**Les Cahiers du GERAD** ISSN: 0711–2440

# Conic optimization

M.F. Anjos

G–2015–03

January 2015



**GERAD** HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

**Tél. : 514 340-6053**
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

# Conic optimization

**Miguel F. Anjos**

*GERAD & Canada Research Chair in Discrete Nonlinear Optimization in Engineering, Polytechnique Montréal, Montréal (Québec) Canada, H3C 3A7*

anjos@stanfordalumni.org

**January 2015**

**Les Cahiers du GERAD**

**G–2015–03**

**Abstract:** Conic optimization refers to the problem of optimizing a linear function over the intersection of an affine space and a closed convex cone. Conic optimization problems are thus a particular class of convex optimization problems. We focus particularly on the special case where the cone is chosen as the cone of positive semidefinite matrices for which the resulting optimization problem is called a semidefinite optimization problem. The class of semidefinite optimization problems includes linear optimization problems as a special case, namely when all the matrices involved are diagonal. Another special case of semidefinite optimization is second-order cone optimization that corresponds to optimizing over the second-order cone, also known as the Lorentz cone. As the case of linear optimization is discussed in detail in elsewhere in this book, we focus here on the use of other cones. Although most research has focused on the positive semidefinite cone, the second-order cone is arguably more important in practical applications, as shown for example in the chapter on Financial Enginering in this part.

This chapter provides an introduction to conic optimization, and the three subsequent chapters present applications of conic optimization in control engineering, truss topology design, and financial engineering. The reader wishing to delve deeper into the area is refereed to the books [28, 7, 2] and their extensive bibliographies.

# 1 Fundamentals

## 1.1 Semidefinite optimization

Semidefinite optimization (SDO) is concerned with the optimization of a linear (or possibly convex quadratic) function of a matrix variable subject to linear constraints on the elements of the matrix, and the additional constraint that the matrix must be positive semidefinite (PSD), i.e., that all its eigenvalues be non-negative. Although it only became widely known and used since the mid-1990s, SDO has been studied (under different names) since at least the 1940s, The importance of SDO grew immensely during the 1990s because polynomial-time interior-point methods for linear programming were extended to solve SDO problems in accordance with the theory of Nesterov and Nemirovskii [19]. For example, this opened up the possibility to use SDO to design polynomial-time approximation schemes for hard combinatorial problems, and also to solve well-known problems in control theory, among many other applications.

SDO problems are an important class of optimization problems for several reasons. First, SDO problems are solvable in polynomial time. This means that any problem that can be expressed using SDP is also solvable in polynomial time. Second, SDO problems can be solved efficiently in practice. This can be done by using one of the software packages available, or alternatively by implementing a suitable algorithm. Third, SDO can be used to obtain tight approximations for hard problems in integer and global optimization.

SDO has a number of similarities with linear optimization (LO). Like LO problems, SDO problems also come in pairs. One of the problems is referred to as the *primal* problem, and the second one is the *dual* problem. Either problem can be chosen as the primal, since the two problems are dual to each other. The most common standard formulation of SDO is as follows:

$$\begin{array}{lll} \text{(P)} & \inf & \langle C, X\rangle \\ & \text{s.t.} & \langle A_i, X\rangle = b_i, i = 1,\ldots,m \\ & & X \succeq 0 \end{array} \qquad \begin{array}{lll} \text{(D)} & \sup & b^T y \\ & \text{s.t.} & \sum\limits_{i=1}^{m} y_i A_i + S = C \\ & & S \succeq 0 \end{array} \tag{1}$$

where (P) denotes the primal problem, and (D) the dual problem; the variables $X$ and $S$ are in $\mathcal{S}^n$, the space of $n \times n$ real symmetric matrices; $X \succeq 0$ denotes that the matrix $X$ is positive semidefinite; the data matrices $A_i$ and $C$ may be assumed to be symmetric without loss of generality; and $b \in \Re^m$ and $y \in \Re^m$ are column vectors. We use the scalar product between two matrices in $\mathcal{S}^n$ defined as

$$\langle R, S\rangle := \operatorname{trace}(RS) = \sum_{i=1}^{n}\sum_{j=1}^{n} R_{i,j} S_{i,j}$$

where trace $M$ denotes the trace of the square matrix $M$, which is the sum of the diagonal elements of $M$. It is normally assumed that, without loss of generality, the matrices $A_i$, $i = 1,\ldots,m$, are linearly independent. When it is known from the context that the optimal values are attained, it is common to replace inf by min and sup by max.

An example of a SDO problem that comes up in multiple contexts is:

$$\begin{array}{ll} \min & \langle C, X\rangle \\ \text{s.t.} & X_{ii} = 1, \quad i = 1,\ldots,n \\ & X \succeq 0. \end{array} \tag{2}$$

The feasible set of (2) is well-known as the *elliptope* and is unique for each $n \geq 2$: the elliptope of dimension $\binom{n}{2}$ is the set of all symmetric matrices $n \times n$ that are PSD and have ones on the diagonal. For instance, if $n = 3$ the corresponding elliptope is:

$$\left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \mathbb{R}^3 : \begin{pmatrix} 1 & x & y \\ x & 1 & z \\ y & z & 1 \end{pmatrix} \succeq 0 \right\}.$$

Figure 1 shows its visualization in $\mathbb{R}^3$. The four vertices of this set correspond to the matrices:

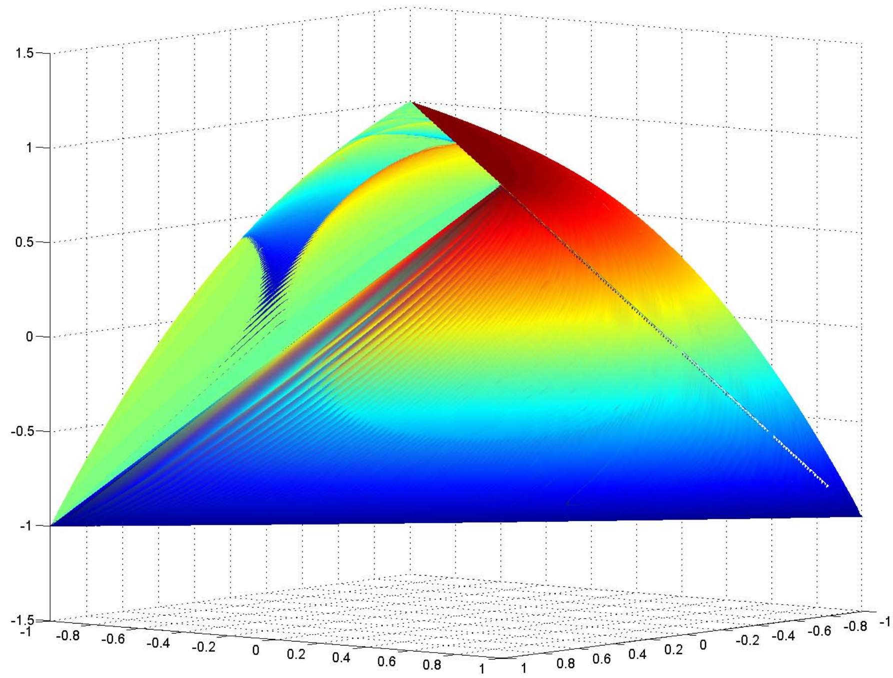


Figure 1: Visualization of the 3-dimensional elliptope

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix}, \text{ and } \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{pmatrix}.$$

It is important to note that unlike for LO, the feasible sets of SDO problems have extreme points that are *not* vertices. For the elliptope in $\mathbb{R}^3$ for example, the matrix

$$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 1 \end{pmatrix}$$

is not a vertex, but it is an extreme point of the elliptope since it cannot be expressed as a convex combination of the four vertices.

The dual SDO problem in (1) can equivalently be written without using the dual variable $S$:

$$\begin{array}{ll} \sup & b^T y \\ \text{s.t.} & C - \sum_{i=1}^{m} y_i A_i \succeq 0. \end{array}$$

The resulting constraint is called a *linear matrix inequality (LMI)*. The inequality is interpreted according to the partial order induced on $\mathcal{S}^n$ by the positive semidefinite cone. This is called the Löwner partial order, and is defined as follows:

$$A \succeq B \quad \Leftrightarrow \quad A - B \succeq 0.$$

### 1.2 Second-order cone optimization

An $(n+1)$-dimensional second-order cone (SOC) is the set of all vectors $(x_0, x_1, \ldots, x_n) \in \mathbb{R}^{n+1}$ that satisfy $x_0 \geq \sqrt{x_1^2 + \ldots + x_n^2}$. An SOC constraint is equivalent to a positive semidefinite constraint with the following special structure:

$$\begin{pmatrix} x_0 & 0 & \cdots & x_1 \\ 0 & x_0 & \cdots & x_2 \\ \vdots & \ddots & \cdots & \vdots \\ x_1 & x_2 & \cdots & x_0 \end{pmatrix} \succeq 0.$$

Second-order cone optimization (SOCO) is the problem of optimizing a linear (or possibly convex quadratic) function over the SOC, with additional linear constraints on the variables if desired.

An equivalent expression of an SOC is in the form of a rotated quadratic cone:

$$\{ (x_0, x_1, \ldots, x_n) \in \mathbb{R}^{n+1} \,|\, x_0 x_1 \geq x_2^2 + \ldots + x_n^2 \}.$$

Mathematically this is a rotation of the quadratic cone as defined above, but for modeling purposes the rotated form is often more convenient.

### 1.3 General conic optimization and duality

Recall that $\mathcal{K} \subset \mathcal{S}^n$ is called a *cone* if it is closed under positive scalar multiplication:

$$\lambda X \in \mathcal{K} \text{ whenever } X \in \mathcal{K} \text{ and } \lambda > 0.$$

For any convex set $\mathcal{K}$ in a normed vector space $V$, the *dual cone* $\mathcal{K}^*$ is defined as

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

It immediately follows that $\mathcal{K}^*$ is a closed convex cone regardless of the choice of $\mathcal{K}$. Using this definition, the self-duality of the non-negative orthant in LO is obvious, that of the SOC follows by the Cauchy-Schwarz inequality, and that of the PSD cone by Fejer's Theorem, see e.g. [14, Corollary 7.5.4].

While the aforementioned cones are still the most commonly used in applications, there is a growing interest in other convex cones that are not self-dual. These include the completely positive cone:

$$\mathcal{C} := \{X \in \mathcal{S}^n : X = \sum_{i=1}^{k} v_i v_i^T, v_i \geq 0\}$$

and its dual cone, the copositive cone:

$$\mathcal{C}^* := \{X \in \mathcal{S}^n : v^T X v \geq 0 \text{ for all } v \geq 0\}.$$

The corresponding area of copositive optimization is a rapidly expanding and fertile field of research with a diversity of formulations using these cones being used in a variety of applications. We refer the reader to the recent survey of Bomze [5] for the state-of-the-art in this dynamic area of research.

The growing interest in using different cones justifies the definition of a general conic optimization problem:

$$\begin{array}{llll} (\mathrm{P}_{\mathcal{K}}) & \inf & \langle C, X \rangle & \\ & \text{s.t.} & \langle A_i, X \rangle = b_i, i = 1, \ldots, m & \\ & & \mathcal{K} \succeq 0 & \end{array} \qquad \begin{array}{lll} (\mathrm{D}_{\mathcal{K}}) & \sup & b^T y \\ & \text{s.t.} & \sum_{i=1}^{m} y_i A_i + S = C \\ & & \mathcal{K}^* \succeq 0 \end{array} \tag{3}$$

where the only difference with (1) is the general choice of primal cone $\mathcal{K}$ and dual cone $\mathcal{K}^*$. For this general problem we have (as in LO) a *weak duality* theorem ensuring that the dual (respectively primal) problem always provides a global upper (respectively lower) bound on the optimal value of the primal (respectively dual):

**Theorem 1** *If $\tilde{X}$ is feasible for $(P_{\mathcal{K}})$ and $\tilde{y}, \tilde{Z}$ for $(D_{\mathcal{K}})$, then $\langle C, \tilde{X}\rangle \leq b^T\tilde{y}$.*

Moreover because of the nonlinear conic constraint, strong duality, i.e., equality of the bounds at the optimum, does not automatically hold for general conic optimization. We illustrate this with following two SDO examples from [28, pp. 71-72].

**Example 1 (Positive duality gap)** *In LO, if both (P) and (D) are feasible, then there is no duality gap. This may fail for SDO problems. For example,*

$$\begin{array}{llll} (P) & \max & -ax_{11} \\ & \text{s.t.} & x_{11} + 2x_{23} = 1 \\ & & x_{22} = 0 \\ & & X \succeq 0. \end{array} \qquad \begin{array}{lll} (D) & \min & -y_2 \\ & \text{s.t.} & \begin{pmatrix} y_2 - a & 0 & 0 \\ 0 & y_1 & y_2 \\ 0 & y_2 & 0 \end{pmatrix} \preceq 0 \end{array}$$

*It is easy to see that (P) has optimal objective value $-a$, while (D) has $0$.*

**Example 2 (Weak infeasibility)** *Even if there is no duality gap at optimality, the optimal value may not be attained for (P) or (D). Consider the primal-dual pair*

$$\begin{array}{lll} (P) & \sup & 2x_{12} \\ & \text{s.t.} & \begin{pmatrix} 1 & x_{12} \\ x_{12} & 0 \end{pmatrix} \succeq 0 \end{array} \qquad \begin{array}{lll} (D) & \inf & y_1 \\ & \text{s.t.} & \begin{pmatrix} y_1 & -1 \\ -1 & y_2 \end{pmatrix} \succeq 0 \end{array}$$

*and observe that the optimal objective value $0$ is attained for (P), but not for (D).*

We can avoid these difficulties by requiring that (P) and (D) satisfy a *constraint qualification* (CQ). This is a standard technique in nonlinear optimization whose purpose is to ensure the existence of Lagrange multipliers at optimality. These multipliers are an optimal solution for the dual problem, and thus the constraint qualification ensures that strong duality holds: it is possible to achieve primal and dual feasibility with no duality gap.

The most commonly used CQ is Slater's CQ:

**Definition 1** *Slater's CQ holds if both primal and dual have feasible positive definite matrices.*

We then have the following result.

**Theorem 2** *Under Slater's CQ, both primal and dual have optimal solutions, and the duality gap is zero at optimality.*

Slater's CQ is usually easy to verify for an SDO problem since it suffices to exhibit for each of (P) and (D) a feasible matrix which is positive definite, i.e., all its eigenvalues are strictly greater than zero. For example, for the SDO problem (2), it is obvious that the $n \times n$ identity matrix is positive definite and feasible. Rewriting (2) in the form

$$\begin{array}{ll} \min & \langle C, X\rangle \\ \text{s.t.} & \langle e_i e_i^T, X\rangle = 1, \ i = 1, \ldots, n, \\ & X \succeq 0, \end{array} \tag{4}$$

where $e_i$ denotes a vector of length $n$ with all zeros except for a 1 in the $\mathrm{i}^{\mathrm{th}}$ component, we deduce that its dual problem is

$$\begin{array}{ll} \min & \langle e, y\rangle \\ \text{s.t.} & \sum\limits_{i=1}^{m} y_i e_i e_i^T + S = C, \\ & S \succeq 0, \end{array} \tag{5}$$

where $e$ denotes the vector of all ones. To verify Slater's CQ for this problem, it suffices to choose $\bar{y}$ sufficiently large so that

$$\bar{y}_i > \sum_{j=1}^{n} C_{ij} \text{ for each } i = 1, \ldots, n.$$

It then follows that $S = \mathrm{Diag}\,(\bar{y}) - Q$ is positive definite (by diagonal dominance, see e.g. [14, Theorem 6.1.10]) and hence feasible for (5).

### 1.4 Polynomial-time solvability

One important reason for the great attention given to the PSD cone and its special cases above is that they are the only cones for which polynomial-time interior-point algorithms to solve (1) exist. This is not due to self-duality alone but rather because the polynomial-time solvability of a conic optimization problem requires the existence of a computable self-concordant barrier function for the cone at hand. Although such a function, called the Universal Barrier Function, exists for a large variety of convex cones, it is very hard to compute in general. Efficient self-concordant barriers exist only for the so-called symmetric cones, and the PSD cone is essentially the most general class of symmetric cones. We refer the reader to [19] for the theoretical aspects of polynomial-time algorithms for convex optimization. In the remainder of this chapter, we shall use the term "symmetric cones" to refer to the three types of symmetric cones commonly used in practice: the non-negative orthant, the SOC, and the PSD cone.

From the weak duality theorem (1) of SDO, we have that the duality gap equals

$$\langle C, \tilde{X}\rangle - b^T\tilde{y} = \langle \tilde{S}, \tilde{X}\rangle \geq 0.$$

Since both $X$ and $S$ are PSD, $\langle X, S\rangle = 0$ implies $XS = SX = 0$, and we obtain the well-known KKT sufficient optimality conditions:

$$\begin{array}{ll} A_i \bullet X = b_i, i = 1, \ldots, m, \text{ and } X \succeq 0 & \text{(primal feasibility)} \\ Z + \sum_{i=1}^{m} y_i A_i = C, \text{ and } Z \succeq 0 & \text{(dual feasibility)} \\ XZ = 0 & \text{(complementarity)} \end{array}$$

If Slater's CQ holds, they are also necessary for optimality.

The KKT optimality conditions can be used as the starting point for implementing interior-point algorithms to solve conic optimization problems. Excellent presentations of these algorithms are given in [22, 29] among others. Extensive research and computational developments led to the availability of excellent implementations of such algorithms that perform extremely well on even large-scale linear and SOC problems. Large-scale SDO problems however remain challenging for interior-point methods, and thus solvers based on alternative techniques have been developed.

We do not present here the principles behind algorithms for solving conic optimization problems. Instead we present in Section 2 the state-of-the-art with respect to computational solvers available for broad use. The reader may find descriptions of the algorithms implemented by each solver in the references of Section 2, and consult their bibliographies for references providing the theory behind the algorithms.

## 2 Software for conic optimization

We divide the solvers available into three groups: those that handle the three types of symmetric cones, those that only recognize linear and SOC constraints, and those that focus exclusively on solving SDO problems. Because non-negativity and SOC constraints can be cast as SD constrants, it is possible to handle these constraints using the SDO-specific solvers; however they will in many cases be less efficient than the first two groups of solvers because they do not exploit the simpler structure of the other cones.

For reasons of brevity, we mention only some of the most commonly used software packages. Our intention is not to provide a catalog of all available tools but rather to provide the reader with an overview of the software available. Unless specified otherwise, all the software mentioned here is available for free via the given URLs (at the time of publication).

A good point of entry for new users of conic optimization are the following Matlab-based packages that can solve small- to medium-scale problems combining the three types of symmetric cones:

- SeDuMi [23] is hosted and maintained at `http://sedumi.ie.lehigh.edu`. Although the core code has not changed much since 2001, it remains a highly popular and efficient code.

- SDPT3 [24, 25] is hosted at `http://www.math.nus.edu.sg/~mattohkc/sdpt3.html`. This software is also popular and fairly well tested, and it is actively maintained. The current version 4.0 dates from 2009, but see also SDPNAL below.

Two commercial interior-point solvers that can handle all symmetric cones are MOSEK and LOQO, both of which have Matlab interfaces:

- MOSEK [1] is a solver for general convex optimization problems, specifically based on optimization over symmetric cones. It also offers the capability to solve mixed-integer versions of some of these problems. Free academic licenses are available. We refer the reader to `https://mosek.com/products/mosek` for more information.
- LOQO [26, 27] is a non-linear optimization solver that can handle a large variety of problems, convex or non-convex, constrained or unconstrained. Specific routines are provided to solve SOCO and SDO problems. We refer the reader to `http://www.princeton.edu/~rvdb/loqomenu.html` for more information.

Other interior-point solvers that focus exclusively on SDO include the following:

- SDPA [30] is a family of solvers: `http://sdpa.sourceforge.net` based on the original algorithm described in [31]. Among the various codes in this family are a version that exploits sparsity in the data, another that exploits structure through a matrix completion technique, and three high-precision arithmetic versions. It offers Matlab and Python interfaces.
- CSDP [6] is written in C and designed to be used as a callable subroutine. It also makes effective use of sparsity in the data. Python and R interfaces are available. It is now a COIN-OR project: `https://projects.coin-or.org/Csdp`.
- DSDP [3] is an open source implementation a dual-scaling algorithm that exploits low-rank structure and sparsity in the data, and requires relatively little memory [4]. A Matlab interface is available. The latest version is hosted at `http://www.mcs.anl.gov/hs/software/DSDP`.

All of these codes have parallel versions as well.

There are two quantities that are critical to determine the "size" of an SDO problem for computational purposes: the size of the matrix ($n$) and the number of linear constraints ($m$), both as defined in (1). While it is widely accepted that interior-point solvers are generally very efficient and robust for moderate values of $m$ and $n$, for SDO problems with $m$ large (say greater than 10000) and $n$ moderate (say up to 5000), the limitations of interior-point approaches are severe due to the need to form and factor the so-called Schur complement matrix of size $m \times m$.

The limitations of interior-point methods motivated the development of SDO solvers that are based on other algorithmic frameworks. The software packages arising from this research include:

- PENOPT is a commercial family of solvers based on a generalized augmented Lagrangian method. Its PENNON solver [15, 16] is applicable to general non-linear optimization problems, and is particularly aimed at large-scale problems with sparsity in the data. The PENSDP variant solves SDO problems, and other variants target bilinear matrix inequalities and free material optimization. PENLAB is a free open source Matlab-based version of PENNON. We refer the reader to `http://www.penopt.com` for more information.
- SDPLR uses a first-order non-linear optimization algorithm that is based on the idea of low-rank factorization of the matrix variable in SDO as described in [8]. The source code and documentation are available at `http://dollar.biz.uiowa.edu/~sburer/projects.html`.
- ConicBundle is a library of C/C++ subroutines that implements a bundle method for minimizing the sum of convex functions that are given by first order oracles or arise from the Lagrangean relaxation of conic optimization problems [12, 11]. This library includes code specific for optimizing over the symmetric cones. The source code and documentation are available at `https://www-user.tu-chemnitz.de/~helmberg/ConicBundle`.

- SDPNAL is a recently developed SDO solver based on a semi-smooth Newton-CG augmented Lagrangian method [32]. It is designed to solve large-scale SDO problems with It uses the same input format as SDPT3, thus users of the latter can easily use SDPNAL as well.

Many other codes have been developed for specific applications in areas such as combinatorial optimization, image processing, etc. We do not cover them here for lack of space and instead refer the reader to the Handbook [2].

We close this section with two notes of practical importance for using conic optimization. The first is that many of the solvers mentioned here can be accessed via the NEOS (Network-Enabled Optimization System) Server `http://www.neos-server.org/neos/solvers/index.html`. This is a free internet-based service for solving most standard forms of optimization problems, including conic problems.

The second is that there are modelling tools that can be of great help in the application of conic optimization. Two examples of such tools are YALMIP and CVX, both of which are Matlab-based:

- YALMIP [18] is a modelling language that supports a variety of optimization problems, including the three types of symmetric cones and their mixed-integer forms, as well as related classes of problems such as SOS problems and robust optimization.
- CVX [10] supports modeling using a particular paradigm called disciplined convex programming [9] and its mixed-integer variant. We do not explain this paradigm here except to mention that it imposes a set of conventions to follow when constructing problems, and that while these conventions do not limit generality, they support an automated solution method based on the use of symmetric cones.

## 3 Polynomial optimization

Until the mid-1990s the standard techniques to handle models with non-linearities were to either use a general non-linear optimization algorithm or to devise a suitable LO approximation. While they have a strong theoretical basis and there exists reliable mature software to solve them, one main weakness of non-linear optimization algorithms is that the optimal solution returned by the algorithm is inescapably determined by the initial point chosen (either by the user or the software implementation) to initialize the algorithm. Hence in the absence of favourable structural properties such a convexity or a priori knowledge to guide the choice of the starting point, it is not possible for non-linear optimization approaches to guarantee convergence to the global optimum in general.

The LO-based approaches can provide guarantees of global optimality under certain conditions, and while there are well-known linearization techniques that have been highly successful in practice, their performance varies dramatically from one application to another.

The development of SDO provided a new means to linearize quadratic problems, and more generally polynomial optimization problems (POPs). A POP is a mathematical optimization problem whose objective and constraints are multivariate polynomials. Polynomial optimization generalizes several special cases that have been thoroughly studied in optimization, including mixed binary linear optimization, convex and non-convex quadratic optimization, and linear complementarity problems.

The general approach to solve POPs is based on sum-of-squares (SOS) certificates of non-negativity for multivariate polynomials. Following the pioneering work of Lasserre [17] and Parrilo [20], this approach builds a hierarchy of relaxations leading to solving a sequence of SDO problems. Under mild conditions, these relaxations provide bounds that converge to the optimal value of the original POP.

A SOS certificate is a sufficient condition for a given polynomial to be non-negative. The idea is that a polynomial that is a sum of squares of (lower degree) polynomials is always non-negative. For example, the polynomial $2x^4 + y^4 - 2x^3y - x^2y^2$ is a non-negative polynomial for all values of $x$ and $y$ because

$$2x^4 + y^4 - 2x^3y - x^2y^2 = (x^2 - xy)^2 + (y^2 - x^2).$$

It is however not true that all non-negative polynomials can be expressed as a sum of squares. A famous counterexample is the Motzkin polynomial $x^4y^2 + x^2y^4?3x^2y^2 + 1$ that is non-negative on $\mathbb{R}^2$ but cannot be expressed as a sum of squares.

The connection between sums of squares and SDO is that checking whether a given polynomial is a sum of squares is equivalent to checking the feasibility of a certain SDO problem. For our example above, to verify that our polynomial is non-negative we need to check that the equation

$$2x^4 + y^4 - 2x^3y - x^2y^2 = \left( \begin{array}{c} x^2 \\ xy \\ y^2 \end{array} \right)^T M \left( \begin{array}{c} x^2 \\ xy \\ y^2 \end{array} \right)$$

has a solution with $M$ positive semidefinite. One possible choice is

$$M = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

hence the polynomial is non-negative for all values of $x$ and $y$. We do not expand further on the rich and deep theory of polynomial optimization but rather refer the reader to the Handbook [2] and the references therein.

For a practical point of view, it is important to know that software exists to automatically build and solve the hierarchy of semidefinite relaxations. Both GloptiPoly [13] and SOSTools [21] are two Matlab-based software packages that provide this functionality, and they can be used in conjunction with several of the solvers presented in Section 2. This theory is finding its way into different areas of applications, including finance, control theory, and signal processing. An application of this approach to nonlinear optimal control is given in one of the accompanying chapters.

## References

[1] E.D. Andersen and K.D. Andersen. The MOSEK interior point optimizer for linear programming: An implementation of the homogeneous algorithm. In High performance optimization, pages 197–232. Springer, 2000.

[2] M.F. Anjos and J.B. Lasserre, editors. Handbook on Semidefinite, Conic and Polynomial Optimization. International Series in Operations Research & Management Science. Springer-Verlag, 2011.

[3] S.J. Benson and Y. Ye. Algorithm 875: Dsdp5—software for semidefinite programming. ACM Transactions on Mathematical Software (TOMS), 34(3):16, 2008.

[4] S.J. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. SIAM J. Optim., 10(2):443–461 (electronic), 2000.

[5] I.M Bomze. Copositive optimization–recent developments and applications. European Journal of Operational Research, 216(3):509–520, 2012.

[6] B. Borchers. CSDP, a c library for semidefinite programming. Optimization methods and Software, 11(1-4):613–623, 1999.

[7] S. Boyd and L. Vandenberghe. Convex optimization. Cambridge University Press, Cambridge, 2004.

[8] S. Burer and R.D.C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. Mathematical Programming, 95(2):329–357, 2003.

[9] M. Grant, S. Boyd, and Y. Ye. Disciplined convex programming. Springer, 2006.

[10] M. Grant, S. Boyd, and Y. Ye. Cvx: Matlab software for disciplined convex programming, version 2.0 beta. Recent Advances in Learning and Control, pages 95–110, 2012.

[11] C. Helmberg and K.C. Kiwiel. A spectral bundle method with bounds. Math. Program., 93(2, Ser. A):173–194, 2002.

[12] C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. SIAM J. Optim., 10(3):673–696 (electronic), 2000.

[13] D. Henrion, J.B. Lasserre, and J. Löfberg. Gloptipoly 3: moments, optimization and semidefinite programming. Optimization Methods & Software, 24(4-5):761–779, 2009.

[14] R.A. Horn and C.R. Johnson. Matrix Analysis. Cambridge University Press, Cambridge, 1990.

[15] M. Kočvara and M. Stingl. Pennon: A code for convex nonlinear and semidefinite programming. Optimization methods and software, 18(3):317–333, 2003.

[16] M. Kočvara and M. Stingl. Pennon: software for linear and nonlinear matrix inequalities. In M.F. Anjos and J.B. Lasserre, editors, Handbook of Semidefinite, Conic and Polynomial Optimization: Theory, Algorithms, Software and Applications, International Series in Operations Research and Management Science, pages 755–791. Springer, New York, 2011.

[17] J.B. Lasserre. Global optimization with polynomials and the problem of moments. SIAM J. Optim., 11(3):796–817 (electronic), 2000/01.

[18] J. Lofberg. Yalmip: A toolbox for modeling and optimization in matlab. In Computer Aided Control Systems Design, 2004 IEEE International Symposium on, pages 284–289. IEEE, 2004.

[19] Yu.E. Nesterov and A. Nemirovskii. Interior-Point Polynomial Algorithms in Convex Programming. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.

[20] P.A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. Math. Program., 96(2, Ser. B):293–320, 2003.

[21] S. Prajna, A. Papachristodoulou, and P.A. Parrilo. Introducing sostools: A general purpose sum of squares programming solver. In Proceedings of the 41st IEEE Conference on Decision and Control, volume 1, pages 741–746. IEEE, 2002.

[22] C. Roos, T. Terlaky, and J.-Ph. Vial. Interior Point Methods for Linear Optimization. Springer, 2006.

[23] J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optim. Methods Softw., 11/12(1-4):625–653, 1999.

[24] K.-C. Toh, M.J. Todd, and R.H. Tütüncü. On the implementation and usage of SDPT3 – a Matlab software package for semidefinite-quadratic-linear programming, version 4.0. In M.F. Anjos and J.B. Lasserre, editors, Handbook of Semidefinite, Conic and Polynomial Optimization: Theory, Algorithms, Software and Applications, International Series in Operations Research and Management Science, pages 715–754. Springer, New York, 2011.

[25] R.H. Tütüncü, K.-C. Toh, and M.J. Todd. Solving semidefinite-quadratic-linear programs using sdpt3. Mathematical programming, 95(2):189–217, 2003.

[26] Robert J. Vanderbei. LOQO: an interior point code for quadratic programming. Optim. Methods Softw., 11/12(1-4):451–484, 1999.

[27] Robert J. Vanderbei. LOQO user's manual—version 3.10. Optim. Methods Softw., 11/12(1-4):485–514, 1999.

[28] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. Handbook of Semidefinite Programming. Kluwer Academic Publishers, Boston, MA, 2000.

[29] S.J. Wright. Primal-Dual Interior-Point Methods. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[30] M. Yamashita, K. Fujisawa, M. Fukuda, K. Kobayashi, K. Nakata, and M. Nakata. Latest developments in the sdpa family for solving large-scale sdps. In M.F. Anjos and J.B. Lasserre, editors, Handbook of Semidefinite, Conic and Polynomial Optimization: Theory, Algorithms, Software and Applications, International Series in Operations Research and Management Science, pages 687–713. Springer, New York, 2011.

[31] M. Yamashita, K. Fujisawa, and M. Kojima. Implementation and evaluation of sdpa 6.0 (semidefinite programming algorithm 6.0). Optimization Methods and Software, 18(4):491–505, 2003.

[32] X.-Y. Zhao, D. Sun, and K.-C. Toh. A newton-cg augmented lagrangian method for semidefinite programming. SIAM Journal on Optimization, 20(4):1737–1765, 2010.