = 1, \ldots, m.$$

where $f, g_1, \ldots, g_m$ are real-valued complex polynomials (e.g. $2|z_1|^2 + (1+\mathbf{i})z_1\bar{z}_2 + (1-\mathbf{i})\bar{z}_1 z_2$, where $\mathbf{i}$ is the imaginary number). This framework is natural for optimization problems with oscillatory phenoma, which are omnipresent in physical systems (e.g. electric power systems, imaging science, signal processing, automatic control, quantum mechanics). One way of viewing the complex Lasserre hierarchy is that it restricts the sums-of-squares in the original Lasserre hierarchy to *Hermitian* sums-of-squares. These are exponentially cheaper to compute yet preserve global convergence, thanks to D'Angelo's and Putinar's Positivstellensatz [139]. On the optimal power flow problem in electrical engineering, they permit a speed-up factor of up to one order of magnitude [138, Table 1].

Example 4. *Consider the following complex polynomial optimization problem*

$$\inf_{z \in \mathbb{C}} \quad z + \bar{z} \quad \text{s.t.} \quad |z|^2 = 1$$

whose optimal value is -2. One way to solve this problem would be to convert it into real numbers $z =: x1 + \mathbf{i}x_2$, i.e.

$$\inf_{x_1, x_2 \in \mathbb{R}} \quad 2x_1 \quad \text{s.t.} \quad x_1^2 + x_2^2 = 1$$

and to use sums-of-squares:

$$2x_1 \; - \; (-2) \quad = \quad 1^2 + (x_1 + x_2)^2 \quad + \quad 1 \times (1 - x_1^2 - x_2^2)$$

But one could instead use Hermitian sums-of-squares, which are cheaper to compute:

$$z + \bar{z} \; - \; (-2) \quad = \quad |1 + z|^2 \quad + \quad 1 \times (1 - |z|^2)$$

When the complex Lasserre hierarchy was introduced, the theory of moments lacked a result to guarantee when global solutions could be extracted. This led its authors to generalize the work of Curto and Fialkow using the

notion of hyponormality in operator theory [138, Theorem 5.1]. They found that in addition to rank conditions, some positive semidefinite conditions must be met. Contrary to the rank conditions, these are convex and can thus be added to the complex Lasserre hierarchy. In [138, Example 4.1], doing so reduces the rank from 3 to 1 and closes the relaxation gap.

The advent of the Lasserre hierarchy not only sparked progress in the theory of moments, but also led to some notable results. In 2004, the author of [140] derived an upper bound on the order of the Lasserre hierarchy needed to obtain a desired relaxation bound to a polynomial optimization problem. This upper bound depends on three factors: 1) a certain description of the feasible set, 2) the degree of the polynomial objective function, and 3) how close the objective function is to reaching the relaxation bound on the feasible set. It must be noted that this upper bound is difficult to compute in practice. As of today, it is therefore not possible to know ahead of time how far in the hierarchy one needs to go in order to solve a given instance of polynomial optimization. Along these lines, not much is yet known about the speed of convergence of the bounds generated by the Lasserre hierarchy, except when optimizing over the hypercube [141, 142]. In practice, they generally reach the global value in a few iterations. Somewhat paradoxically, there are some nice results [143] on the speed of convergence of SDP hierarchies of upper bounds [144], although their converge is slow in practice.

Another notable result is contained in [145]. As discussed previously, the discrepancy between nonnegative polynomials and sum-of-squares was noticed towards the end of the nineteenth century. In some applications of sum-of-squares, the nonnegativity of a function on $\mathbb{R}^n$ is replaced by requiring it to be a sum-of-squares. The result in [145] quantifies how small the cone of sums-of-squares is with respect to the cone of nonnegative polynomials. It is perhaps the title of the paper that best sums up the finding: "*There are significantly more nonnegative polynomials than sums of squares*".

Having discussed several theoretical aspects of the Lasserre hierarchy, we now turn our attention to practical considerations. In order to make the Lasserre hierarchy tractable, it is crucial to exploit the problem structure. We have already gotten a flavor of this with the complex hierarchy, which exploits the complex structure of physical problems with oscillatory phenoma (electricity, light, etc.). In the following, some key results on sparsity and symmetry are highlighted.

In order to exploit sparsity in the Lasserre hierarchy, it was proposed to use chordal sparsity in sums-of-squares in [146]. We briefly explain this approach. Each constraint in a polynomial optimization problem is associated a sum-of-squares in the Lasserre hierarchy. In fact, each sum-of-squares can be interpreted as a generalized Lagrange multiplier [21, Theorem 5.12]. If a constraint only depends on a few variables, say x_1, x_3, x_{20} among $x_1, \ldots, x_{100}$, it seems naturally that the associated sum-of-squares should depend only on the variables x_1, x_3, x_{20}, or some slightly larger set of variables. This was made possible by the work in [146]. To do so, the authors consider the correlative sparsity pattern of the polynomial optimization problem. It can be viewed as a graph where the nodes are the variables and the edges signify a coupling of the variables. Taking a chordal extension of this graph and computing the maximal cliques, one can restraint the variables appearing in the sum-of-squares to belong to these cliques. This provides a more tractable hierarchy of relaxations while preserving global convergence, as was shown by Lasserre [21, Theorems 2.28 and 4.7]. What Lasserre proved was a sparse version of Punitar's Positivstellensatz. This sparse version has two applications that we next discuss.

One application is to the bounded sum-of-squares hierarchy (BSOS) [113]. The idea of this SDP hierarchy is to fix the size of the SDP constraint as the order of the hierarchy increases. The size of the SDP constraint can be set by the user. The hierarchy builds on the LP hierarchy based on Krivine's Positivstellensatz discussed in the section on LP hierarchies. As the order of the hierarchy increases, the number of LP constraints augments. These are the LP constraints that arise when multiplying constraints by one another. A sparse BSOS [147] is possible, thanks to the sparse version of Putinar's Positivstellensatz. Global convergence is guaranteed by [147, Theorem 1], but the ill-conditioning associated with LP hierarchies is inherited.

Another applicaton of the sparse version of Punitar's Positivstellensatz is the *multi-ordered* Lasserre hierarchy [138, 148]. It is based on two ideas: 1) to use a different relaxation order for each constraint, and 2) to iteratively seek a closest measure to the truncated moment data until a measure matches the truncated data. Global convergence is a consequence of the aforementioned sparse Positivstellensatz.

Example 5. *The multi-ordered Lasserre hierarchy can solve a key industrial problem of the twentieth century to global optimality on instances of polynomial optimization with up to* 4,500 *variables and* 14,500 *constraints* (*see also* [138]). *The relaxation order is typically augmented at a hundred or so constraints before reaching global optimality. The test cases correspond to the highly non-convex optimal flow problem, and in particular to instances of the European high-voltage synchronous electricity network comprising data from* 23 *different countries* (*available at* [149, 150]).

Table 2 shows the globally optimal objective value and corresponding solver time for select test cases. The convex relaxations were parsed using YALMIP 2015.06.26 [151] and MATLAB 2013a, and solved using MOSEK 7.1.0.28 on a quad-core 2.70 GHz processor with 16 GB of RAM. Parsing time is not included in the comparison; it could be substantially improved within an industrial-scale imple-mentation. Better runtimes (by up to an order of magni-

Table 2: Numerical results for Example 5

Case Name	Num. Var.	Num. Constr.	Multi-ordered Lasserre	
			Global val.	Time (s.)
case57Q	114	192	7,352	3.4
case57L	114	352	43,984	1.4
case118Q	236	516	81,515	15.7
case118L	236	888	134,907	10.5
case300	600	1,107	720,040	7.2
nesta_case24	48	526	6,421	246.1
nesta_case30	60	272	372	302.7
nesta_case73	146	1,605	20,125	506.9
PL-2383wp	4,354	12,844	24,990	583.4
PL-2746wop	4,378	13,953	19,210	2,662.4
PL-3012wp	4,584	14,455	27,642	318.7
PL-3120sp	4,628	13,948	21,512	386.6
PEGASE-1354	1,966	6,444	74,043	406.9
PEGASE-2869	4,240	12,804	133,944	921.3

tude) and higher precision can be obtained with the complex Lasserre hierarchy mentioned above.

We finish by discussing symmetry. In the presence of symmetries in a polynomial optimization problem, the authors of [152] proposed to seek an invariant measure when deploying the Lasserre hierarchy. This reduces the computational burden in the semidefinite relaxations. It was shown recently that in the presence of commonly encountered symmetries, one actually obtains a *block diagonal* Lasserre hierarchy [138, Section 7].

The above approaches for making hierarchies more tractable lead to convex models that are more amenable for off-the-shelf solvers. The next section deals with numerical algorithms for general conic optimization problems.

5. Numerical Algorithms

The previous section described *convexification* techniques that relax hard, nonconvex problems into a handful of standard classes of convex optimization problems, all of which can be approximated to arbitrary accuracy in polynomial time using the ellipsoid algorithm [153, 154]. Whenever the relaxation is tight, a solution to the original nonconvex problem can be recovered after solving the convexified problem. Accordingly, convexification establishes the original problem to be tractable or "easy to solve", at least in a theoretical sense. This approach was used as early as 1980 by Gröteschel, Lovász and Schijver [155] to develop polynomial-time algorithms for combinatorial optimization.

However, the practical usefulness of convexification was less clear at the time of its development. The ellipsoid method was notoriously slow in practice, so specialized algorithms had to be used to solve the resulting convexified problems. The fastest was the simplex method for the solution of linear programs (LPs), but LP convexifications are rarely tight. Conversely, semidefinite program (SDP) convexifications are often exact, but SDPs were particularly difficult to solve, even in very small instances (see [1, 156] for the historial context). As a whole, convexification remained mostly of theoretical interest.

In the 1990s, advancements in numerical algorithms overhauled this landscape. The interior-point method — originally developed as a practical but rigorous algorithm for LPs by Karmarkar [157] — was extended to SDPs by Nesterov and Nemirovsky [158, Chapter 4] and Alizadeh [22]. In fact, this line of work showed interior-point methods to be *particularly suitable* for SDPs, generalizing and unifying the much of the previous framework developed for LPs. For control theorists, the ability to solve SDPs from convexification had a profound impact, giving rise to the disciplines of LMI control [1] and polynomial control [12].

Today, the growth in the size of SDPs has outpaced the ability of general-purpose interior-point methods to solve them, fueled in a large part by the application of convexification techniques to control and machine learning applications. First-order methods have become popular, because they have very low per-iteration costs that can often be custom-tailored to exploit problem structure in a specific application. On the other hand, these methods typically require considerably more iterations to converge to reasonable accuracy. Ultimately, the most effective algorithms for the solution of large-scale SDPs are those that combine the convergence guarantees of interior-point methods with the ability of first-order methods to exploit problem structure.

This section reviews in detail three numerical algorithms for SDPs. First, we describe the theory of interior-point methods in Section 5.2, presenting them as a general-purpose algorithm for solving SDPs to arbitrary accuracy in polynomial time. Next, we describe a popular first-order method in Section 5.3 known as ADMM, and explain how it is able to reduce computational cost by exploiting sparsity. In Section 5.4, we describe a modified interior-point method for low-rank SDPs that exploits sparsity like ADMM while also enjoying the strong convergence guarantees of interior-point methods. Finally, we briefly review other structure-exploiting algorithms in Section 5.5.

5.1. Problem description

In order to simplify our presentation, we will focus our efforts on the standard form semidefinite program

$$\begin{aligned} X^{\text{opt}} = \text{minimize} \quad & C \bullet X \\ \text{subject to} \quad & A_i \bullet X = b_i \quad \forall i \in \{1, \ldots, m\},\ X \succeq 0, \end{aligned} \tag{SDP}$$

over the data $C, A_1, \ldots, A_m \in \mathbb{S}^n$, $b \in \mathbb{R}^m$, and its Lagrangian dual

$$\begin{aligned} \{y^{\text{opt}}, S^{\text{opt}}\} = \text{maximize} \quad & b^T y \\ \text{subject to} \quad & \sum_{i=1}^{m} y_i A_i + S = C,\ S \succeq 0. \end{aligned} \tag{SDD}$$

Here, $X \succeq 0$ indicates that X is positive semidefinite, and $A_i \bullet X = \text{trace}\{A_i X\}$ is the usual matrix inner product.

In case of nonunique solutions, we use $\{X^{\text{opt}}, y^{\text{opt}}, S^{\text{opt}}\}$ to refer to the *analytic center* of the solution set. We make the following nondegeneracy assumptions.

Assumption 1 (Linear independence)**.** *The matrix* $\mathbf{A} = [\operatorname{vec} A_1, \ldots, \operatorname{vec} A_m]$ *has full column-rank, meaning the matrix* $\mathbf{A}^T\mathbf{A}$ *is invertible.*

Assumption 2 (Slater's Condition)**.** *There exist* y *and positive definite* X *and* S *such that* $A_i \bullet X = b_i$ *holds for all* i*, and* $\sum_i y_i A_i + S = C$*.*

These are generic properties of SDPs, and are satisfied by almost all instances [159]. Linear independence implies that the number of constraints m cannot exceed the number of degrees of freedom $\frac{1}{2}n(n+1)$. Slater's condition is commonly satisfied by embedding (SDP) and (SDD) within a slightly larger problem using the homogenous self-dual embedding technique [160].

All of our algorithms and associated complexity bounds can be generalized in a straightforward manner to conic programs posed on the Cartesian product of many semidefinite cones $\mathcal{K} = \mathbb{S}^{n_1}_+ \times \mathbb{S}^{n_2}_+ \times \cdots \times \mathbb{S}^{n_\ell}_+$, as in

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{\ell} C_j \bullet X_j \\ \text{subject to} \quad & \sum_{j=1}^{\ell} A_{i,j} \bullet X_j = b_i \quad \forall i \in \{1, \ldots, m\} \\ & X_j \succeq 0 \quad \forall j \in \{1, \ldots, \ell\} \end{aligned} \tag{17}$$

and

$$\begin{aligned} \text{maximize} \quad & b^T y \\ \text{subject to} \quad & \sum_{i=1}^{m} y_i A_{i,j} + S_j = C_j \quad \forall j \in \{1, \ldots, \ell\} \\ & S_j \succeq 0 \quad \forall j \in \{1, \ldots, \ell\}. \end{aligned} \tag{18}$$

We will leave the specific details as an exercise for the reader. Note that this generalization includes linear programs (LPs), since the positive orthant is just the Cartesian product of many size-1 semidefinite cones, as in $\mathbb{R}^n_+ = \mathbb{S}^1_+ \times \cdots \times \mathbb{S}^1_+$.

At least in principle, our algorithms also generalize to second-order cone programs (SOCPs) by converting them into SDPs, as in

$$\|u\|_2 \leq u_0 \iff \begin{bmatrix} u_0 & u^T \\ u & u_0 I \end{bmatrix} \succeq 0.$$

However, as demonstrated by Lobo et al. [161], considerable efficiency can be gained by treating SOCPs as its own distinct class of conic problems.

5.2. Interior-point methods

The original interior-point methods were inspired by the logarithmic barrier method, which replaces each inequality constraint of the form $c(x) \geq 0$ by a logarithmic penalty term $-\mu \log c(x)$ that is well-defined at *interior-points* where $c(x) > 0$, but becomes unbounded from above as x approaches the boundary where $c(x) = 0$. (This behavior constitutes an infinite *barrier* that restricts x to lie within the feasible region where $c(x) > 0$.)

Consider applying this strategy to (SDP) and (SDD). If we intepret the semidefinite condition $X \succeq 0$ as a set of eigenvalue constraints $\lambda_j(X) \geq 0$ for all $j \in \{1, \ldots, n\}$, then the resulting logarithmic barrier is none other than the log-determinant penalty for determinant maximization (see [162] and the references therein)

$$-\sum_{j=1}^{n} \mu \log \lambda_j(X) = -\mu \log \prod_{j=1}^{n} \lambda_j(X) = -\mu \log \det X.$$

Substituting the penalty in place of the constraints $X \succeq 0$ and $S \succeq 0$ results in a sequence of unconstrained problems

$$\begin{aligned} X_\mu = \ & \text{minimize } C \bullet X - \mu \log \det X \\ & \text{subject to } A_i \bullet X = b_i \ \forall i \in \{1, \ldots, m\}, \end{aligned} \tag{SDPμ}$$

and

$$\begin{aligned} \{y_\mu, S_\mu\} = \ & \text{maximize } b^T y + \mu \log \det S \\ & \text{subject to } \sum_{i=1}^{m} y_i A_i + S = C, \end{aligned} \tag{SDDμ}$$

which can be shown to be primal-dual pairs (up to a constant offset).

After converting inequality constraints into logarithmic penalties, the barrier method repeatedly solves the resulting unconstrained problem using progressively smaller values of μ, each time reusing the most recent solution as the starting point for the next minimization. Applying this sequential strategy to solve (SDPμ) using Newton's method yields a *primal-scaled* interior-point method; doing the same for (SDDμ) results in a *dual-scaled* interior-point method. It is a seminal result of Nesterov and Nemirovski [158] that either interior-point methods converge to an approximate solution accurate to L digits after at most $O(nL)$ Newton iterations. (This can be further reduced to $O(\sqrt{n}L)$ Newton iterations by limiting the rate at which μ is reduced.) In practice, convergence almost never occurs in more than tens of iterations.

In finite precision, the primal-scaled and dual-scaled interior-point methods can suffer from severe accuracy and robustness issues; these are the same reasons that had originally caused the barrier method to fall out favor in the 1970s. Today, the most robust and accurate interior-point methods are *primal-dual*, and simultaneously solve (SDPμ) and (SDDμ) through their joint Karush-Kuhn-Tucker (KKT) optimality conditions

$$\begin{aligned} A_i \bullet X_\mu &= b_i \quad \forall i \in \{1, \ldots, m\}, \\ \sum_{i=1}^{m} y_i^\mu A_i + S_\mu &= C, \\ X_\mu S_\mu &= \mu I. \end{aligned}$$

Here, the barrier parameter $\mu > 0$ controls the duality gap between the point X_μ in (SDP) and the point $\{y_\mu, S_\mu\}$ in (SDD), as in $n\mu = X_\mu \bullet S_\mu = C \bullet X_\mu - b^T y_\mu$. In other words, the candidate solutions X_μ and $\{y_\mu, S_\mu\}$ are suboptimal for (SDP) and (SDD) respectively by an absolute figure no worse than $n\mu$,

$$C \bullet X^{\text{opt}} \leq C \bullet X_\mu \leq C \bullet X^{\text{opt}} + n\mu,$$
$$b^T y^{\text{opt}} - n\mu \leq b^T y_\mu \leq b^T y^{\text{opt}}.$$

The solutions $\{X_\mu, y_\mu, S_\mu\}$ for different values of μ trace a trajectory in the feasible region that approaches $\{X^{\text{opt}}, y^{\text{opt}}, S^{\text{opt}}\}$ as $\mu \to 0^+$, known as the *central path*. Modern primal-dual interior-point methods for SDPs like SeDuMi, SDPT3, and MOSEK use Newton's method to solve the KKT equations (19), while keeping each iterate within a *wide neighborhood* of the central path

$$\mathcal{N}_\infty^-(\gamma) \triangleq \left\{ \{X, y, S\} \in \mathcal{F} : \lambda_{\min}(XS) \geq \frac{\gamma}{n} X \bullet S \right\},$$

where $\mathcal{F}$ denotes the feasible region

$$\mathcal{F} \triangleq \left\{ \{X, y, S\} : \begin{array}{rl} A_i \bullet X & = b_i \ \forall i, \\ \sum_i y_i A_i + S & = C, \\ X, S & \text{pos. def.} \end{array} \right\}.$$

Here, $\gamma \in (0, 1)$ quantifies the "size" of the neighborhood, and is typically chosen with an aggressive value like 10^{-3}. The resulting algorithm is guarantee to converge to an approximate solution accurate to L digits after at most $O(nL)$ Newton iterations. (This can be further reduced to $O(\sqrt{n}L)$ Newton iterations by adopting a narrow neighborhood.) In practice, convergence almost always occurs with 30-50 iterations.

5.2.1. Complexity

All interior-point methods converge to L accurate digits in between $O(\sqrt{n}L)$ and $O(nL)$ Newton iterations, and practical implementations almost always occurs with tens of iterations. Accordingly, the complexity of solving (SDP) and (SDD) using an interior-point method is—up to a small multiplicative factor—the same as the cost of solving the associated Newton subproblem

$$\begin{aligned} &\text{maximize } b^T y - \frac{1}{2} \|W^{\frac{1}{2}}(S - Z)W^{\frac{1}{2}}\|_F^2 \\ &\text{subject to } \sum_{i=1}^m y_i A_i + S = C, \end{aligned} \tag{20}$$

in which $W, Z \in \mathbb{S}^n_{++}$ are used by the algorithm to approximate the log-det penalty[1]. The positive definite matrix W is known as the *scaling matrix*, and is always fully-dense.

[1]Here, we assume that primal-, dual-, or Nesterov–Todd primal-dual scalings are used. The less common H..K..M and AHO primal-dual scalings have a sightly different version of (20); see [163] for a comparison.

The standard approach found in the vast majority of interior-point solvers is to form and solve the *Hessian equation* (also known as the Schur complement equation), obtained by substituting $S = C - \sum_{i=1}^m y_i A_i$ into the objective (20) and taking first-order optimality conditions:

$$A_i \bullet \left[W \left(\sum_{j=1}^m y_j A_j \right) W \right] = \underbrace{b_i + A_i \bullet W(C - Z)W}_{r_i} \tag{21}$$

for all $i \in \{1, \ldots, m\}$. Once y is computed, the variables $S = C - \sum_{i=1}^m y_i A_i$ and $X = W(Z - S)W$ are easily recovered. Vectorizing the matrix variables allows (21) to be compactly written as $\mathbf{H}y \equiv [\mathbf{A}^T(W \otimes W)\mathbf{A}]y = r$ where $\mathbf{A} = [\operatorname{vec} A_1, \ldots, \operatorname{vec} A_m]$. It is common to solve it by forming the Hessian matrix $\mathbf{H}$ explicitly and factoring it using dense Cholesky factorization, in $O(n^3m + n^2m^2 + m^3)$ time and $\Theta(m^2 + n^2)$ memory. The overall interior-point method then has complexity between $\sim n^3$ time and $\sim n^2$ memory (for $m = O(1)$ constraints) and $\sim n^6$ time and $\sim n^4$ memory (for $m = O(n^2)$ constraints).

5.2.2. Bibliography

The modern study of interior-point methods was initiated by Karmarkar [164] and their extension to SDPs was due to Nesterov and Nemirovsky [158, Chapter 4] and Alizadeh [22]. Earlier algorithms were essentially the same as the barrier methods from the 1960s; M. Wright [165] gives an overview of this historical context. The effectiveness of these methods was explained by the seminal work of Nesterov and Nemirovski [158] on self-concordant barrier methods. For an accessible introduction to these classical results, see Boyd and Vandenberghe [8, Chapters 9 and 11]. The development of primal-dual interior-point methods began with Kojima et al. [166, 167], and was eventually extended to semidefinite programming and second-order cone programming in a unified way by Nesterov and Todd [23, 36]. For a survey on primal-dual interior-point methods for SDPs, see Sturm [37]. Today, the best interior-point solvers for SDPs are SeDuMi, SDPT3, and MOSEK; the interested reader is referred to [37, 168, 169] for their implementation details.

5.3. ADMM

One of the most successful first-order methods for SDPs has been ADMM. Part of its appeal is that it is simple and easy to implement at a large scale, and that convergence is guaranteed under very mild assumptions. Furthermore, the algorithm is often "lucky": for many large-scale SDPs, it converges at a *linear rate*—like an interior-point method—to 6+ digits of accuracy in just a few hundred iterations [170]. However, the worst-case behavior is regularly attained, particularly for SDPs that arise from convexification; thousands of iterations are required to obtain solutions of only modest accuracy [171, 172].

ADMM, or the alternating direction method of multipliers, originates from the Douglas-Rachford splitting algorithm [173] which was developed to find numerical solution to heat conduction problems. It is closely related to the augmented Lagrangian method, a popular optimization algorithm in the 1970s for constrained optimization that was historically known as "the method of multipliers" [174, 175]. Both methods begin by augmenting the dual problem (SDD), written here as a minimization

$$\begin{array}{ll} \text{minimize} & -b^T y \\ \text{subject to} & \sum_{i=1}^m y_i A_i + S = C \ , \quad S \succeq 0 \end{array} \tag{22}$$

with a quadratic penalty term that does not affect the minimum nor the minimizer

$$\begin{array}{ll} \text{minimize} & -b^T y + \frac{t}{2} \left\| \sum_{i=1}^m y_i A_i + S - C \right\|_F^2 \\ \text{subject to} & \sum_{i=1}^m y_i A_i + S = C \ , \quad S \succeq 0. \end{array} \tag{23}$$

However, the quadratic term makes (23) strongly convex (for $t > 0$ and under Assumption 1). The convex conjugate of a strongly convex function is Lipschitz smooth (see e.g. [176]), and this mean that the Lagrangian dual, written

$$\underset{X \succeq 0}{\text{maximize}} \ \left\{ \min_{y, S \succeq 0} \mathscr{L}_t(X, y, S) \right\}, \tag{24}$$

where the augmented Lagrangian $\mathscr{L}_t(X, y, S) = -b^T y + X \bullet (\sum_{i=1}^m y_i A_i + S - C) + \frac{t}{2} \left\| \sum_{i=1}^m y_i A_i + S - C \right\|_F^2$ is differentiable with a Lipschitz-continuous gradient. Essentially, adding a quadratic regularization to the dual problem (22) smoothes the corresponding primal problem (24), thereby allowing a gradient-based optimization algorithm to be effectively used for its solution.

The augmented Lagrangian algorithm is derived by applying projected gradient ascent to the maximization (24) while setting the step-size to exactly t, as in

$$X^{k+1} = \left[X^k + t \nabla_X \left\{ \min_{y, S \succeq 0} \mathscr{L}_t(X^k, y, S) \right\} \right]_+ ,$$

where $[W]_+$ denotes the projection of the matrix W onto the semidefinite cone $\mathbb{S}^n_+$, as in $[W]_+ = \arg\min_{Z \succeq 0} \|W - Z\|_F^2$. Some algebra shows that the gradient term can be evaluated by solving the inner minimization problem, and that the special step-size of t guarantees $X^k \succeq 0$ so long as $X^0 \succeq 0$. Substituting these two simplifications yields the classic form of the augmented Lagrangian sequence

$$\begin{aligned} \{y^{k+1}, S^{k+1}\} &= \arg \min_{y, S \succeq 0} \mathscr{L}_t(X^k, y, S) \\ X^{k+1} &= X^k + t \left(\sum_{i=1}^m y_i^{k+1} A_i + S^{k+1} - C \right). \end{aligned}$$

The iterates converge to the solutions of (SDP) and (SDD) for all fixed $t > 0$, and that the convergence rate is superlinear if t is allowed to increase after every iteration [177]. In practice, convergence to high accuracy is achieved in tens of iterations by picking a very large value of t.

A key difficulty of the augmented Lagrangian method is the evaluation of the joint y- and S- update step, which requires us to solve a minimization problem that is not too much easier than the original dual problem (SDD). ADMM overcomes this difficulty by adopting an alternating-directions approach, updating y while holding S fixed, then updating S using the new value of y computed, as in

$$\begin{aligned} y^{k+1} &= \arg \min_{y} \mathscr{L}_t(X^k, y, S^k) \\ S^{k+1} &= \arg \min_{S \succeq 0} \mathscr{L}_t(X^k, y^{k+1}, S) \\ X^{k+1} &= X^k + t \left(\sum_{i=1}^m y_i^{k+1} A_i + S^{k+1} - C \right). \end{aligned}$$

Here, the y- update is the unconstrained minimization of a quadratic objective, and has closed-form solution

$$y^{k+1} = (\mathbf{A}^T \mathbf{A})^{-1} \left[\frac{1}{t} \left(b - \mathbf{A}^T \operatorname{vec} X^k \right) + \mathbf{A}^T \operatorname{vec} \left(C - S^k \right) \right],$$

where $\mathbf{A} = [\operatorname{vec} A_1, \ldots, \operatorname{vec} A_m]$. Similarly, the S-update is the projection of a specific matrix matrix onto the positive-semidefinite cone

$$S^{k+1} = [D]_+ \text{ where } D = C - \sum_{i=1}^m y_i^{k+1} A_i - \frac{1}{t} X^k,$$

and also has a closed-form solution in terms of the eigenvalue decomposition

$$D = \sum_{i=1}^n d_i v_i v_i^T, \qquad [D]_+ = \sum_{i=1}^n \max\{d_i, 0\} v_i v_i^T,$$

where d_i and v_i denote eigenvalues and eigenvectors, respectively. The iterates converge towards to the solutions of (SDP) and (SDD) for all fixed $t > 0$ [170, Theorem 2], although the sequence now typically converge much more slowly. In practice, a heuristic based on balancing the primal and dual residuals seems to work very well; see [170, Section 3.2] or [4, Section 3.4.1] for its implementation details.

5.3.1. Complexity

Unfortunately, it is difficult to bound the convergence rate of ADMM. It has been shown that the sequence converges with sublinear objective error $O(1/k)$ in an ergodic sense [178], so in the worst case, the method converges to L accurate digits in $O(\exp(L))$ iterations. (This exponential factor precludes ADMM from being a polynomial-time algorithm.) In practice, ADMM often performs much better than the worst-case, converging to L accurate digits in just $O(L)$ iterations for a large array of SDP test problems [170].

The per-iteration cost of ADMM can be dominated by the y-update, due to the solution of the following system $(\mathbf{A}^T\mathbf{A})y = r$ with a different right-hand side at each iteration. The standard approach is to precompute the Cholesky factor, and to solve each instance by solving two triangular systems via forward- and back-substitution. When the matrix $\mathbf{A}$ is fully-dense, the worst-case complexity of solving $(\mathbf{A}^T\mathbf{A})y = r$ is $O(n^6)$ time and $O(n^4)$ memory.

In practice, the matrix $\mathbf{A}$ is usually large-and-sparse, and complexity of $(\mathbf{A}^T\mathbf{A})y = r$ can be dramatically reduced by exploiting sparsity. Indeed, the problem of solving a large-and-sparse symetric positive definite system with multiple right-hand sides is classical in numerical linear algebra, and the associated literature is replete. Efficiency can be substantially improved by reordering the columns of $\mathbf{A}$ using a fill-minimizing ordering like minimum degree and nested dissection [179], and by using an incomplete factorization as the preconditioner within an iterative solution algorithm like conjugate gradients [180]. The cost of solving practical instances of $(\mathbf{A}^T\mathbf{A})y = r$ can be as low as $O(m)$.

Assuming that the cost of solving $(\mathbf{A}^T\mathbf{A})y = r$ can be made negligible by exploiting sparsity in $\mathbf{A}$, the per-iteration cost is then dominated by the eigenvalue decomposition required for the S-update. Performing this step using dense linear algebra requires $\Theta(n^3)$ time and $\Theta(n^2)$ memory. For larger problems where X^{opt} is known to be low-rank, it may be possible to use low-rank linear algebra and an iterative eigendecomposition like Lanczos iterations to push the complexity figure down to as low as $O(n)$ per-iteration.

5.3.2. Bibliography

ADMM was originally proposed in the mid-1970s by Glowinski and Marrocco [181] and Gabay and Mercier [182], and was studied extensively in the context of maximal monotone operators (see [4, Section 3.5] for a summary of the historical developments). The algorithm experienced a revival in the past decade, in a large part due to the publication of a popular and influential survey by Boyd et al. [4] for applications in distributed optimization and statistical learning. The algorithm described in this subsection was first proposed by Wen, Goldfarb and Yin [170], and is one of two popular variations of ADMM specifically designed for the solution of large-scale SDPs, alongside the algorithm of O'Donoghue et. al [183].

5.4. Modified interior-point method for low-rank SDPs

A fundamental issue with standard off-the-shelf interior-point solvers is their inability to exploit problem structure to substantially reduce complexity. In this subsection, we describe a modification to the standard interior-point method that makes it substantially more efficient for large-and-sparse low-rank SDPs, for which the number of nonzeros in the data $A_1, \ldots, A_m$ is small, and $\theta \triangleq \text{rank}\{X^{\text{opt}}\}$ is known *a priori* to be very small relative to the dimensions of the problem, i.e. $\theta \ll n$. As we have previous reviewed in the previous two sections, such problems widely appear by applying convexification to problems in graph theory [71], approximation theory [21, 90], control theory [21, 184, 185], and power systems [66]. They are also the fundamental building blocks for global optimization techniques based upon polynomial sum-of-squares [131] and the generalized problem of moments [21].

To describe this modification, let us recall that modern primal-dual interior-point methods almost always converge in 30-50 iterations, and that their per-iteration cost is dominated by the solution of the Hessian equation

$$\underbrace{[\mathbf{A}^T(W \otimes W)\mathbf{A}]}_{\mathbf{H}} y = r, \tag{25}$$

in which $\mathbf{A} = [\text{vec}\, A_1, \ldots, \text{vec}\, A_m]$, and W is the positive definite scaling matrix. An important feature of interior-point methods for SDPs is that the matrix W is fully-dense, and this makes $\mathbf{H}$ fully-dense, despite any apparent sparsity in the data matrix $\mathbf{A}$. The standard approach of dense Cholesky factorization takes approximately the same amount of time and memory for sparse, low-rank problems as it does for dense, high-rank problems.

Alternatively, the Hessian equation may be solved iteratively using the preconditioned conjugate gradients (PCG) algorithm. We defer to standard texts [186] for the implementation details of PCG, and only note that at each iteration, the method requires a single matrix-vector product with the governing matrix $\mathbf{H}$, and a single *solve*[2] with the preconditioner $\tilde{\mathbf{H}}$. The key ingredient is a good preconditioner: a matrix $\tilde{\mathbf{H}}$ that is similar to $\mathbf{H}$ in a spectral sense, but is otherwise much cheaper to invert. The following iteration bound is standard; a proof can be found in standard references like [180] and [187].

Proposition 6. *Consider using preconditioned conjugate gradients to solve $\mathbf{H}y = r$, with $\tilde{\mathbf{H}}$ as preconditioner. Define $y^\star = \mathbf{H}^{-1}r$ as the exact solution, and*

$$\kappa = \lambda_{\max}(\tilde{\mathbf{H}}^{-1}\mathbf{H})/\lambda_{\min}(\tilde{\mathbf{H}}^{-1}\mathbf{H})$$

as the joint condition number. Then at most

$$i \leq \left\lceil \frac{\sqrt{\kappa}}{2} \log\left(\frac{2\sqrt{\kappa}}{\epsilon}\right) \right\rceil \quad \textit{PCG iterations}$$

are required to compute an ϵ-accurate iterate y^i satisfying $\|y^i - y^\star\| \leq \epsilon\|y^\star\|$.

A preconditioner that guarantees joint condition number $\kappa = O(1)$ was described in [30]. The preconditioner based on the insight that the scaling matrix W can be decomposed into two components,

$$W = W_0 + UU^T, \tag{26}$$

[2] i.e. matrix-vector product with the inverse $\tilde{\mathbf{H}}^{-1}$.

in which the rank of U is at most $\theta = \text{rank}\{X^{\text{opt}}\}$, and W_0 is well-conditioned, meaning that all of its eigenvalues are roughly the same value. Substituting (26) into (25) reveals the same decomposition for the Hessian matrix,

$$\mathbf{H} = \mathbf{A}^T(W_0 \otimes W_0)\mathbf{A} + \underbrace{\mathbf{A}^T(U \otimes Z)(U \otimes Z)^T\mathbf{A}}_{\mathbf{U}\mathbf{U}^T} \quad (27)$$

where Z is any matrix (not necessarily unique) satisfying $ZZ^T = 2E + UU^T$. Note that the rank of $\mathbf{U}$ is at most $n\theta$.

To develop a preconditioner based on this insight, we make the approximation $W_0 \approx \tau I$ in (27), to yield

$$\tilde{\mathbf{H}} = \tau^2\mathbf{A}^T\mathbf{A} + \mathbf{U}\mathbf{U}^T. \quad (28)$$

This dense matrix is a low-rank perturbation of the sparse matrix $\mathbf{A}^T\mathbf{A}$, and can be inverted using the Sherman-Morrison-Woodbury formula

$$\tilde{\mathbf{H}}^{-1} = (\tau^2\mathbf{A}^T\mathbf{A})^{-1}(I - \mathbf{U}\mathbf{S}^{-1}\mathbf{U}^T(\mathbf{A}^T\mathbf{A})^{-1}), \quad (29)$$

in which the Schur complement $\mathbf{S} = \tau^2 I + \mathbf{U}^T(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{U}$ can be precomputed.

Lemma 7 ([30, Lemma 7])**.** *Let $\tilde{\mathbf{H}}$ be defined in (28), and choose τ to satisfy $\lambda_{\min}(E) \le \tau \le \lambda_{\max}(E)$. Then, the joint condition number $\kappa = \lambda_{\max}(\tilde{\mathbf{H}}^{-1}\mathbf{H})/\lambda_{\min}(\tilde{\mathbf{H}}^{-1}\mathbf{H})$ is an absolute constant.*

Combining Proposition 6 and Lemma 7, we find that PCG with $\tilde{\mathbf{H}}$ as preconditioner will solve the Hessian equation to L digits of accuracy in at most $O(L)$ iterations.

5.4.1. Complexity

All interior-point methods converge to L accurate digits in between $O(\sqrt{n}L)$ and $O(nL)$ Newton iterations, and practical implementations almost always occurs with 30-50 Newton iterations. Performing each Newton iteration using the PCG procedure described above, this translates into a formal complexity bound of $O(\sqrt{n}L^2)$ to $O(nL^2)$ PCG iterations, and a practical value of between 500-1500 PCG iterations.

The per-iteration cost of PCG can be dominated by the cost of applying the Sherman–Morrison–Woodbury formula to invert the preconditioner $\tilde{\mathbf{H}}$, due to the need of repeatedly making matrix-vector products with $(\mathbf{A}^T\mathbf{A})^{-1}$. Like discussed in Section 5.3 for ADMM, the matrix $\mathbf{A}$ is usually large-and-sparse in practical applications, and standard techniques from numerical linear algebra can often substantially reduce the cost of this operation. Assuming that this matrix-vector product can be performed in $O(m)$ time and memory, [30] showed that the cost of inverting $\tilde{\mathbf{H}}$ is $O(\theta^2 n^2)$ time and memory, after an initial factorization step requiring $O(\theta^3 n^3)$ time, where $\theta = \text{rank}\{X^{\text{opt}}\}$.

Assming that the cost of applying the preconditioner $\tilde{\mathbf{H}}$ is negligible, the per-iteration cost of PCG becomes dominated by the matrix-vector with $\mathbf{H}$, which is $\Theta(n^3)$ time and $\Theta(n^2)$ memory. Assuming that θ and L are both significantly smaller than n, the formal complexity of the algorithm is $\Theta(n^{3.5})$ time and $\Theta(n^2)$ memory, and the practical complexity is closer to $\Theta(n^3)$ time.

5.4.2. Bibliography

The idea of using conjugate gradients (CG) to solve the Hessian equation dates back to the original Karmarkar interior-point method [157], and was widely used in early interior-point codes for SDPs [156, 188]. However, subsequent numerical experience [189, 190] found the approach to be ineffective: the Hessian matrix $\mathbf{H}$ becomes ill-conditioned as the interior-point iterate approaches the solution, and CG requires more and more iterations to converge. Toh and Kojima [25] were the first to develop highly effective *spectral* preconditioners, based on the same decomposition of the scaling matrix W as above. However, its use required almost as much time and memory as a single iteration of the regular interior-point method. The modified interior-point method of [30], which we had described in this subsection, makes the same idea efficient by utilizing the Sherman–Morrison–Woodbury formula.

5.5. Other specialized algorithms

This tutorial has given an overview of the interior-point method as a general-purpose algorithm for SDPs, and described two specialized structure-exploiting algorithms for large-scale SDPs in detail. Numerous other structure-exploiting algorithms also exist. In general, it is convenient to categorize them into three distinct groups:

The first group comprises first-order methods, like ADMM in Section 5.3, but also smooth gradient methods [31], conjugate gradients [25, 29, 185], augmented Lagrangian methods [27], applied either to (SDP) directly, or to the Hessian equation associated with an interior-point solution of (SDP). All of these algorithms have inexpensive per-iteration costs but a sublinear worst-case convergence rate, computing an ϵ-accurate solution in $O(1/\epsilon)$ time. They are most commonly used to solve very large-scale SDPs to modest accuracy, though in fortunate cases, they can also converge to high accuracy.

The second group comprises second-order methods that use sparsity in the data to decompose the size-n conic constraint $X \succeq 0$ into many smaller conic constraints over submatrices of X. In particular, when the matrices $C, A_1, \ldots, A_m$ share a common sparsity structure with a chordal graph with bounded treewidth τ, a technique known as *chordal decomposition* or *chordal conversion* can be used to reformulate (SDP)-(SDD) into a problem containing only size-$(\tau+1)$ semidefinite constraints [24]; see also [32]. While the technique is only applicable to chordal SDPs with bounded treewidths, it is able to reduce the cost of a size-n SDP all the way down to the cost of a size-n linear program, sometimes as low as $O(\tau^3 n)$. Indeed, chordal sparsity can be guaranteed in many impor-

tant applications [32, 85], and software exist to automate the chordal reformulation [191].

The third group comprises formulating low-rank SDPs as nonconvex optimization problems, based on the outer product factorization $X = RR^T$. The number of decision variables is dramatically reduced from $\sim n^2$ to n [26, 28], though the problem being solved is no longer convex, so only local convergence can be guaranteed. Nevertheless, time and memory requirements are substantially reduced, and these methods have been used to solve very large-scale low-rank SDPs to excellent precision; see the computation results in [26, 28].

6. Conclusion

Optimization lies at the core of classical control theory, as well as up-and-coming fields of statistical and machine learning. This tutorial paper provides an overview of conic optimization, and its application to the design, analysis, control and operation of real-world systems. In particular, we give concrete case studies on machine learning, power systems, and state estimation, as well as the abstract problems of rank minimization and quadratic optimization. We show that a wide range of nonconvex problems can be converted in a principled manner into a hierarchy of convex problems, using a range of techniques collectively known as convexification. Finally, we develop numerical algorithms to solve these convex problems in a highly efficient manner, by exploiting problem structure like sparsity and low solution rank.

7. References

[1] S. Boyd, L. El Ghaoui, E. Feron, V. Balakrishnan, Linear matrix inequalities in system and control theory, SIAM, 1994.

[2] K. Zhou, J. C. Doyle, K. Glover, et al., Robust and optimal control, Vol. 40, Prentice hall New Jersey, 1996.

[3] G. Dullerud, F. Paganini, A Course in Robust Control Theory: A Convex Approach, Springer-Verlag, 2000.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends® in Machine Learning 3 (1) (2011) 1–122.

[5] L. Ljung, System identification: Theory for the user, Prentice Hall, 1998.

[6] E. F. Camacho, C. B. Alba, Model predictive control, Springer Science & Business Media, 2013.

[7] D. P. Bertsekas, Dynamic programming and optimal control, Vol. 1, Athena scientific Belmont, MA, 1995.

[8] S. Boyd, L. Vandenberghe, Convex optimization, Cambridge university press, 2004.

[9] F. R. Bach, G. R. Lanckriet, M. I. Jordan, Multiple kernel learning, conic duality, and the SMO algorithm, in: Proc. 21st Int. Conf. Machine Learning, ACM, 2004, p. 6.

[10] P. Bühlmann, S. Van De Geer, Statistics for high-dimensional data: methods, theory and applications, Springer Science & Business Media, 2011.

[11] S. Foucart, H. Rauhut, A mathematical introduction to compressive sensing, Vol. 1, Basel: Birkhäuser, 2013.

[12] P. A. Parrilo, Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization, Ph.D. thesis, California Institute of Technology (2000).

[13] G. R. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, M. I. Jordan, Learning the kernel matrix with semidefinite programming, J. Mach. Learn. Res. 5 (2004) 27–72.

[14] K. Q. Weinberger, L. K. Saul, Unsupervised learning of image manifolds by semidefinite programming, Int. J. Comput. Vision 70 (1) (2006) 77–90.

[15] M. Conforti, G. Cornuéjols, G. Zambelli, Integer Programming, Springer International Publishing Switzerland, 2014.

[16] B. Kocuk, S. S. Dey, X. Sun, New formulation and strong misocp relaxations for AC optimal transmission switching problem, IEEE Trans. Power Syst.

[17] S. Fattahi, M. Ashraphijuo, J. Lavaei, A. Atamtürk, Conic relaxations of the unit commitment problem, Energy.

[18] P. Parrilo, Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization, Ph.D. thesis, Cal. Inst. of Tech. (May 2000).

[19] J. B. Lasserre, Global Optimization with Polynomials and the Problem of Moments 11 (2001) 796–817.

[20] M. Laurent, A Comparison of Sherali-Adams, Lovasz-Schrijver, and Lasserre Relaxations for 0-1 Programming 28 (2003) 470–496.

[21] J. B. Lasserre, Moments, Positive Polynomials and Their Applications, no. 1 in Imperial College Press Optimization Series, Imperial College Press, 2010.

[22] F. Alizadeh, Interior point methods in semidefinite programming with applications to combinatorial optimization, SIAM J. Optim. 5 (1) (1995) 13–51.

[23] Y. E. Nesterov, M. J. Todd, Self-scaled barriers and interior-point methods for convex programming, Math. Oper. Res. 22 (1) (1997) 1–42.

[24] M. Fukuda, M. Kojima, K. Murota, K. Nakata, Exploiting sparsity in semidefinite programming via matrix completion I: General framework, SIAM J. Optim. 11 (3) (2001) 647–674.

[25] K.-C. Toh, M. Kojima, Solving some large scale semidefinite programs via the conjugate residual method, SIAM J. Optim. 12 (3) (2002) 669–691.

[26] S. Burer, R. D. Monteiro, A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization, Math. Program. 95 (2) (2003) 329–357.

[27] M. Kočvara, M. Stingl, PENNON: A code for convex nonlinear and semidefinite programming, Optim. Method. Softw. 18 (3) (2003) 317–333.

[28] M. Journée, F. Bach, P.-A. Absil, R. Sepulchre, Low-rank optimization on the cone of positive semidefinite matrices, SIAM J. Optim. 20 (5) (2010) 2327–2351.

[29] X.-Y. Zhao, D. Sun, K.-C. Toh, A Newton-CG augmented lagrangian method for semidefinite programming, SIAM J. Optim. 20 (4) (2010) 1737–1765.

[30] R. Y. Zhang, J. Lavaei, Modified interior-point method for large-and-sparse low-rank semidefinite programs, in: IEEE 56th Ann. Conf. Decis. Contr. (CDC), IEEE, 2017.

[31] Y. Nesterov, Smoothing technique and its applications in semidefinite optimization, Math. Program. 110 (2) (2007) 245–259.

[32] L. Vandenberghe, M. S. Andersen, et al., Chordal graphs and semidefinite optimization, Foundations and Trends in Optimization 1 (4) (2015) 241–433.

[33] D. Henrion, A. Garulli, Positive polynomials in control, Vol. 312, Springer Science & Business Media, 2005.

[34] G. Chesi, A. Garulli, A. Tesi, A. Vicino, Homogeneous Polynomial Forms for Robustness Analysis of Uncertain Systems, Springer, 2009.

[35] G. Chesi, LMI techniques for optimization over polynomials in control: a survey, IEEE Transactions on Automatic Control 55 (11) (2010) 2500–2510.

[36] Y. E. Nesterov, M. J. Todd, Primal-dual interior-point methods for self-scaled cones, SIAM J. Optim. 8 (2) (1998) 324–364.

[37] J. F. Sturm, Implementation of interior point methods for mixed semidefinite and second order cone optimization problems, Optim. Method. Softw. 17 (6) (2002) 1105–1154.

[38] T. Kato, H. Kashima, M. Sugiyama, K. Asai, Multi-task learn-

ing via conic programming, in: Adv. Neural Inf. Process. Syst., 2008, pp. 737–744.
[39] T. Graepel, Kernel matrix completion by semidefinite programming, in: Int. Conf. Artificial Neural Networks, Springer, 2002, pp. 694–699.
[40] T. Graepel, R. Herbrich, Invariant pattern recognition by semidefinite programming machines, in: Adv. Neural Inf. Process. Syst., 2004, pp. 33–40.
[41] T. F. Coleman, Y. Li (Eds.), Large-scale numerical optimization, Vol. 46, SIAM, 1990.
[42] F. Bach, R. Jenatton, J. Mairal, G. Obozinski, Optimization with sparsity-inducing penalties, Foundations and Trends® in Machine Learning 4 (1) (2012) 1–106.
[43] S. J. Benson, Y. Ye, X. Zhang, Solving large-scale sparse semidefinite programs for combinatorial optimization, SIAM J. Optim. 10 (2) (2000) 443–461.
[44] J. Garcke, M. Griebel, M. Thess, Data mining with sparse grids, Computing 67 (3) (2001) 225–253.
[45] S. Muthukrishnan, Data streams: Algorithms and applications, Foundations and Trends® in Theoretical Computer Science 1 (2) (2005) 117–236.
[46] X. Wu, X. Zhu, G. Q. Wu, W. Ding, Data mining with big data 26 (1) (2014) 97–107.
[47] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, S. Yan, Sparse representation for computer vision and pattern recognition 98 (6) (2010) 1031–1044.
[48] L. Qiao, S. Chen, X. Tan, Sparsity preserving projections with applications to face recognition, Pattern Recognit. 43 (1) (2010) 331–341.
[49] S. Sojoudi, J. Doyle, Study of the brain functional network using synthetic data, 52nd Annu. Allerton Conf. Communication, Control, and Computing (Allerton) (2014) 350–357.
[50] E. Candes, J. Romberg, Sparsity and incoherence in compressive sampling, Inverse Prob. 23 (3) (2007) 969–985.
[51] J. Fan, J. Lv, A selective overview of variable selection in high dimensional feature space, Statistica Sinica 20 (1) (2010) 101.
[52] J. Friedman, T. Hastie, R. Tibshirani, Sparse inverse covariance estimation with the graphical lasso, Biostatistics 9 (3) (2008) 432–441.
[53] O. Banerjee, L. El Ghaoui, A. d'Aspremont, Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data, J. Mach. Learn. Res. 9 (2008) 485–516.
[54] M. Yuan, Y. Lin, Model selection and estimation in the gaussian graphical model, Biometrika 94 (1) (2007) 19–35.
[55] H. Liu, K. Roeder, L. Wasserman, Stability approach to regularization selection (stars) for high dimensional graphical models, in: Adv. Neural Inf. Process. Syst., 2010, pp. 1432–1440.
[56] N. Krämer, J. Schäfer, A.-L. Boulesteix, Regularized estimation of large-scale gene association networks using graphical gaussian models, BMC Bioinf. 10 (1) (2009) 384.
[57] P. Danaher, P. Wang, D. M. Witten, The joint graphical lasso for inverse covariance estimation across multiple classes, J. Royal Stat. Soc. B (Statistical Methodology) 76 (2) (2014) 373–397.
[58] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, P. Ravikumar, Quic: quadratic approximation for sparse inverse covariance estimation, J. Mach. Learn. Res. 15 (1) (2014) 2911–2947.
[59] S. Sojoudi, Equivalence of graphical lasso and thresholding for sparse graphs, J. Mach. Learn. Res. 17 (115) (2016) 1–21.
[60] S. Sojoudi, Graphical lasso and thresholding: Conditions for equivalence, IEEE 55th Ann. Conf. Decis. Contr. (CDC) (2016) 7042–7048.
[61] S. Fattahi, S. Sojoudi, Graphical lasso and thresholding: Equivalence and closed-form solutions, `https://arxiv.org/abs/1708.09479`.
[62] R. Mazumder, T. Hastie, Exact covariance thresholding into connected components for large-scale graphical lasso, J. Mach. Learn. Res. 13 (2012) 781–794.
[63] D. M. Witten, J. H. Friedman, N. Simon, New insights and faster computations for the graphical lasso, J. Comput. Graph. Stat. 20 (4) (2011) 892–900.
[64] D. Bienstock, A. Verma, Strong NP-hardness of AC power flows feasibility, [Online]. Available: `http://arxiv.org/pdf/1512.07315v1.pdf` (Dec. 2015).
[65] Report by Federal Energy Regulatory Commission. URL `http://www.ferc.gov/industries/electric/indus-act/market-planning/opf-papers.asp`
[66] J. Lavaei, S. H. Low, Zero duality gap in optimal power flow problem 27 (1) (2012) 92–107.
[67] J. Lavaei, B. Zhang, D. Tse, Geometry of power flows and optimization in distribution networks, IEEE Trans. Power Syst. 29 (2) (2014) 572–583.
[68] S. Sojoudi, R. Madani, J. Lavaei, Low-rank solution of convex relaxation for optimal power flow problem, IEEE SmartGridComm.
[69] S. Sojoudi, J. Lavaei, Convexification of optimal power flow problem by means of phase shifters, EEE SmartGridComm.
[70] S. Sojoudi, J. Lavaei, Physics of power networks makes hard optimization problems easy to solve, IEEE Power & Energy Society General Meeting.
[71] S. Sojoudi, J. Lavaei, Exactness of semidefinite relaxations for nonlinear optimization problems with underlying graph structure, SIAM J. Optim. 4 (2014) 1746 – 1778.
[72] R. Madani, M. Ashraphijuo, J. Lavaei, Promises of conic relaxation for contingency-constrained optimal power flow problem 31 (2) (2016) 1297–1307.
[73] R. Madani, M. Ashraphijuo, J. Lavaei, OPF Solver, `http://www.ieor.berkeley.edu/~lavaei/Software.html`.
[74] D. Molzahn, C. Josz, I. Hiskens, P. Panciatici, A Laplacian-Based Approach for Finding Near Globally Optimal Solutions to OPF Problems, IEEE Trans. Power Syst. 32 (2017) 305–315.
[75] C. Chen, A. Atamtürk, S. S. Oren, Bound tightening for the alternating current optimal power flow problem 31 (5) (2016) 3729–3736.
[76] C. Coffrin, H. Hijazi, P. Van Hentenryck, The QC Relaxation: Theoretical and Computational Results on Optimal Power Flow 31 (2016) 3008–3018.
[77] C. Coffrin, H. L. Hijazi, P. V. Hentenryck, Strengthening convex relaxations with bound tightening for power network optimization, Inter. Conf. on Principles and Practice of Constraint Prog. (2015) 39–57.
[78] B. Ghaddar, J. Marecek, M. Mevissen, Optimal power flow as a polynomial optimization problem, IEEE Trans. Power Syst. 31 (2016) 539–546.
[79] B. Kocuk, S. Dey, X. Sun, Inexactness of SDP Relaxation and Valid Inequalities for Optimal Power Flow, IEEE Trans. Power Syst. 31 (2015) 642–651.
[80] Y. Weng, M. D. Ilić, Q. Li, R. Negi, Convexification of bad data and topology error detection and identification problems in AC electric power systems, IET Gener. Transm. Distrib. 9 (16) (2015) 2760–2767.
[81] R. Madani, J. Lavaei, R. Baldick, A. Atamturk, Power system state estimation and bad data detection by means of conic relaxation, in: Hawaii International Conference on System Sciences (HICSS-50), 2017.
[82] S. Fattahi, J. Lavaei, A. Atamturk, Promises of conic relaxations in optimal transmission switching of power systems, in: IEEE 56th Ann. Conf. Decis. Contr. (CDC), 2017.
[83] K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima, K. Murota, Exploiting sparsity in semidefinite programming via matrix completion II: Implementation and numerical results, Math. Program. 95 (2) (2003) 303–327.
[84] R. Grone, C. R. Johnson, E. M. Sá, H. Wolkowicz, Positive definite completions of partial Hermitian matrices, Linear Algebra Appl. 58 (1984) 109–124.
[85] R. Madani, S. Sojoudi, G. Fazelnia, J. Lavaei, Finding low-rank solutions of sparse linear matrix inequalities using convex optimization, SIAM J. Optim. 27 (2) (2017) 725–758.
[86] C. R. Johnson, Matrix completion problems: a survey, in: Proc. Symp. Applied Mathematics, Vol. 40, 1990, pp. 171–198.
[87] E. J. Candès, B. Recht, Exact matrix completion via con-

vex optimization, Foundations of Computational mathematics 9 (6) (2009) 717–772.
[88] B. Recht, M. Fazel, P. A. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, SIAM Rev. 52 (3) (2010) 471–501.
[89] R. H. Keshavan, A. Montanari, S. Oh, Matrix completion from a few entries, IEEE Trans. Inf. Theory 56 (6) (2010) 2980–2998.
[90] E. J. Candès, T. Tao, The power of convex relaxation: Near-optimal matrix completion 56 (5) (2010) 2053–2080.
[91] M. Fazel, Matrix rank minimization with applications, Ph.D. thesis, Stanford University (2002).
[92] M. Fazel, H. Hindi, S. P. Boyd, Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices, in: Proc. 2003 American Control Conf., Vol. 3, IEEE, 2003, pp. 2156–2162.
[93] M. Laurent, A. Varvitsiotis, A new graph parameter related to bounded rank positive semidefinite matrix completions, Math. Program. 145 (1-2) (2014) 291–325.
[94] A. Beck, Y. C. Eldar, Sparsity constrained nonlinear optimization: Optimality conditions and algorithms, SIAM J. Optim. 23 (3) (2013) 1480–1509.
[95] A. Beck, Y. C. Eldar, Y. Shechtman, Nonlinear compressed sensing with application to phase retrieval, in: Global Conference on Signal and Information Processing, 2013, pp. 617–617.
[96] Y. Chen, E. Candes, Solving random quadratic systems of equations is nearly as easy as solving linear systems, in: Adv. Neural Inf. Process. Syst., 2015, pp. 739–747.
[97] E. Candès, Y. C. Eldar, T. Strohmer, V. Voroninski, Phase Retrieval via Matrix Completion, SIAM J. Imaging Sci. 6 (2013) 199—-225.
[98] E. J. Candes, X. Li, M. Soltanolkotabi, Phase retrieval via Wirtinger flow: Theory and algorithms 61 (4) (2015) 1985–2007.
[99] E. J. Candès, T. Strohmer, V. Voroninski, PhaseLift: Exact and Stable Signal Recovery from Magnitude Measurements via Convex Programming, Commun. Pure and Appl. Math. 66 (2013) 1241—1274.
[100] R. Madani, J. Lavaei, R. Baldick, Convexification of power flow equations for power systems in presence of noisy measurements, `http://www.ieor.berkeley.edu/~lavaei/SE_J_2016.pdf`.
[101] Y. Zhang, R. Madani, J. Lavaei, Conic relaxations for power system state estimation with line measurements, IEEE Trans. Control Network Syst.
[102] H. Sherali, W. Adams, A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems, SIAM J. Discrete Math. 3 (1990) 311–430.
[103] L. Lovász, A. Schrijver, Cones of matrices and set-functions and 0-1 optimization, SIAM J. Optim. 1 (1991) 166–190.
[104] J. L. Krivine, Quelques propriétés des préordres dans les anneaux commutatifs unitaires, C.R. Acad. Sci. Paris 258 (1964) 3417–3418.
[105] J. Krivine, Anneaux préordonnés, J. d'Anal. Math. 12 (1964) 307–326.
[106] G. Cassier, Problème des moments sur un compact de Rn et décomposition de polynômes à plusieurs variables, J. Funct. Anal. 58 (1984) 254–266.
[107] D. Handelman, Representing polynomials by positive linear functions on compact convex polyhedra, Pac. J. Math. 132 (1988) 35–62.
[108] G. Pólya, Uber positive Darstellung von Polynomen, Vierteljahresschrift der Naturforschenden Gesellschaft in Zürich, reprinted in: Collected Papers, Volume 2, 309–313, Cambridge: MIT Press (1974) 73 (1928) 141–145.
[109] J. Pena, J. C. Vera, L. F. Zuluaga, Positive polynomials on unbounded domains, `https://arxiv.org/pdf/1709.03435.pdf`.
[110] J. Pena, J. C. Vera, L. F. Zuluaga, Completely positive reformulations for polynomial optimization, Math. Program. Ser. B. 151 (2015) 405–431.
[111] A. A. Ahmadi, A. Majumdar, DSOS and SDSOS Optimization: More Tractable Alternatives to Sum of Squares and Semidefinite Optimization, `https://arxiv.org/pdf/1706.02586.pdf`.
[112] G. Stengle, A Nullstellensatz and a Positivstellensatz in Semialgebraic Geometry, Math. Ann. 207 (1974) 87–97.
[113] J. Lasserre, K. Toh, S. Yang, A Bounded Degree SOS Hierarchy for Polynomial Optimization, EURO J. Comp. Optim. 5 (2015) 87–117.
[114] A. A. Ahmadi, A. Majumdar, DSOS and SDSOS Optimization: LP and SOCP-based Alternatives to Sum of Squares Optimization, 48th Annu. Conf. Information Sciences and Systems (CISS).
[115] X. Kuang, B. Ghaddar, J. Naoum-Sawaya, L. F. Zuluaga, Alternative SDP and SOCP Approximations for Polynomial Optimization, `https://arxiv.org/pdf/1510.06797.pdf`.
[116] X. Kuang, B. Ghaddar, J. Naoum-Sawaya, L. F. Zuluaga, Alternative LP and SOCP Hierarchies for ACOPF Problems, IEEE Trans. Power Syst. 32 (2017) 2828–2836.
[117] D. Molzahn, I. Hiskens, Mixed SDP/SOCP Moment Relaxations of the Optimal Power Flow Problem, in: IEEE Eindhoven PowerTech, 2015.
[118] M. Carpentier, Contribution à l'Étude du Dispatching Économique, Bull. de la Soc. Fran. des Élec. 8 (1962) 431—-447.
[119] C. Josz, Counterexample to Global Convergence of DSOS and SDSOS hierarchies, `https://arxiv.org/pdf/1707.02964.pdf`.
[120] P. Parrilo, Sum of Squares Optimization in the Analysis and Synthesis of Control Systems, Slides for ACC 2006.
[121] S. Lall, Sums of Squares, Slides for EE364b.
[122] N. Z. Shor, Quadratic optimization problems, Soviet J. Comput. Systems Sci. 25 (1987) 1–11.
[123] D. Hilbert, Uber die Darstellung definiter Funktionen durch Quadrateber die darstellung definiter formen als summe von Formenquadraten, Math. Ann. 32 (1888) 342–350.
[124] N. Z. Shor, Nondifferentiable Optimization and Polynomial Problems, Kluwer, Dordrecht.
[125] C. Ferrier, Hilbert's 17th problem and best dual bounds in quadratic minimization, Kibernetika i Sistemnyi Analiz 5 (1998) 76–91.
[126] Y. Nesterov, Squared functional systems and optimization problems, High Performance Optimization, H. Frenk, K. Roos, T. Terlaky, and S. Zhang, eds., Kluwer, Dordrecht.
[127] M. Putinar, Positive Polynomials on Compact Semi-Algebraic Sets, Indiana Univ. Math. J. 42 (1993) 969–984.
[128] K. Schmüdgen, The K-Moment Problem for Semi-Algebraic Sets, Math. Ann. 289 (1991) 203–206.
[129] A. A. Ahmadi, P. A. Parrilo, Towards Scalable Algorithms with Formal Guarantees for Lyapunov Analysis of Control Systems via Algebraic Optimization, Tutorial paper for 53rd IEEE CDC.
[130] J. B. Lasserre, D. Henrion, C. Prieur, E. Trélat, Nonlinear optimal control via occupation measures and LMI relaxations, SIAM J. Control Optim. 47 (2008) 1643–1666.
[131] P. Parrilo, Semidefinite Programming Relaxations for Semialgebraic Problems, Math. Program. 96 (2003) 293–320.
[132] C. Josz, D. Henrion, Strong Duality in Lasserre's Hierarchy for Polynomial Optimization, Springer Optim. Lett.
[133] J. Nie, Optimality Conditions and Finite Convergence of Lasserre's Hierarchy, Math. Program. 146 (2014) 97–121.
[134] M. Marshall, Representation of non-negative polynomials with finitely many zeros, Annales de la Faculte des Sciences Toulouse 15 (2006) 599–609.
[135] M. Marshall, Representation of non-negative polynomials, degree bounds and applications to optimization, Canad. J. Math. 61 (2009) 205–221.
[136] R. E. Curto, L. A. Fialkow, Recursiveness, Positivity, and Truncated Moment Problems, Houston J. Of Math. 17.
[137] R. E. Curto, L. A. Fialkow, Truncated K-Moment Problems in Several Variables, J. Operator Theory 54 (2005) 189–226.
[138] C. Josz, D. K. Molzahn, Multi-ordered Lasserre hierarchy for large scale polynomial optimization in real and complex vari-

ables, https://arxiv.org/pdf/1709.04376.pdf.
[139] J. D'Angelo, M. Putinar, Polynomial Optimization on Odd-Dimensional Spheres, in: Emerging Applications of Algebraic Geometry, Springer New York, 2008.
[140] M. Schweighofer, On the complexity of Schmüdgen's Positivstellensatz, J. of Complexity 20 (2004) 529–543.
[141] E. de Klerk, M. Laurent, Error bounds for some semidefinite programming approaches to polynomial minimization on the hypercube, SIAM J. Optim. 20 (6) (2010) 3104–3120.
[142] V. Magron, Error bounds for polynomial optimization over the hypercube using putinar type representations, Springer Optim. Lett. 9 (2015) 887–895.
[143] E. D. Klerk, R. Hess, M. Laurent, Improved convergence rates for Lasserre-type hierarchies of upper bounds for box-constrained polynomial optimization, SIAM J. Optim. 27 (2017) 347–367.
[144] J. B. Lasserre, A new look at nonnegativity on closed sets and polynomial optimization, SIAM J. Optim. 21 (2011) 864–885.
[145] G. Blekherman, There are significantly more nonnegative polynomials than sums of squares, Isreal J. Math. 153 (2006) 355–380.
[146] H. Waki, S. Kim, M. Kojima, M. Muramatsu, Sums of Squares and Semidefinite Program Relaxations for Polynomial Optimization Problems with Structured Sparsity, SIAM J. Optim. 17 (1) (2006) 218–242.
[147] T. Weisser, J. Lasserre, K. Toh, Sparse-BSOS: a Bounded Degree SOS Hierarchy for Large Scale Polynomial Optimization with Sparsity, Math. Program. Comp. 5 (2017) 1–32.
[148] D. Molzahn, I. Hiskens, Sparsity-Exploiting Moment-Based Relaxations of the Optimal Power Flow Problem, IEEE Trans. Power Syst. 30 (6) (2015) 3168–3180.
[149] C. Josz, S. Fliscounakis, J. Maeght, P. Panciatici, AC Power Flow Data in MATPOWER and QCQP format: iTesla, RTE Snapshots, and PEGASE, https://arxiv.org/abs/1603.01533.
[150] C. Coffrin, D. Gordon, P. Scott, NESTA, the NICTA energy system test case archive, arXiv:1411.0359.
[151] J. Lofberg, YALMIP: A toolbox for modeling and optimization in MATLAB, in: IEEE International Symposium on Computer Aided Control Systems Design, IEEE, 2004, pp. 284–289.
[152] C. Riener, T. Theobald, L. J. Andrén, J. B. Lasserre, Exploiting Symmetries in SDP-Relaxations for Polynomial Optimization, Math. Oper. Res. 38 (1) (2013) 122–141.
[153] A. Nemirovskii, D. B. Yudin, Problem complexity and method efficiency in optimization.
[154] N. Z. Shor, Minimization methods for non-differentiable functions, Springer-Verlag, 1985.
[155] M. Grötschel, L. Lovász, A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, Combinatorica 1 (2) (1981) 169–197.
[156] L. Vandenberghe, S. Boyd, Semidefinite programming, SIAM Rev. 38 (1) (1996) 49–95.
[157] N. Karmarkar, A new polynomial-time algorithm for linear programming, in: Proc. 16th Annu. ACM Symp. Theory of computing, ACM, 1984, pp. 302–311.
[158] Y. Nesterov, A. Nemirovskii, Y. Ye, Interior-point polynomial algorithms in convex programming, Vol. 13, SIAM, 1994.
[159] F. Alizadeh, J.-P. A. Haeberly, M. L. Overton, Complementarity and nondegeneracy in semidefinite programming, Math. Program. 77 (1) (1997) 111–128.
[160] Y. Ye, M. J. Todd, S. Mizuno, An $O(\sqrt{nL})$-iteration homogeneous and self-dual linear programming algorithm, Math. Oper. Res. 19 (1) (1994) 53–67.
[161] M. S. Lobo, L. Vandenberghe, S. Boyd, H. Lebret, Applications of second-order cone programming, Linear Algebra Appl. 284 (1-3) (1998) 193–228.
[162] L. Vandenberghe, S. Boyd, S.-P. Wu, Determinant maximization with linear matrix inequality constraints, SIAM J. Matrix Anal. Appl. 19 (2) (1998) 499–533.
[163] M. Todd, K. Toh, R. Tütüncü, On the nesterov–todd direction in semidefinite programming, SIAM J. Optim. 8 (3) (1998) 769–796.
[164] N. Karmarkar, A new polynomial-time algorithm for linear programming, Combinatorica 4 (4) (1984) 373–395.
[165] M. Wright, The interior-point revolution in optimization: history, recent developments, and lasting consequences, Bull. Am. Math. Soc. 42 (1) (2005) 39–56.
[166] M. Kojima, S. Mizuno, A. Yoshise, Algorithm for linear programming, Progress in mathematical programming (1989) 29.
[167] M. Kojima, N. Megiddo, T. Noma, A. Yoshise, A unified approach to interior point algorithms for linear complementarity problems, Vol. 538, Springer Science & Business Media, 1991.
[168] R. H. Tütüncü, K.-C. Toh, M. J. Todd, Solving semidefinite-quadratic-linear programs using SDPT3, Math. Program. 95 (2) (2003) 189–217.
[169] E. D. Andersen, K. D. Andersen, The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm, in: High performance optimization, Springer, 2000, pp. 197–232.
[170] Z. Wen, D. Goldfarb, W. Yin, Alternating direction augmented lagrangian methods for semidefinite programming, Math. Program. Comp. 2 (3-4) (2010) 203–230.
[171] R. Madani, A. Kalbat, J. Lavaei, ADMM for sparse semidefinite programming with applications to optimal power flow problem, in: IEEE 54th Ann. Conf. Decis. Contr. (CDC), IEEE, 2015, pp. 5932–5939.
[172] Y. Zheng, G. Fantuzzi, A. Papachristodoulou, P. Goulart, A. Wynn, Fast ADMM for semidefinite programs with chordal sparsity, arXiv:1609.06068.
[173] J. Douglas, H. H. Rachford, On the numerical solution of heat conduction problems in two and three space variables, Trans. Am. Math. Soc.
[174] M. R. Hestenes, Multiplier and gradient methods, J. Optim. Theory Appl. 4 (5) (1969) 303–320.
[175] M. Powell, A method for non-linear constraints in minimization problems, in: R. Fletcher (Ed.), Optimization, Academic Press, 1969.
[176] Y. Nesterov, Smooth minimization of non-smooth functions, Math. Program. 103 (1) (2005) 127–152.
[177] R. T. Rockafellar, Monotone operators and the proximal point algorithm, SIAM J. Control Optim. 14 (5) (1976) 877–898.
[178] B. He, X. Yuan, On the o(1/n) convergence rate of the douglas-rachford alternating direction method, SIAM J. Numer. Anal. 50 (2) (2012) 700–709.
[179] A. George, J. W. Liu, Computer solution of large sparse positive definite systems, Prentice-Hall, 1981.
[180] Y. Saad, Iterative methods for sparse linear systems, Siam, 2003.
[181] R. Glowinski, A. Marroco, Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires, Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique 9 (2) (1975) 41–76.
[182] D. Gabay, B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximation, Computers & Mathematics with Applications 2 (1) (1976) 17–40.
[183] B. O'Donoghue, E. Chu, N. Parikh, S. Boyd, Conic optimization via operator splitting and homogeneous self-dual embedding, J. Optim. Theory Appl. 169 (3) (2016) 1042–1068.
[184] G. Valmorbida, M. Ahmadi, A. Papachristodoulou, Stability analysis for a class of partial differential equations via semidefinite programming 61 (6) (2016) 1649–1654.
[185] R. Y. Zhang, Robust stability analysis for large-scale power systems, Ph.D. thesis, Massachusetts Institute of Technology (2016).
[186] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, H. Van der Vorst, Templates for the solution of linear systems: building blocks for iterative methods, SIAM, 1994.
[187] A. Greenbaum, Iterative methods for solving linear systems, SIAM, 1997.

[188] L. Vandenberghe, S. Boyd, A primal-dual potential reduction method for problems involving matrix inequalities, Math. Program. 69 (1-3) (1995) 205–236.
[189] K. Fujisawa, M. Fukuda, M. Kojima, K. Nakata, Numerical evaluation of sdpa (semidefinite programming algorithm), in: High performance optimization, Springer, 2000, pp. 267–301.
[190] S. J. Benson, Y. Ye, DSDP3: Dual-scaling algorithm for semidefinite programming, Tech. rep., Mathematics and Computer Science Division, Argonne National Laboratory (2001).
[191] S. Kim, M. Kojima, M. Mevissen, M. Yamashita, Exploiting sparsity in linear and nonlinear matrix inequalities via positive semidefinite matrix completion, Math. Program. 129 (1) (2011) 33–68.