

# A Guide to Conic Optimisation and its Applications

Adam N. Letchford\*      Andrew J. Parkes†

To Appear in *RAIRO-OR*

## Abstract

Most OR academics and practitioners are familiar with linear programming (LP) and its applications. Many are however unaware of *conic optimisation*, which is a powerful generalisation of LP, with a prodigious array of important real-life applications. In this invited paper, we give a gentle introduction to conic optimisation, followed by a survey of applications in OR and related areas. Along the way, we try to help the reader develop insight into the strengths and limitations of conic optimisation as a tool for solving real-life problems.

**Key Words:** conic optimisation, second-order cone programming, semidefinite programming.

## 1 Introduction

Most OR students, academics and practitioners are familiar with *linear programming* (LP). For many problems arising in industry and elsewhere, LP is an attractive option, due to its simplicity, the ease of doing sensitivity analysis, the existence of effective algorithms, and, perhaps most importantly, the availability of good software. With modern software packages such as CPLEX, Gurobi or Xpress, it is now possible to routinely solve LP instances with thousands of variables and/or constraints to proven optimality. These software packages can also cope with integer-constrained variables.

Of course, in real-world applications, one often encounters problems that have *non-linear* aspects. Sometimes, one can construct good linear approximations to such problems, perhaps with the help of binary variables, and thereby make them amenable to solution with the above-mentioned packages. In other cases, however, the nonlinearity is significant and must be faced head-on. For such cases, various *non-linear programming* (NLP) algorithms have been developed (see, e.g., [13, 14]).

---

\*Department of Management Science, Lancaster University, United Kingdom.  
E-mail: [A.N.Letchford@lancaster.ac.uk](mailto:A.N.Letchford@lancaster.ac.uk)

†School of Computer Science, University of Nottingham, United Kingdom.  
E-mail: [andrew.parkes@nottingham.ac.uk](mailto:andrew.parkes@nottingham.ac.uk)

The purpose of this paper is to introduce a very important special case of NLP called *conic optimisation* (CO). Despite having a very special structure, CO is remarkably powerful. It has a prodigious array of important real-life applications, not only in OR, but also in related areas, such as statistics, computer science, engineering and finance. Moreover, CO inherits some of the nice features enjoyed by LP, such as the existence of efficient (polynomial-time) algorithms, a well-developed duality theory, and the availability of good software.

Apart from LP, the most important special cases of CO are *second order cone programming* (SOCP) and *semidefinite programming* (SDP). There already exist several good surveys on both SOCP (e.g., [2, 53, 62]) and SDP (e.g., [43, 93, 94, 96]). There also exists an excellent monograph on CO in general (Nesterov & Nemirovsky [73]). It is fair to say, however, that most of these works assume that the reader is a nonlinear programming expert. This can make them rather inaccessible to the OR generalist.

In this work, we assume only minimal knowledge of LP, geometry and linear algebra. Moreover, we place an emphasis on applications in OR and related areas, covering, e.g., inventory control, facility location, portfolio optimisation, problems involving binary variables, and various methods for optimising under uncertainty, such as mean-variance, chance-constrained and robust optimisation. Along the way, we try to help the reader develop insight into the strengths and limitations of CO as a tool for modelling and solving real-life problems.

The paper is structured as follows. In Sect. 2, we define CO and present some basic theory. In Sect. 3, we cover algorithms and complexity. In Sect. 4, we review the main applications of SOCP in OR and related fields. In Sect. 5, we do the same for SDP. In Sect. 6, we list some of the available software packages, and in Sect. 7, we make some concluding remarks.

Throughout the paper,  $\mathbb{R}_+$ ,  $\mathbb{R}_+^n$  and  $\mathcal{S}^n$  denote the non-negative reals, the real vectors with  $n$  non-negative components, and the real symmetric matrices of order  $n$ , respectively. Moreover, given a vector  $x \in \mathbb{R}^n$ , we let  $\|x\|_1$  and  $\|x\|_2$  denote  $\sum_{i=1}^n |x_i|$  and  $\sqrt{\sum_{i=1}^n x_i^2}$ , respectively.

## 2 Definitions and Basic Theory

In this section, we define CO formally and present the minimal amount of theory and notation needed to understand how it works. We start by defining *cones* in Subsect. 2.1. We define CO itself in Subsect. 2.2. We then present elementary duality theory in Subsect. 2.3.

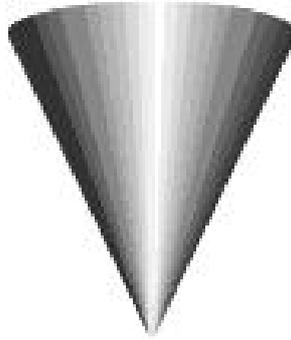


Figure 1: An idealised ice-cream cone.

## 2.1 Cones

If a non-mathematician hears the word *cone*, the thing most likely to come to mind is either an ice-cream cone or a traffic cone. An idealised version of an ice-cream cone is displayed in Fig. 1. The idealised version is supposed to extend upward forever. One can check that it corresponds to the following set:

$$\left\{ x \in \mathbb{R}^3 : x_3 \geq \sqrt{x_1^2 + x_2^2} \right\}.$$

The ancient Greek astronomer Apollonius of Perga (c. 262 BC – c. 190 BC) discovered that, if one “slices” the idealised ice-cream cone at various angles, one can obtain interesting convex shapes; see Fig. 2. This already gives a hint as to why cones could be of relevance to optimisation: whereas LP forces us to optimise over polyhedra, we can optimise over various non-polyhedral convex sets by “slicing” cones in various ways. This enables us to model and solve a variety of important nonlinear problems, as we will see in Sections 4 and 5.

Mathematically speaking, a cone is a set of points (in some underlying space, such as Euclidean space) with the following property: if a point  $x$  belongs to the cone, then so does the point  $\lambda x$  for any  $\lambda \in \mathbb{R}_+$ . Note that cones can be extremely complicated. Indeed, consider an *arbitrary* set  $S \subset \mathbb{R}^n$ . The set

$$C(S) = \left\{ (x, \lambda) \in \mathbb{R}^n \times \mathbb{R}_+ : x = \lambda x' \text{ for some } x' \in S \right\}$$

is a cone, and it is “just as complicated” as  $S$  itself. Indeed, if we intersect  $C(S)$  with the hyperplane defined by the equation  $\lambda = 1$ , the resulting “slice” is  $S$  (or, more precisely, an embedding of  $S$  into a space of higher dimension).

If one wishes to say anything useful about cones, then, one must restrict attention to cones with a special structure. In the optimisation context, one

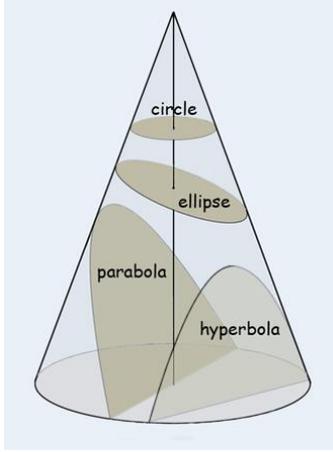


Figure 2: The conic sections (taken from Seligman [88]).

is typically interested in so-called *proper* cones. A cone  $C$  is called proper if it is:

- convex: if  $x^1, x^2 \in C$ , then  $\mu x^1 + (1 - \mu)x^2 \in C$  for all  $\mu \in [0, 1]$ ;
- closed:  $C$  contains all of its limit points;
- full-dimensional: there is no hyperplane containing  $C$ ;
- pointed: if  $x \in C$ , then  $-x \notin C$ .

One can check that the idealised ice-cream cone is proper. The following five proper cones arise frequently in various contexts:

- the *non-negative* cone, which is simply  $\mathbb{R}_+$ ;
- the *second-order cone* of order  $n$ , which is

$$\{(x, t) \in \mathbb{R}^n \times \mathbb{R}_+ : t \geq \|x\|_2\};$$

- the *positive semidefinite* (psd) cone of order  $n$ , which is

$$\{X \in \mathcal{S}^n : v^T X v \geq 0 (\forall v \in \mathbb{R}^n)\}; \quad (1)$$

the *copositive* cone of order  $n$  (first defined by Motzkin [71]), which is

$$\{X \in \mathcal{S}^n : v^T X v \geq 0 (\forall v \in \mathbb{R}_+^n)\};$$

- the *completely positive* cone of order  $n$  (first defined by Hall [39]), which is

$$\{X \in \mathcal{S}^n : X = A^T A \text{ for some real non-negative matrix } A\}.$$

(There are several other cones of interest to optimisation, such as the correlation cone [26], the  $p$ th-order cone [32], the exponential cone [31] and the relative entropy cone [21], but we do not give details, for the sake of brevity.)

The second-order cone is a natural generalisation of the ice-cream cone to higher dimensions. It is sometimes called the *Lorentz cone*, after the Dutch physicist Hendrik Lorentz (1853–1928). (Indeed, for those familiar with special relativity, the second-order cone with  $n = 3$  is the forward light cone of the origin, where  $x$  represents space and  $t$  represents time.)

The psd cone of order  $n$  is usually denoted by  $\mathcal{S}_+^n$ . It can be defined in many different ways. It is known (see, e.g., Horn [47]) that a matrix  $X \in \mathcal{S}^n$  belongs to  $\mathcal{S}_+^n$  if and only if any of the following (equivalent) conditions hold:

- the quadratic function  $f(v) = v^T X v$  is non-negative for all  $v \in \mathbb{R}^n$  (this is just definition (1));
- the same function  $f(v)$  is convex;
- the region  $\{v \in \mathbb{R}^n : v^T X v \leq 1\}$  is an ellipsoid;
- all eigenvalues of  $X$  are non-negative;
- all principal submatrices of  $X$  have non-negative determinants;
- there exists a lower-triangular matrix  $A \in \mathbb{R}^{n \times n}$  such that  $X = A^T A$  (Cholesky factorisation);
- there exist vectors  $u^1, \dots, u^n \in \mathbb{R}^n$  such that  $X_{ij} = u^i \cdot u^j$  for all  $i, j$  (Gram representation);
- $X$  can be written as a non-negative linear combination of symmetric rank-1 real matrices of the form  $vv^T$ .

From these definitions, one can see that, for a given value of  $n$ , the completely positive cone is contained in  $\mathcal{S}_+^n$ , which in turn is contained in the copositive cone.

To make the above ideas more concrete, we give some examples. The matrix  $\begin{pmatrix} 5 & 1 \\ 1 & 1 \end{pmatrix}$  is completely positive, since it has the factorisation

$$\begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 1 & 1 \end{pmatrix}.$$

The matrix  $\begin{pmatrix} 5 & -1 \\ -1 & 1 \end{pmatrix}$  is psd, since it has the factorisation:

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

It is not however completely positive, since it contains negative entries. Finally, the matrix  $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$  is copositive, since  $v^T X v = 2v_1 v_2 \geq 0$  for all  $v \in \mathbb{R}_+^2$ . It is however not psd, since

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

and therefore it has  $-1$  as an eigenvalue.

## 2.2 Conic optimisation

A *conic programme* or *conic optimisation problem* is a problem that can be modelled as the problem of optimising a linear function over the intersection of a hyperplane and a proper cone. That is, a CO problem can be written in the form

$$\sup \{ c^T x : Ax = b, x \in C \}, \quad (2)$$

where  $c \in \mathbb{Z}^n$ ,  $b \in \mathbb{Z}^m$ ,  $A \in \mathbb{Z}^{m \times n}$  and  $C \subset \mathbb{R}^n$  is a proper cone. (We have to use supremum rather than maximum here, for technical reasons. See the last example in this subsection.)

Actually, as it stands, the definition (2) is too general, since it includes  $\mathcal{NP}$ -hard problems as special cases (see the next section). Nesterov & Nemirovsky [73] showed that particularly effective solution methods could be devised for CO when the proper cones in question are *symmetric*. For brevity, we do not define symmetric cones here. Instead, we just remark that most symmetric cones of interest can be constructed using three basic building blocks; namely, the non-negative, second order and psd cones mentioned in the previous subsection. More specifically:

- The product of  $n$  non-negatives cones is  $\mathbb{R}_+^n$ . This is a symmetric cone, and it is the cone used in LP.
- The product of a finite number of second-order cones is symmetric. Cones of this type are used in *second-order cone programming* or SOCP (sometimes also called *conic quadratic programming* or CQP).
- The product of a finite number of psd cones is also symmetric. Cones of this type are used in *semidefinite programming* (SDP).

It might seem at first that restricting attention to products of cones is a big limitation, since it implies that each cone must involve a different set of variables. However, this apparent limitation can be overcome by “splitting” variables into two or more copies. For example, if we wish to impose the second-order conic constraints  $t \geq \|(x/y)\|_2$  and  $s \geq \|(x/z)\|_2$  simultaneously, where the vector  $x$  is involved in both constraints, we replace the vector

$x$  with two vectors, say  $x'$  and  $x''$ , add the constraints  $t \geq \|(x'_y)\|_2$  and  $s \geq \|(x''_z)\|_2$ , and add the (linear) constraint  $x' = x''$ .

It turns out that SDP generalises SOCP, which in turn generalises LP. Indeed, the second-order conic constraint  $t \geq \|x\|_2$  is equivalent to

$$\begin{pmatrix} t & x^T \\ x & tI_n \end{pmatrix} \in \mathcal{S}_+^{n+1},$$

where  $I_n$  is the identity matrix of order  $n$ ; and the non-negativity constraint  $t \geq 0$  is implied by  $t \geq \|x\|_2$  together with  $x_i = 0$  for  $i = 1, \dots, n$ . (Geometrically speaking, the non-negative cone is a “slice” of the second-order cone, which is in turn a “slice” of the psd cone.)

In the case of LP and SOCP, it is usual to view the variables as being arranged in a vector, which is usually denoted by  $x$ . In the case of SDP, however, it is usual to view them as being arranged in a square symmetric matrix, which is usually denoted by  $X$ . Accordingly, people sometimes refer to *vector variables* and *matrix variables*. When  $X$  is a matrix variable, the cone  $C$  is to be thought of as a subset of  $\mathcal{S}^n$  rather than  $\mathbb{R}^n$ .

To help the reader, we now give a couple of examples.

**Example 1:** Consider the following SOCP:

$$\sup \left\{ x_1 : x_1 - x_2 = 0, x_3 + x_4 = 2, x_3 \geq \sqrt{x_1^2 + x_2^2}, x_4 \geq 0 \right\}.$$

The largest value that  $x_3$  can take is 2, which means that  $x_1^2 + x_2^2$  cannot exceed 2. The optimal solution is therefore  $x_1^* = x_2^* = \sqrt{2}$ ,  $x_3^* = 2$  and  $x_4^* = 0$ , with a profit of  $\sqrt{2}$ .  $\square$

**Example 2:** Consider the following SDP:

$$\sup \left\{ X_{12} + X_{21} : X_{11} = 1, X_{22} = 3, X = \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix} \in \mathcal{S}_+^2 \right\}.$$

Note that the determinant of  $X$  is  $X_{11}X_{22} - X_{12}X_{21}$ , and this must be non-negative. Since  $X_{11}X_{22}$  must equal 3, and  $X_{12}$  must equal  $X_{21}$ , the profit is maximised by setting  $X_{12}$  and  $X_{21}$  to  $\sqrt{3}$ . This yields a profit of  $2\sqrt{3}$ .  $\square$

The above examples show that an SOCP or SDP can have a unique optimal solution in which one or more variables take *irrational* values. The following example shows that it is also possible for the supremum not to be attainable in an SOCP:

**Example 3:** Consider the following SOCP:

$$\sup \left\{ x_1 - x_3 : x_2 = 1, x_3 \geq \sqrt{x_1^2 + x_2^2} \right\}.$$

This is equivalent to:

$$\sup \left\{ x_1 - \sqrt{x_1^2 + 1} : x_1 \in \mathbb{R} \right\}.$$

We can bring the profit arbitrarily close to 0 (by making  $x_1$  arbitrarily large), but we cannot actually reach 0.  $\square$

In a similar way, it can be shown that the supremum may not be attainable in an SDP.

### 2.3 Duality

There is an elegant duality theory for CO, which can be viewed as a generalisation of LP duality. First, assume that we are working with a vector variable  $x \in \mathbb{R}^n$ . Associated with any cone  $C \subset \mathbb{R}^n$  we can define the *dual* (a.k.a. *polar*) cone

$$C^* := \{x \in \mathbb{R}^n : y^T x \geq 0 (\forall y \in C)\}.$$

If we are dealing with a matrix variable  $X \in \mathcal{S}^n$  instead, and a cone  $C \subset \mathcal{S}^n$ , then the definition of the polar cone must be modified slightly to:

$$C^* := \{X \in \mathcal{S}^n : X \bullet Y \geq 0 (\forall Y \in C)\},$$

where  $X \bullet Y$  denotes the matrix inner product  $\sum_{i=1}^n \sum_{j=1}^n X_{ij} Y_{ij}$ , which is also equal to  $\text{tr}(Y^T X)$ .

It is easy to show that (a) the dual of a proper cone is proper, (b) if a cone is proper, then it is the dual of its dual, (c) the dual of the completely positive cone is the copositive cone (and vice-versa), and (d) the non-negative, second-order and psd cones are *self-dual*. (This is a property shared by all symmetric cones.)

The following fact can be shown, e.g., by an application of Lagrangian relaxation:

**Theorem 1 (Weak Duality for CO)** *For any proper cone  $C \subset \mathbb{R}^n$ , and any  $c \in \mathbb{R}^n$ ,  $b \in \mathbb{R}^m$  and  $A \in \mathbb{R}^{m \times n}$ , we have:*

$$\sup \{c^T x : Ax = b, x \in C\} \leq \inf \{b^T y : A^T y - c \in C^*, y \in \mathbb{R}^m\}.$$

Specialised to the case of LP, Theorem 1 reduces to the well-known fact that

$$\max \{c^T x : Ax = b, x \in \mathbb{R}_+^n\} \leq \min \{b^T y : A^T y \geq c, y \in \mathbb{R}^m\}.$$

In that case, we have strong duality, i.e., the inequality can be changed to an equation. For CO in general, however, strong duality is not guaranteed. Fortunately, it is guaranteed to hold under certain conditions that commonly

occur in practice. For example, it holds if the *Slater condition* is satisfied, i.e., if there is a feasible  $x$  that lies in the interior of  $C$  (see, e.g., [14, 73]).

Even when duality is strong, however, some care must be taken when interpreting the dual. For example, although the optimal dual solution  $y^*$  provides meaningful dual prices, the components of the vector  $A^T y^* - c$  are not always meaningful “reduced costs” (see, e.g., Helmberg [42]). Moreover, concepts such as degeneracy, dual degeneracy and complementary slackness must be handled carefully (e.g., Alizadeh *et al.* [3]).

### 3 Algorithms and Complexity

Now we turn our attention to algorithms for solving CO problems. To begin, it is helpful to recall the following facts about LP:

1. The *simplex method* (Dantzig [22, 23]) is typically very fast in practice, but takes exponential time in the worst case.
2. The *ellipsoid method* (Khachiyan [50]) runs in polynomial time in the worst case, but is too slow to be of practical use.
3. There exist various *interior-point methods* (IPMs) that run in polynomial time in the worst case, and tend to be reasonably fast in practice (see, e.g., Gondzio [36]).

The situation for CO is a bit more complicated:

1. There is no known effective analogue of the simplex method for general CO, nor indeed for SOCP or SDP in particular.
2. The ellipsoid method can solve CO in polynomial time (Grötschel *et al.* [38]), but only to *fixed precision*, and only when two technical conditions are met: (i) it must be possible to test in polynomial time whether a given rational point lies in the given cone, and (ii) the feasible region must be explicitly bounded (e.g., by the addition of a constraint which forces the solution to lie inside a ball of known centre and radius). Unfortunately, as in the case of LP, this result is of little use in practice.
3. There exist IPMs that can solve SOCPs and SDPs in polynomial time, but again, only to fixed precision and only when the feasible region is bounded (Nesterov & Nemirovsky [73]). The best of these IPMs are now fast enough to be of practical use.

To see why COs can only be solved to fixed precision, recall from Subsect. 2.2 that even SOCPs and SDPs can have irrational solutions. Such solutions cannot be represented exactly with a finite number of bits. (For a discussion

on how one might represent solutions exactly as *algebraic numbers*, see Nie *et al.* [75].) One consequence of this limited precision is that even testing feasibility of an SOCP or SDP is a non-trivial problem, if one wants a precise answer. In fact, it is not known whether it can be done in polynomial time; see Ramana [84] for a discussion.

The other technical condition mentioned above is that the feasible region must be explicitly bounded. To see why, consider the following constraints:

$$x_3 - x_2 = 1, x_4 - 2x_2 = 1, x_3 \geq \sqrt{x_1^2 + x_2^2}.$$

These can be shown to imply  $x_4 \geq x_1^2$ . Chaining such constraints together, we can get  $x_7 \geq x_4^2$ ,  $x_{10} \geq x_7^2$ , and so on. At the end, we get  $x_{3p+1} \geq x_1^{(2^p)}$ . If we add the constraint  $x_1 = 2$ , we find that  $x_{3p+1} \geq 2^{(2^p)}$ . So there exist SOCPs such that it takes an exponential number of bits to represent any feasible solution. An analogous example for SDPs is given in Alizadeh [1]. Requiring the feasible region to lie inside a ball eliminates such “pathological” SOCP and SDP instances.

As for the question of testing membership of a cone, testing whether a given  $x^* \in \mathbb{Q}$  lies in the non-negative cone is trivial, and so is testing whether a given rational point  $(x^*, t^*)$  lies in the second-order cone. Less obviously, one can also check whether a given rational matrix  $X^* \in \mathcal{S}^n$  belongs to  $\mathcal{S}_+^n$  in polynomial time, via a modified form of Gaussian elimination [38]. (In practice, one can just compute the minimum eigenvalue of  $X^*$ , to some desired precision, and check whether it is non-negative.)

On the other hand, testing whether a rational matrix is completely positive is  $\mathcal{NP}$ -hard [72]. (By duality, testing copositivity is co- $\mathcal{NP}$ -hard.) In fact, Burer [16] showed that any mixed 0-1 linear or quadratic program can be transformed into a completely positive program, i.e., a conic optimisation problem over the completely positive cone. As a result, completely positive programming is  $\mathcal{NP}$ -hard in the strong sense. (By duality, copositive programming is co- $\mathcal{NP}$ -hard in the strong sense.) For a survey of results on completely positive and copositive programming, see Burer [17].

At the end of the previous subsection, we mentioned that dual information for SOCP or SDP, such as reduced costs and dual prices, must be interpreted with care. The lack of a simplex method leads to some other issues. For example:

- It can be hard to exploit sparsity in the constraint matrix (e.g., Benson *et al.* [10]).
- It is not trivial to re-optimize SOCPs and SDPs efficiently after, e.g., adding a constraint or variable (e.g., [69, 92]). In other words, it is difficult to do ‘warm starts’.

Work on these issues is ongoing within the CO community.

We now wish to mention two other remarkable papers. Ben-Tal & Nemirovski [12] proved that one can “simulate” an SOCP with  $n$  variables to arbitrary precision  $\epsilon$  using an LP with  $\mathcal{O}(n \log(1/\epsilon))$  variables. This means that one can solve an SOCP approximately by solving a single LP of reasonable size. Conversely, Braun *et al.* [15] proved that one cannot simulate SDPs by LPs or SOCPs in an analogous way, thereby proving that SDP is, in some sense, fundamentally more powerful than LP or SOCP.

Finally, we mention that there exist several other algorithms for solving SDPs, such as the spectral bundle method (Helmberg & Rendl [45]), the boundary point method (Povh *et al.* [82]), augmented Lagrangian methods [67, 98], cutting-plane methods (Krishnan & Mitchell [52]), and a method that works directly with the Cholesky factorisation (Burer & Monteiro [19]).

## 4 Applications of SOCP

In this section, we give some examples of the kinds of problems that can be tackled using SOCP. For additional examples, see, e.g., [2, 53, 62, 73].

### 4.1 Problems involving convex quadratic functions

SOCP includes as a special case all convex nonlinear programs involving quadratic functions. For brevity, we give just two examples, one from finance and one from statistics.

In the famous portfolio selection model of Markowitz [68], there are  $n$  stocks, for which we have a vector  $r \in \mathbb{R}^n$  of *expected returns* and a psd matrix  $Q \in \mathbb{R}^{n \times n}$  of *covariances*. We have a vector  $x \in \mathbb{R}_+^n$  of decision variables, where  $x_i$  represents the proportion invested in stock  $i$ . The expected return of a portfolio  $x$  is then  $r^T x$  and the variance (a measure of risk) is  $x^T Q x$ . Minimising risk subject to a lower bound  $L$  on the expected return is then equivalent to the following convex quadratic program:

$$\min \left\{ x^T Q x : r^T x \geq L, \|x\|_1 = 1, x \in \mathbb{R}_+^n \right\}.$$

If we compute the Cholesky factorisation  $Q = A^T A$ , then the objective function is equivalent to  $\|Ax\|_2^2$ . So we can reformulate the problem as:

$$\min \left\{ t : t \geq \|y\|_2, y = Ax, r^T x \geq L, \|x\|_1 = 1, x \in \mathbb{R}_+^n, y \in \mathbb{R}^n, t \in \mathbb{R}_+ \right\}.$$

This is an SOCP. Similarly, the problem of maximising expected return subject to an upper bound  $U$  on the variance can be modelled as the SOCP:

$$\max \left\{ r^T x : t \geq \|y\|_2, y = Ax, t \leq \sqrt{U}, \|x\|_1 = 1, x \in \mathbb{R}_+^n, y \in \mathbb{R}^n, t \in \mathbb{R}_+ \right\}.$$

In statistical estimation, one often encounters problems of the form:

$$\min \left\{ \sum_{j=1}^m w_j \|y^j - x\|_2^2 : x \in S \right\},$$

where

- $y^1, \dots, y^m$  are observations of a random  $n$ -vector with unknown mean;
- $x$  is an estimate of the mean (to be determined);
- $S \subset \mathbb{R}^n$  is a convex set representing some prior information on  $x$ ;
- $w_j$  is the weight (importance) given to the  $j$ th observation;
- the objective is to minimise a weighted sum of squared residuals.

By introducing a new variable  $z \in \mathbb{R}_+$  and new vectors of variables  $u \in \mathbb{R}_+^m$  and  $t^1, \dots, t^m \in \mathbb{R}^m$ , we can reformulate the estimation problem as:

$$\begin{aligned} \min \quad & z \\ \text{s.t.} \quad & z \geq \|u\|_2 \\ & u_j \geq \|t^j\|_2 \quad (j = 1, \dots, m) \\ & t^j = \sqrt{w_j}(y^j - x) \quad (j = 1, \dots, m) \\ & x \in S \\ & (z, u, t) \in \mathbb{R}_+^{1+m+m \times n}. \end{aligned}$$

If  $S$  is a polyhedron, the reformulated problem is an SOCP. If  $S$  is defined by a mixture of linear and convex quadratic constraints, the problem can be easily converted into an SOCP.

## 4.2 Problems involving hyperbolic functions

Now recall from Subsect. 2.1 that the *hyperbola* is a cross-section of the ice-cream cone. More precisely, the convex set

$$\{(x, t) \in \mathbb{R}_+^2 \times \mathbb{R} : t \geq \|x\|_2, x_2 = 1\}$$

is easily shown to be an affine image of the convex set

$$\{(x, y) \in \mathbb{R}_+^2 : y \geq 1/x\},$$

the boundary of which is a hyperbola. For this reason, one can easily force the feasible region to be a hyperbola via SOCP.

Note that the function  $y \geq 1/x$  is equivalent to  $xy \geq 1$ . More generally, a *hyperbolic function* is any function of the form  $t_1 t_2 \geq \|Ax\|^2$ , where  $t_1, t_2 \in$

$\mathbb{R}_+$  and  $x \in \mathbb{R}^n$  are variables, and  $A$  is a real matrix with  $n$  columns. It is shown in [62] that any convex nonlinear program involving a combination of linear, convex quadratic and hyperbolic functions can be converted to an SOCP. (The feasible region of a constraint involving a hyperbolic function is sometimes called a *hyperboloid*.)

Here is a simple example of an OR problem that involves hyperbolic functions. The classic inventory control model of Harris [40] is

$$\min \{hx/2 + cd/x : x \in \mathbb{R}_+\},$$

where  $x$  is the order quantity (to be determined),  $h$  is the annual cost of holding one unit in stock,  $c$  is the charge for a delivery, and  $d$  is the annual demand. Harris solved this problem by calculus, and the optimal value of  $x$  is the well-known *economic order quantity* or EOQ. Ziegler [99] considered the following multi-item extension of this model:

$$\begin{aligned} \min \quad & \sum_{i=1}^n (h_i x_i / 2 + c_i d_i / x_i) \\ \text{s.t.} \quad & \sum_{i=1}^n b_i x_i \leq b_0 \\ & \ell_i \leq x_i \leq u_i \quad (i = 1, \dots, n). \end{aligned}$$

Here, for a given product  $i$ ,  $x_i$  represent the order quantity,  $h_i$  is the annual holding cost,  $c_i$  is the delivery charge,  $d_i$  is the annual demand,  $b_i$  is the space occupied by one unit, and  $\ell_i$  and  $u_i$  are lower and upper bounds on the order quantity. The constant  $b_0$  represents the storage space available.

Kuo & Mittelmann [53] showed that this problem can be transformed into an SOCP as follows. Define new variables  $s_i$  and  $t_i$  satisfying:

$$\begin{aligned} x_i &= (s_i - t_i) / 2 \\ 1/x_i &= (s_i + t_i) / 2. \end{aligned}$$

Then model the problem as:

$$\begin{aligned} \min \quad & \sum_{i=1}^n ((h_i/2 + c_i d_i) s_i + (h_i/2 - c_i d_i) t_i) \\ \text{s.t.} \quad & \sum_{i=1}^n b_i (s_i - t_i) \leq 2b_0 \\ & 2\ell_i \leq s_i - t_i \leq 2u_i \quad (i = 1, \dots, n) \\ & s_i \geq \sqrt{t_i^2 + 2} \quad (i = 1, \dots, n) \\ & s, t \in \mathbb{R}^n. \end{aligned}$$

### 4.3 Problems involving norms

SOCP can also be used to model problems involving Euclidean or other norms. For brevity, we give three examples.

The first example concerns facility location. Suppose we have  $n$  facilities in the plane and wish to locate a new facility so as to minimise the sum of the weighted distances from the existing facilities. This can be modelled as:

$$\begin{aligned} \min \quad & \sum_{i=1}^n w_i \sqrt{(x_i - \tilde{x})^2 + (y_i - \tilde{y})^2} \\ \text{s.t.} \quad & (\tilde{x}, \tilde{y}) \in \mathbb{R}^2, \end{aligned}$$

where  $(x_i, y_i)$  are the co-ordinates of the  $i$ th existing facility, and  $(\tilde{x}, \tilde{y})$  are the co-ordinates of the new facility. (This problem is sometimes called the *Fermat-Weber* problem.) Defining new variables  $d_1, \dots, d_n$ , representing the Euclidean distance from the new facility to each of the existing facilities, the problem can be converted into the following SOCP:

$$\begin{aligned} \min \quad & \sum_{i=1}^n w_i d_i \\ \text{s.t.} \quad & d_i \geq \sqrt{u_i^2 + v_i^2} \quad (i = 1, \dots, n) \\ & u_i = x_i - \tilde{x} \quad (i = 1, \dots, n) \\ & v_i = y_i - \tilde{y} \quad (i = 1, \dots, n) \\ & (\tilde{x}, \tilde{y}) \in \mathbb{R}^2 \\ & u, v, d \in \mathbb{R}^n. \end{aligned}$$

Moreover, SOCP can also easily handle extensions of this problem in which there are linear, convex quadratic and/or hyperbolic constraints on the co-ordinates  $\tilde{x}$  and  $\tilde{y}$  and/or the distances  $d_i$ .

Our second example is a modified version of the Markowitz portfolio selection model (see Subsect. 4.1). Suppose we wish to maximise expected return subject to a *chance constraint*, which states that the probability of the return not exceeding some quantity  $\alpha \in \mathbb{R}$  must be less than some small quantity  $\beta > 0$ . (For example, we may require that there is only a 0.1% chance of losing one million euros or more.) The chance constraint can be written as:

$$r^T x + \Phi^{-1}(\beta) \sqrt{x^T Q x} \geq \alpha,$$

where  $\Phi$  is the Cumulative Distribution Function of the Normal distribution. Using again the Cholesky factorisation  $Q = AA^T$ , the constraint can be written as:

$$r^T x + \Phi^{-1}(\beta) \|Ax\|_2 \geq \alpha.$$

To handle this with SOCP, we just add new variables  $y \in \mathbb{R}^n$  and  $z \in \mathbb{R}$ , and add the constraints:

$$\begin{aligned} y &= Ax \\ z &\geq \|y\|_2 \\ r^T x + \Phi^{-1}(\beta) z &\geq \alpha. \end{aligned}$$

(Actually, this transformation only works when  $\Phi^{-1}(\beta) < 0$ . Fortunately, this is always the case in practice, since  $\beta$  is always less than  $1/2$ .)

Our third and final example comes from *robust optimisation*. Consider the LP

$$\min \{c^T x : Ax \leq b, x \in \mathbb{R}_+^n\},$$

and suppose that the precise values of the components of the matrix  $A$  are uncertain. Ben-Tal & Nemirovski [11] suggest writing the LP as:

$$\min \{c^T x : a_i^T x \leq b_i \ (i = 1, \dots, m), x \in \mathbb{R}_+^n\},$$

and then considering the case in which, for  $i = 1, \dots, m$ , the vector  $a_i$  is known to lie inside the *ellipsoid*

$$\{\hat{a}_i + Q_i u : \|u\|_2 \leq 1\},$$

where the vectors  $\hat{a}_i$  and the matrices  $Q_i$  are known. They then show that the problem of minimising the *worst-case* cost, the so-called *robust counterpart* of the LP, can be formulated as:

$$\min \{c^T x : \hat{a}_i^T x + \|Q_i x\|_2 \leq b_i \ (i = 1, \dots, m), x \in \mathbb{R}_+^n\}.$$

This is again easy to handle via SOCP.

## 5 Applications of SDP

Although SOCP is a useful modelling tool, SDP is much more powerful. In this section, we give just a few examples of the kinds of problems that can be tackled using SDP. For additional examples, see, e.g., [4, 43, 73, 93, 94, 96, 97].

### 5.1 Problems involving special types of matrices

*Covariance* and *correlation* matrices play a fundamental role in statistics, probability and (as we saw in Subsect. 4.1 and 4.3) finance. It is well known that a real symmetric matrix is a covariance matrix if and only if it is psd, and a correlation matrix if and only if, in addition, it has 1s on the main diagonal. This means that one can use SDP to solve various optimisation problems involving such matrices, such as:

- The *positive semidefinite matrix completion problem*: given a matrix with missing entries, check if it can be completed to a covariance (or correlation) matrix (see, e.g., Johnson [49]).
- The *nearest correlation matrix problem*: given a matrix that is not a correlation matrix, find a correlation matrix that is as close as possible, where “close” is measured according to, e.g., an  $L_1$ ,  $L_2$  or  $L_\infty$  norm (see, e.g., Higham [46]).

Somewhat less well known are *Euclidean distance* (ED) matrices. A matrix  $M \in \mathcal{S}^n$  is an ED matrix if and only if there exist points  $x^1, \dots, x^n \in \mathbb{R}^n$  such that, for  $i, j = 1, \dots, n$ ,  $M_{ij} = \|x_i - x_j\|_2^2$ , i.e., the square of the Euclidean distance between  $x_i$  and  $x_j$ . A classic result of Schoenberg [87] states that a given matrix  $M \in \mathcal{S}^n$  is an ED matrix if and only if the symmetric matrix  $M'$  belongs to  $\mathcal{S}_+^{n-1}$ , where:

$$\begin{aligned} M'_{ii} &= M_{i,n} & (i = 1, \dots, n-1) \\ M'_{ij} &= \frac{1}{2}(M_{in} + M_{jn} - M_{ij}) & (1 \leq i < j \leq n-1) \end{aligned}$$

Laurent [55] observes that one can therefore also use SDP to solve various optimisation problems involving ED matrices. This includes problems in, e.g., computational biology (such as molecular conformation problems) and engineering (such as wireless sensor network localisation problems); see Liberti *et al.* [61] for a survey.

## 5.2 Combinatorial optimisation

SDP has proven to be a remarkably useful tool for constructing strong bounds for various *combinatorial optimisation* problems. For brevity, we consider just two examples: the *stable set* and *max-cut* problems. For other examples, see, e.g., [1, 33, 34, 60, 64, 86, 96].

Let  $G = (V, E)$  be an undirected graph. A set  $S \subset V$  is called *stable* if no two nodes in  $S$  are adjacent in  $G$ . The *stability number*, denoted by  $\alpha(G)$ , is the maximum cardinality of a stable set in  $G$ . The *stable set problem* calls for a stable set of maximum cardinality. It is not only  $\mathcal{NP}$ -hard in the strong sense, but hard to approximate [41].

A simple 0-1 LP formulation of the stable set problem is:

$$\begin{aligned} \max \quad & \sum_{i \in V} x_i \\ \text{s.t.} \quad & x_i + x_j \leq 1 \quad (\{i, j\} \in E) \\ & x \in \{0, 1\}^n. \end{aligned} \tag{3}$$

Unfortunately, the LP relaxation of this formulation yields an extremely weak upper bound, since one can just set every variable to  $1/2$ . Padberg [77] noted that one can strengthen the LP relaxation by replacing the constraints (3) with *clique inequalities* of the form  $\sum_{i \in C} x_i \leq 1$ , where  $C$  is a maximal clique (set of pairwise-adjacent nodes) in  $G$ . Unfortunately, the number of cliques is in general exponential in  $|V|$ . Even worse, the *separation problem* for the clique inequalities (i.e., the problem of detecting when an LP solution violates a clique inequality) is  $\mathcal{NP}$ -hard (e.g., Grötschel *et al.* [38]).

In his seminal paper, Lovász [63] defined a new upper bound for the stable set problem, which he called  $\theta(G)$ . Grötschel *et al.* [37, 38] showed that  $\theta(G)$  can be computed by solving an SDP. One way to do it is as follows.

We begin by formulating the stable set problem as the following continuous quadratic optimisation problem:

$$\begin{aligned} \max \quad & \sum_{i \in V} x_i \\ \text{s.t.} \quad & x_i^2 - x_i = 0 \quad (i \in V) \end{aligned} \tag{4}$$

$$\begin{aligned} & x_i x_j = 0 \quad (\{i, j\} \in E) \\ & x \in \mathbb{R}^{|V|}. \end{aligned} \tag{5}$$

Now we introduce the matrix

$$Y = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T = \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix},$$

and note that  $Y$  should be psd and have rank 1. We then replace  $xx^T$  with a *matrix variable*  $X$ , and replace the quadratic terms in (4) and (5) with the corresponding entries in  $X$ . This yields the following alternative formulation of the stable set problem:

$$\begin{aligned} \max \quad & \sum_{i \in V} x_i \\ \text{s.t.} \quad & X_{ii} - x_i = 0 \quad (i \in V) \end{aligned} \tag{6}$$

$$X_{ij} = 0 \quad (\{i, j\} \in E) \tag{7}$$

$$\begin{aligned} Y = \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \in \mathcal{S}_+^{n+1} \\ \text{rank}(Y) = 1. \end{aligned} \tag{8}$$

Dropping the rank constraint, which is non-convex, we obtain the desired SDP relaxation. The corresponding upper bound is  $\theta(G)$ .

The following result is due to Grötschel *et al.* [38]. To help the reader, we give a short proof here.

**Proposition 1** *If  $(x, X) \in \mathbb{R}^n \times \mathcal{S}^n$  satisfies (6)–(8), then  $x$  satisfies all clique inequalities.*

**Proof.** Given any clique  $C \subseteq V$ , let  $v(C) \in \{0, 1\}^n$  be a vector with a “1” in position  $i$  if and only if node  $i$  belongs to  $C$ . From constraint (8) and the first definition of psd-ness in Subsect. 2.1, we have

$$\begin{pmatrix} -1 \\ v(C) \end{pmatrix}^T \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \begin{pmatrix} -1 \\ v(C) \end{pmatrix} \geq 0,$$

or, equivalently,

$$2 \sum_{i \in C} x_i - \sum_{i \in C} \sum_{j \in C} X_{ij} \leq 1. \tag{9}$$

Now, due to constraints (6) and (7), we have  $X_{ii} = x_i$  for all  $i \in C$  and  $X_{ij} = 0$  for all  $\{i, j\} \subseteq C$ . Thus, (9) reduces to  $2 \sum_{i \in C} x_i - \sum_{i \in C} x_i \leq 1$ , which is equivalent to the clique inequality on  $C$ .  $\square$

Although easy to prove, this result is remarkable, given the above-mentioned fact that clique separation is  $\mathcal{NP}$ -hard. Indeed, it is a good illustration of the power of SDP relative to LP. Intuitively, the extra power comes from the fact that imposing psd-ness is equivalent to imposing an infinite number of linear inequalities.

Now we move on to max-cut. Given a graph  $G = (V, E)$  and an arbitrary set  $S \subseteq V$ , the set

$$\left\{ \{i, j\} \in E : i \in S, j \in V \setminus S \right\}$$

is called an *edge cutset* or simply *cut*. Suppose we are also given a vector  $w \in \mathbb{R}^{|E|}$  of edge weights. The *max-cut problem* calls for a cut of maximum total weight. The problem is  $\mathcal{NP}$ -hard in the strong sense (Garey *et al.* [30]).

A simple 0-1 LP formulation of max-cut is:

$$\begin{aligned} \max \quad & \sum_{e \in E} w_e x_e \\ \text{s.t.} \quad & x_{ij} + x_{ik} + x_{jk} \leq 2 \quad (\{i, j, k\} \subset V) \end{aligned} \quad (10)$$

$$x_{ik} + x_{jk} \geq x_{ij} \quad (\{i, j\} \subset V, k \in V \setminus \{i, j\}) \quad (11)$$

$$x_{ij} \in \{0, 1\} \quad (\{i, j\} \subset V).$$

Unfortunately, the LP relaxation can be rather weak. Poljak & Tuza [81] showed that, even when  $w \geq 0$ , the upper bound from the relaxation can approach twice the weight of the optimal cut.

An SDP relaxation of the max-cut problem was proposed by Schrijver (unpublished), and then studied in, e.g., [25, 35, 57, 79]. For each  $i \in V$ , let  $z_i$  be a variable taking the value 1 if  $i \in S$ , and  $-1$  otherwise. Then the max-cut problem can be formulated as:

$$\begin{aligned} \max \quad & \frac{1}{2} \sum_{\{i, j\} \in E} (1 - z_i z_j) \\ \text{s.t.} \quad & z_i^2 = 1 \quad (i \in V) \\ & z \in \mathbb{R}^{|V|}. \end{aligned}$$

The corresponding SDP relaxation is:

$$\begin{aligned} \max \quad & \frac{1}{2} \sum_{\{i, j\} \in E} (1 - Z_{ij}) \\ \text{s.t.} \quad & Z_{ii} = 1 \quad (i \in V) \\ & Z \in \mathcal{S}_+^n. \end{aligned}$$

In a major breakthrough, Goemans & Williamson [35] proved that, when  $w \geq 0$ , the upper bound from the SDP is no more than 1.131 times the weight of the optimal cut. (They did this by showing how to compute a cut whose weight is at least 0.878 times the SDP bound.)

Observe that the feasible region to the SDP can be projected into  $x$ -space via the identities  $Z_{ij} = 1 - 2x_{ij}$ . It turns out that the projection “wraps” rather closely around the convex hull of cut vectors. For example, from a consideration of 3-by-3 principal submatrices of  $Z$ , it can be shown that a point in the projection cannot violate any of the triangle inequalities (10), (11) by more than  $1/4$ . This provides a partial explanation for the strength of the SDP bound; see, e.g., [8, 29, 57, 58] for more details.

In a similar way, one can construct SDP relaxations of any problem that can be formulated as a 0-1 LP. In fact, it is possible to construct entire *hierarchies* of SDP relaxations; see, e.g., [56, 59, 65, 85, 89]. Moreover, SDP has been successfully applied to many 0-1 *quadratic* programs; see [44, 60, 80, 91] for early work on the subject, and [51, 95] for some recent applications.

### 5.3 Non-convex quadratic (and polynomial) optimisation

Finally, we consider general quadratic optimisation problems, and their natural generalisation, polynomial optimisation problems.

A *quadratically constrained quadratic program* (QCQP) is a problem of the form:

$$\begin{aligned} \inf \quad & x^T Q^0 x + c^0 \cdot x \\ \text{s.t.} \quad & x^T Q^k x + c^k \cdot x \leq b_k \quad (k = 1, \dots, m) \\ & x \in \mathbb{R}^n, \end{aligned}$$

where  $Q^k \in \mathcal{S}^n$ ,  $c^k \in \mathbb{R}^n$  and  $b_k \in \mathbb{R}$  for  $k = 0, \dots, m$ . If  $Q^0, \dots, Q^m$  are all psd, then the QCQP is convex and can be converted into an SOCP (see Subsect. 4.1). In general, however, QCQP is  $\mathcal{NP}$ -hard in the strong sense. (Indeed, this follows from the fact that the stable set and max-cut problems can be formulated as QCQPs; see the previous subsection). Moreover, non-convex QCQPs arise in many contexts besides OR, including economics and finance (Horst *et al.* [48]) and signal processing (Luo *et al.* [66]).

Ramana [83] proposed the following natural SDP relaxation of QCQP:

$$\begin{aligned} \inf \quad & Q^0 \bullet X + c_0 \cdot x \\ \text{s.t.} \quad & Q^k \bullet X + c_k \cdot x \leq b_k \quad (k = 1, \dots, m) \\ & \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \in \mathcal{S}_{n+1}^+. \end{aligned}$$

The derivation of this relaxation is similar to the one for the stable set problem described in the previous subsection. (Earlier, Shor [90] derived essentially the same relaxation, but in dual form.)

A hierarchy of SDP relaxations for QCQP, similar to the one of Lovász & Schrijver for 0-1 LPs, was proposed by Fujie & Kojima [28]. Some results on

the quality of the SDP bound, similar to the result of Goemans & Williamson [35], are surveyed in Nesterov *et al.* [74].

In practice, the vector  $x$  is usually constrained to be non-negative, or even to lie in a hypercube. This fact can be exploited to derive stronger SDP relaxations; see, e.g., [5, 7, 9, 18, 27]. The best such relaxations are shown by Anstreicher [6] to dominate many other known relaxations. There is also evidence that SDP-based algorithms for QCQP can be effective in practice; e.g., [5, 20, 66].

A natural generalisation of QCQP is *polynomial optimisation*, in which the objective and constraint functions can be arbitrary polynomials. Polynomial optimisation is a fascinating inter-disciplinary field, with relevance not only to OR, but also to statistics, computer science, engineering, and branches of pure mathematics such as algebraic geometry, commutative algebra and moment theory [4, 24]. Interesting and powerful SDP relaxations and hierarchies for polynomial optimisation have been proposed by, e.g., Lasserre [54] and Parrilo [78]. However, the complexity of solving the SDPs in the hierarchies is still not fully settled; see O’Donnell [76].

## 6 Software

Finally, we mention some of the available software systems. We cover modelling interfaces in Subsect. 6.1 and solvers in Subsect. 6.2. For brevity, we do not cover all systems, and refer the reader to Mittelmann [70] for a more comprehensive survey. (Readers may also find the “Decision Tree for Optimization Software”<sup>1</sup> useful.)

### 6.1 Modelling interfaces

We are aware of the following five modelling interface systems:

- CVX<sup>2</sup> is an influential and long-established system for conic programming based on **MatLab**. The CVX website states “CVX is a popular modeling framework for disciplined convex programming that... turns MatLab into a modeling language, allowing constraints and objectives to be specified using standard MatLab syntax.”
- CVXOPT<sup>3</sup> is similar to CVX but based on **Python**. It can be used within the open-source mathematics software system **SageMath**.
- **JuliaOpt**<sup>4</sup> now contains a modelling language called **JuMP**.

---

<sup>1</sup><http://plato.asu.edu/guide.html>

<sup>2</sup><http://cvxr.com/>

<sup>3</sup><http://cvxopt.org/>

<sup>4</sup><http://www.juliaopt.org>

- YALMIP<sup>5</sup> is also based on MatLab, and acts as a convenient front-end to many solvers.
- PICOS<sup>6</sup> provides a front-end to many solvers and is based on Python. It also does some automatic reformulation (e.g., it has a function that “replaces quadratic constraints by equivalent second order cone constraints”).

We remark that, although using modelling languages makes it much easier to model and solve problems, there is usually a computational overhead, in terms of both time and memory.

## 6.2 CO solvers

Now we list some of the commonly-used solvers. (A more comprehensive list can be found on the YALMIP site<sup>7</sup>.)

- CSDP<sup>8</sup> is a C library for solving SDPs using an interior-point method. It is now part of COIN-OR.<sup>9</sup>
- ECOS<sup>10</sup> is an SOCP solver designed for embedded systems. It is written using less than a thousand lines of C code, and it can also handle constraints involving the exponential cone.
- MOSEK<sup>11</sup> has supported both SOCP and SDP since version 7.0. A recent blog entry<sup>12</sup> reports on the solution of an SOCP with over 0.5 million variables (on a powerful parallel processor).
- PENNON<sup>13</sup> is an implementation of a generalized augmented Lagrangian algorithm for SDPs. It can solve SDPs with general convex objective and constraint functions.
- SDPA<sup>14</sup> is a collection of C++ routines for SDP, based on a primal-dual interior-point method. It is designed to exploit sparsity in the constraints.

---

<sup>5</sup><http://yalmip.github.io/allsolvers/>

<sup>6</sup><http://picos.zib.de>

<sup>7</sup><https://yalmip.github.io/allsolvers/>

<sup>8</sup><https://projects.coin-or.org/Csdp/wiki/CSDPUsed>

<sup>9</sup><https://projects.coin-or.org/>

<sup>10</sup><https://www.embotech.com/ECOS>

<sup>11</sup><http://www.mosek.com/>

<sup>12</sup><http://blog.mosek.com/2017/05/biggest-conic-quadratic-problem-solved.html>

<sup>13</sup><http://web.mat.bham.ac.uk/kocvara/pennon/>

<sup>14</sup><http://sdpa.sourceforge.net/>

<sup>14</sup><http://sdpa.sourceforge.net/>

- `SDPLR`<sup>15</sup> is a C library for SDP, based on the augmented Lagrangian method.
- `SDPT3`<sup>16</sup> is a library of MatLab routines that can solve SOCPs, SDPs, and various other problems. An infeasible path following algorithm is used.
- `SeDuMi`<sup>17</sup> is another useful MatLab toolbox, based on the concept of “self-dual embedding”.

Finally, we mention two MatLab packages that are specifically designed for solving polynomial optimisation problems: `GloptiPoly`<sup>18</sup> and `SOSTOOLS`<sup>19</sup>.

## 7 Conclusions

In this guide, we hope to have convinced the reader that Conic Optimisation is an elegant and powerful generalisation of standard linear programming, which allows one to capture many forms of non-linearity that arise in practical problems. In particular, both SOCP and SDP enable one to model many *convex* non-linearities arising in practice (such as convex quadratic and hyperbolic functions, and functions involving norms or eigenvalues); and SDP also provides good bounds and approximation algorithms for many *non-convex* (and  $\mathcal{NP}$ -hard) problems, including a wide range of combinatorial and global optimisation problems.

Moreover, software for SOCP and SDP, and CO in general, is developing rapidly. This includes not only solvers, but also modelling languages and procedures for automatic reformulation. Hence, although CO is more difficult to master than LP, we believe that it will soon become a standard technique for both practitioners and researchers in optimisation, just as LP is at present.

## References

- [1] F. Alizadeh (1995) Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optim.*, 5, 13–51.
- [2] F. Alizadeh & D. Goldfarb (2003) Second-order cone programming. *Math. Program.*, 95, 3–51.

---

<sup>15</sup><http://sburer.github.io/projects.html>

<sup>16</sup><http://www.math.cmu.edu/~reha/sdpt3.html>

<sup>17</sup><http://sedumi.ie.lehigh.edu/>

<sup>18</sup><http://homepages.laas.fr/henrion/software/gloptipoly3/>

<sup>19</sup><http://www.cds.caltech.edu/sostools/>

- [3] F. Alizadeh, J.-P.A. Haeberly & M.L. Overton (1997) Complementarity and nondegeneracy in semidefinite programming. *Math. Program.*, 77, 111–128.
- [4] M. Anjos & J.B. Lasserre (eds.) (2012) *Handbook on Semidefinite, Conic and Polynomial Optimization*. International Series in OR/MS, vol. 166. New York: Springer.
- [5] K.M. Anstreicher (2009) Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. *J. Glob. Optim.*, 43, 471–484.
- [6] K.M. Anstreicher (2012) On convex relaxations for quadratically constrained quadratic programming. *Math. Program.*, 136, 233–251.
- [7] K.M. Anstreicher & S. Burer (2010) Computable representations for convex hulls of low-dimensional quadratic forms. *Math. Program.*, 124: 33–43.
- [8] D. Avis & J. Umemoto (2003) Stronger linear programming relaxations of max-cut. *Math. Program.*, 97, 451–469.
- [9] X. Bao, N.V. Sahinidis & M. Tawarmalani (2011) Semidefinite relaxations for quadratically constrained quadratic programming: a review and comparisons. *Math. Program.*, 129, 129–157.
- [10] S.J. Benson, Y. Ye & X. Zhang (2000) Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM J. Optim.*, 10, 443–461.
- [11] A. Ben-Tal & A. Nemirovski (1999) Robust solutions of uncertain linear programs. *Oper. Res. Lett.*, 25, 1–13.
- [12] A. Ben-Tal & A. Nemirovski (2001) On polyhedral approximations of the second order cone. *Math. Oper. Res.*, 26, 193–205.
- [13] D. Bertsekas (2016) *Nonlinear Programming* (3rd edn.) Belmont, MA: Athena Scientific.
- [14] S. Boyd & L. Vandenberghe (2004) *Convex Optimization*. Cambridge: Cambridge University Press.
- [15] G. Braun, S. Fiorini, S. Pokutta & D. Steurer (2015) Approximation limits of linear programs (beyond hierarchies). *Math. Oper. Res.*, 40, 756–772.
- [16] S. Burer (2009) On the copositive representation of binary and continuous nonconvex quadratic programs. *Math. Program.*, 120, 479–495.

- [17] S. Burer (2012) Copositive programming. In M. Anjos & J.B. Lasserre (eds.) *op. cit.*, 201–218.
- [18] S. Burer & A.N. Letchford (2009) On non-convex quadratic programming with box constraints. *SIAM J. Optim.*, 20, 1073–1089.
- [19] S. Burer & R.D.C. Monteiro (2005) Local minima and convergence in low-rank semidefinite programming. *Math. Program.*, 103, 427–444.
- [20] S.A. Burer & D. Vandembussche (2008) A finite branch-and-bound algorithm for nonconvex quadratic programming via semidefinite relaxations. *Math. Program.*, 113, 259–282.
- [21] V. Chandrasekaran & P. Shah (2017) Relative entropy optimization and its applications. *Math. Program.*, 161, 1–32.
- [22] G.B. Dantzig (1951) Maximization of a linear function of variables subject to linear inequalities. In T.C. Koopmans (ed.) *Activity Analysis of Production and Allocation*, pp. 339–347. New York: Wiley.
- [23] G.B. Dantzig (1963) *Linear Programming and Extensions*. Princeton, NJ: Princeton University Press.
- [24] J.A. De Loera, R. Hemmecke & M. Köppe (2013) *Algebraic and Geometric Ideas in the Theory of Discrete Optimization*. SIAM-MOS Series on Optimization, vol. 14. Philadelphia, PA: SIAM.
- [25] C. Delorme & S. Poljak (1993) Combinatorial properties and the complexity of an eigenvalue approximation of the max-cut problem. *Eur. J. Combin.*, 14, 313–333.
- [26] M.M. Deza & M. Laurent (1997) *Geometry of Cuts and Metrics*. Berlin: Springer-Verlag.
- [27] M. Dür (2010) Copositive programming: a survey. In M. Diehl *et al.* (eds.) *Recent Advances in Optimization and its Applications in Engineering*, pp. 3–20. Berlin: Springer.
- [28] T. Fujie & M. Kojima (1997) Semidefinite programming relaxation for nonconvex quadratic programs. *J. Glob. Optim.*, 10, 367–380.
- [29] L. Galli, K. Kaparis & A.N. Letchford (2012) Complexity results for the gap inequalities for the max-cut problem. *Oper. Res. Lett.*, 40, 149–152.
- [30] M.R. Garey, D.S. Johnson & L. Stockmeyer (1976) Some simplified NP-complete graph problems. *Theor. Comput. Sci.*, 1, 237–267.
- [31] F. Glineur (2000) An extended conic formulation for geometric optimization. *Found. Comput. Decis. Sci.*, 25, 161–174.

- [32] F. Glineur & T. Terlaky (2004) Conic formulation for  $\ell_p$ -norm optimization. *J. Optim. Th. Appl.*, 122, 285–307.
- [33] M.X. Goemans (1997) Semidefinite programming in combinatorial optimization. *Math. Program.*, 79, 143–161.
- [34] M.X. Goemans & F. Rendl (2000) Combinatorial optimization. In H. Wolkowicz *et al.* (eds.) *op. cit.*, pp. 343–360.
- [35] M.X. Goemans and D. Williamson (1995) Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. of the ACM*, 42, 1115–1145.
- [36] J. Gondzio (2012). Interior point methods 25 years later. *Eur. J. Oper. Res.*, 218, 587–601.
- [37] M. Grötschel, L. Lovász & A.J. Schrijver (1981) The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1, 169–197.
- [38] M. Grötschel, L. Lovász & A.J. Schrijver (1988) *Geometric Algorithms and Combinatorial Optimization*. New York: Wiley.
- [39] M. Hall Jr. (1962) Discrete problems. In J. Todd (ed.) *A Survey of Numerical Analysis*, pp. 518–542. New York: McGraw-Hill.
- [40] F.M. Harris (1913) How many parts to make at once. *Factory, The Magazine of Management*, 10: 135–136, 152.
- [41] J. Håstad (1999) Clique is hard to approximate within  $n^{1-\epsilon}$ . *Acta Mathematica*, 182, 105–142.
- [42] C. Helmberg (2000) Fixing variables in semidefinite relaxations. *SIAM J. Matrix Anal. & Appl.*, 21 952–969.
- [43] C. Helmberg (2002) Semidefinite programming. *Eur. J. Oper. Res.*, 137, 461–482.
- [44] C. Helmberg & F. Rendl (1998) Solving quadratic (0,1)-programs by semidefinite programs and cutting planes. *Math. Program.*, 82, 291–315.
- [45] C. Helmberg & F. Rendl (1999) A spectral bundle method for semidefinite programming. *SIAM J. Optim.*, 10, 673–696.
- [46] N. Higham (2002) Computing the nearest correlation matrix—A problem from finance. *IMA J. Numer. Anal.*, 22, 329–343.
- [47] R.A. Horn (2012) *Matrix Analysis* (2nd edn). Cambridge: Cambridge University Press.

- [48] R. Horst, P.M. Pardalos & N. Thoai (2000) *Introduction to Global Optimization* (2nd edn.). Dordrecht: Kluwer.
- [49] C.R. Johnson (1990) Matrix completion problems: a survey. In C.R. Johnson (ed.) *Matrix Theory and Applications*, pp. 171–198. Providence, RI: Amer. Math. Soc.
- [50] L.G. Khachiyan (1979) A polynomial algorithm in linear programming. *Soviet Math. Doklady*, 20, 191–194.
- [51] E. de Klerk, R. Sotirov & U. Truetsch (2015) A new semidefinite programming relaxation for the quadratic assignment problem and its computational perspectives. *INFORMS J. Comput.*, 27, 378–391.
- [52] K. Krishnan & J.E. Mitchell (2006) A unifying framework for several cutting plane methods for semidefinite programming. *Optim. Meth. & Soft.*, 21, 57–74.
- [53] Y.-J. Kuo & H.D. Mittelmann (2004) Interior-point methods for second-order cone programming and OR applications. *Comput. Optim. & Appl.*, 28, 255–285.
- [54] J.B. Lasserre (2001) Global optimization with polynomials and the problem of moments. *SIAM J. Optim.*, 11, 796–817.
- [55] M. Laurent (1998) A connection between positive semidefinite and Euclidean distance matrix completion problems. *Lin. Alg. & Appl.*, 273, 9–22.
- [56] M. Laurent (2003) A comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre relaxations for 0–1 programming. *Math. Oper. Res.*, 28, 470–496.
- [57] M. Laurent & S. Poljak (1995) On a positive semidefinite relaxation of the cut polytope. *Lin. Alg. Appl.*, 223/224, 439–461.
- [58] M. Laurent & S. Poljak (1996) Gap inequalities for the cut polytope. *Eur. J. Combin.*, 17, 233–254.
- [59] M. Laurent & F. Rendl (2005) Semidefinite programming and integer programming. In K. Aardal *et al.* (eds.) *Handbook on Discrete Optimization*, pp. 393–514. Amsterdam: Elsevier.
- [60] C. Lémaréchal & F. Oustry (2001) SDP relaxations in combinatorial optimization from a Lagrangian viewpoint. In N. Hadjisawas & P.M. Pardalos (eds.) *Advances in Convex Analysis and Global Optimization*, pp. 119–134. Dordrecht: Kluwer.

- [61] L. Liberti, C. Lavor, N. Maculan & A. Mucherino (2014) Euclidean distance geometry and applications. *SIAM Review*, 56, 3–69.
- [62] M.S. Lobo, L. Vandenberghe, S. Boyd & H. Lebrecht (1998) Applications of second-order cone programming. *Lin. Alg. Appl.*, 284, 193–228.
- [63] L. Lovász (1979) On the Shannon capacity of a graph. *IEEE Trans. Inform. Th.*, IT-25, 1–7.
- [64] L. Lovász (2003) Semidefinite programs and combinatorial optimization. In B. Reed & C.L. Sales *Recent Advances in Algorithms and Combinatorics*, pp. 137–194. New York: Springer.
- [65] L. Lovász & A.J. Schrijver (1991) Cones of matrices and set-functions and 0–1 optimization. *SIAM J. Optim.*, 1, 166–190.
- [66] Z.-Q. Luo, W.-K. Ma, A.M.-C. So, Y. Ye & S. Zhang (2010) Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27, 20–34.
- [67] J. Malick, J. Povh, F. Rendl & A. Wiegele (2009) Regularization methods for semidefinite programming. *SIAM J. Optim.*, 20, 336–356.
- [68] H.M. Markowitz (1952) Portfolio selection. *J. Finance*, 7, 77–91.
- [69] J.E. Mitchell (2001) Restarting after branching in the SDP approach to MAX-CUT and similar combinatorial optimization problems. *J. Combin. Optim.*, 5, 151–166.
- [70] H.D. Mittelmann (2012) The state-of-the-art in conic optimization software. In M. Anjos & J.B. Lasserre (eds.), *op. cit.*, pp. 671–686.
- [71] T.S. Motzkin (1952) Copositive quadratic forms. *Nat. Bur. Stand. Rep. 1818*, 11–22.
- [72] K.G. Murty & S.N. Kabadi (1987) Some  $\mathcal{NP}$ -complete problems in quadratic and nonlinear programming. *Math. Program.*, 39, 117–129.
- [73] Y. Nesterov & A. Nemirovsky (1994) *Interior Point Methods in Convex Programming: Theory and Applications*. Philadelphia, PA: SIAM Press.
- [74] Y. Nesterov, H. Wolkowicz & Y. Ye (2000) Semidefinite programming relaxations of nonconvex quadratic optimization. In H. Wolkowicz *et al.* (eds.) *op. cit.*, pp. 361–419.
- [75] J. Nie, K. Ranestad & B. Sturmfels (2010) The algebraic degree of semidefinite programming. *Math. Program.*, 122, 379–405.
- [76] R. O’Donnell (2016) SOS is not obviously automatizable, even approximately. ECCO Report No. 141, 2016.

- [77] M.W. Padberg (1973) On the facial structure of set packing polyhedra. *Math. Program.*, 5, 199–215.
- [78] P. Parrilo (2003) Semidefinite programming relaxations for semialgebraic problems. *Math. Program.*, 96, 293–320.
- [79] S. Poljak & F. Rendl (1995) Non-polyhedral relaxations of graph-bisection problems. *SIAM J. Optim.*, 5, 467–487.
- [80] S. Poljak, F. Rendl & H. Wolkowicz (1995) A recipe for semidefinite relaxation for (0,1)-quadratic programming. *J. Glob. Optim.*, 7, 51–73.
- [81] S. Poljak & Zs. Tuza (1994) The expected relative error of the polyhedral approximation of the max-cut problem. *Oper. Res. Lett.*, 16, 191–198.
- [82] J. Povh, F. Rendl & A. Wiegele (2009) A boundary point method to solve semidefinite programs. *Computing*, 78, 277–286.
- [83] M. Ramana (1993) *An Algorithmic Analysis of Multiquadratic and Semidefinite Programming Problems*. PhD thesis, Johns Hopkins University, Baltimore, MD.
- [84] M.V. Ramana (1997) An exact duality theory for semidefinite programming and its complexity implications. *Math. Program.*, 77, 129–162.
- [85] F. Rendl (2010) Semidefinite relaxations for integer programming. In M. Jünger *et al.* (eds.) *50 Years of Integer Programming 1958-2008*, pp. 687-726. Heidelberg: Springer.
- [86] F. Rendl (2012) Semidefinite relaxations for partitioning, assignment and ordering problems. *4OR*, 10, 321–346.
- [87] I.J. Schoenberg (1938) Metric spaces and positive definite functions. *Trans. Amer. Math. Soc.*, 44, 522–536.
- [88] C. Seligman (1993) *Online Astronomy Text*. Available at: <http://cseligman.com/text/history/ellipses.htm>
- [89] H.D. Sherali & W.P. Adams (1990) A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discr. Math.*, 3, 411–430.
- [90] N.Z. Shor (1987) Quadratic optimization problems. *Sov. J. Comput. Syst. Sci.*, 25, 1–11.
- [91] N.Z. Shor (1990) Dual quadratic estimates in polynomial and Boolean programming. *Ann. Oper. Res.*, 25, 163–168.

- [92] A. Skajaa, E.D. Andersen & Y. Ye (2013) Warmstarting the homogeneous and self-dual interior point method for linear and conic quadratic problems. *Math. Program. Comput.*, 5, 1–25.
- [93] M.J. Todd (2001) Semidefinite optimization. *Acta Numerica*, 10, 515–560.
- [94] L. Vanderberghe & S. Boyd (1996) Semidefinite programming. *SIAM Review*, 38, 49–95.
- [95] P. Wang, C. Shen, A. van den Hengel & P.H.S. Torr (2017) Large-scale binary quadratic optimization using semidefinite relaxation and applications. *IEEE Trans. Patt. Anal. & Mach. Intel.*, 39, 470–485.
- [96] H. Wolkowicz & M.F. Anjos (2002) Semidefinite programming for discrete optimization and matrix completion problems. *Discr. Appl. Math.*, 123, 513–577.
- [97] H. Wolkowicz, E. Saigal & L. Vandenberghe (eds.) (2000) *Handbook of Semidefinite Programming*. International Series in OR/MS, vol. 27. New York: Springer.
- [98] Z. Wen, D. Goldfarb & W. Yin (2010) Alternating direction augmented Lagrangian methods for semidefinite programming. *Math. Program. Comput.*, 2, 203–230.
- [99] H. Ziegler (1982) Solving certain singly constrained convex optimization problems in production planning. *Oper. Res. Lett.*, 1, 246–252.