Second-order Cone Programming

Miguel Sousa Lobo[1] Lieven Vandenberghe[2] Stephen Boyd[3]
Hervé Lebret[4]

October 20, 1997

[1]Information Systems Laboratory, Electrical Engineering Department, Stanford University. (mlobo@isl.stanford.edu)
[2]Electrical Engineering Department, University of California, Los Angeles. (vandenbe@isl.stanford.edu)
[3]Information Systems Laboratory, Electrical Engineering Department, Stanford University. (boyd@isl.stanford.edu)
[4]Ecole Nationale Supérieure de Techniques Avancées, Paris. (lebret@ensta.fr)

Abstract

In a *second-order cone program* (SOCP) a linear function is minimized over the intersection of an affine set and the product of second-order (quadratic) cones. SOCPs are nonlinear convex problems that include linear and (convex) quadratic programs as special cases, and arise in many engineering problems, such as filter design, antenna array weight design, truss design, robust estimation, and problems involving friction (*e.g.*, robot grasp).

In this paper we describe the basic theory of SOCPs, a variety of engineering applications, and an efficient primal-dual interior-point method for solving SOCPs. The algorithm we describe shares many of the features of primal-dual interior-point methods for linear programming (LP): Worst-case theoretical analysis shows that the number of iterations required to solve a problem grows at most as the square root of the problem size, while numerical experiments indicate that the typical number of iterations ranges between 5 and 50, almost independent of the problem size.

- Submitted to *Linear Algebra and Applications,* special issue on linear algebra in control, signals and image processing.
- Research supported in part by the Portuguese Government (under Praxis XXI), AFOSR (under F49620-95-1-0318), NSF (under ECS-9222391 and EEC-9420565), and MURI (under F49620-95-1-0525).
- Associated software is available at URL `http://www-isl.stanford.edu/people/boyd` and from anonymous FTP to `isl.stanford.edu` in `pub/boyd/socp`. This includes an implementation in Matlab and C, with a Matlab interface.
- Future versions of the paper will also be made available at the same URL and FTP-site (`pub/boyd/reports/socp.ps.Z`).

1 Introduction

1.1 Second-order cone programming

We consider the *second-order cone problem* (SOCP)

$$\begin{array}{ll} \text{minimize} & f^T x \\ \text{subject to} & \|A_i x + b_i\| \le c_i^T x + d_i \quad i = 1, \dots, L, \end{array} \tag{1}$$

where $x \in \mathbf{R}^n$ is the optimization variable, and the problem parameters are $f \in \mathbf{R}^n$, $A_i \in \mathbf{R}^{(n_i-1)\times n}$, $b_i \in \mathbf{R}^{n_i-1}$, $c_i \in \mathbf{R}^n$, and $d_i \in \mathbf{R}$. The norm appearing in the constraints is the standard Euclidean norm, *i.e.*, $\|u\| = \left(u^T u\right)^{1/2}$. The constraint

$$\|A_i x + b_i\| \le c_i^T x + d_i \tag{2}$$

is called a *second-order cone constraint of dimension* n_i, for the following reason. The standard or unit second-order (convex) cone of dimension k is defined as

$$\mathcal{C}_k = \left\{ \left[\begin{array}{c} u \\ t \end{array} \right] \;\middle|\; u \in \mathbf{R}^{k-1},\ t \in \mathbf{R},\ \|u\| \le t \right\}$$

(which is also called the quadratic, ice-cream, or Lorentz cone). For $k = 1$ we define the unit second-order cone as

$$\mathcal{C}_1 = \{\ t \mid t \in \mathbf{R},\ 0 \le t\ \}.$$

A second-order cone constraint is the inverse image of a second-order cone under an affine mapping:

$$\|A_i x + b_i\| \le c_i^T x + d_i \iff \left[\begin{array}{c} A_i \\ c_i^T \end{array} \right] x + \left[\begin{array}{c} b_i \\ d_i \end{array} \right] \in \mathcal{C}_{n_i},$$

and hence is convex. Thus, the SOCP (1) is a convex programming problem since the objective is a convex function and the constraints are convex.

To simplify notation, we will often use

$$u_i \ =\ A_i x + b_i,\ t_i \ =\ c_i^T x + d_i \quad i = 1, \dots, L$$

so that we can rewrite the problem (1) as

$$\begin{array}{lll} \text{minimize} & f^T x & \\ \text{subject to} & \|u_i\| \le t_i & i = 1, \dots, L \\ & u_i = A_i x + b_i,\ t_i = c_i^T x + d_i & i = 1, \dots, L, \end{array} \tag{3}$$

with $u_i \in \mathbf{R}^{n_i-1}$ and $t \in \mathbf{R}^L$.

Second-order cone constraints can be used to represent several common convex constraints. For example, when all constraints are linear, *i.e.*, when $n_i = 1$ for $i = 1, \dots, L$, the SOCP reduces to the linear program (LP)

$$\begin{array}{ll} \text{minimize} & f^T x \\ \text{subject to} & 0 \le c_i^T x + d_i \quad i = 1, \dots, L. \end{array}$$

Another interesting special case arises when $c_i = 0$, and the ith second-order cone constraint reduces to $\|A_i x + b_i\| \le d_i$, which is equivalent to the (convex) quadratic constraint $\|A_i x + b_i\|^2 \le d_i^2$. When all c_i vanish, the SOCP reduces to a quadratically constrained linear program (QCLP). We will soon see that (convex) quadratic programs (QPs), quadratically-constrained quadratic programs (QCQPs), and many other nonlinear convex optimization problems can be reformulated as SOCPs as well.

We will say $x \in \mathbf{R}^n$ is *feasible* if it satisfies the second-order constraints in (1) and *strictly feasible* if it satisfies the constraints with strict inequality, *i.e.*, $\|A_i x + b_i\| < c_i^T x + d_i$ for $i = 1, \ldots, L$. The SOCP is (strictly) feasible if there exists a (strictly) feasible x. The optimal value of (1) will be denoted as p^*, with the convention that $p^* = +\infty$ if the problem is infeasible.

1.2 The dual SOCP

The *dual* of the SOCP (1) is given by

$$\begin{array}{ll} \text{maximize} & -\sum_{i=1}^{L} \left(b_i^T z_i + d_i w_i \right) \\ \text{subject to} & \sum_{i=1}^{L} \left(A_i^T z_i + c_i w_i \right) = f \\ & \|z_i\| \le w_i, \ \ i = 1, \ldots, L. \end{array} \tag{4}$$

The dual optimization variables are the vectors $z_i \in \mathbf{R}^{n_i - 1}$, and $w \in \mathbf{R}^L$. We denote a set of z_i's, $i = 1, \ldots, L$, by z. The dual SOCP (4) is also a convex programming problem since the objective (which is maximized) is concave, and the constraints are convex. Indeed, by eliminating the equality constraints we can recast the dual SOCP in the same form as the SOCP (1). We will refer to the original SOCP as the *primal SOCP* when we need to distinguish it from the dual.

The vectors z and w are called *dual feasible* if they satisfy the constraints in (4) and *strictly dual feasible* if in addition they satisfy $\|z_i\| < w_i$, $i = 1, \ldots, L$. We say the dual SOCP (4) is (strictly) feasible if there exist (strictly) feasible z_i, w. The optimal value of the dual problem will be denoted d^* (with $d^* = -\infty$ if the dual problem is infeasible).

The basic facts about the dual problem are:

1. *(weak duality)* $p^* \ge d^*$;
2. *(strong duality)* if the primal *or* dual problem is strictly feasible, then $p^* = d^*$;
3. if the primal *and* dual problems are strictly feasible, then there exist primal and dual feasible points that attain the (equal) optimal values.

We only prove the first of these three facts; for a proof of 2 and 3, see, *e.g.*, Nesterov and Nemirovsky [NN94, §4.2.2].

The difference between the primal and dual objectives is called the *duality gap* associated with x, z, w, and will be denoted by $\eta(x, z, w)$, or simply η:

$$\eta(x, z, w) = f^T x + \sum_{i=1}^{L} \left(b_i^T z_i + d_i w_i \right). \tag{5}$$

Weak duality corresponds to the fact that the duality gap is always nonnegative, for any feasible x, z, w. To see this, we observe that the duality gap associated with primal and dual feasible points x, z, w can be expressed as a sum of nonnegative terms, by writing it in the form

$$\eta(x,z,w) = \sum_{i=1}^{L} \left(z_i^T(A_i x + b_i) + w_i(c_i^T x + d_i) \right) = \sum_{i=1}^{L} \left(z_i^T u_i + w_i t_i \right). \tag{6}$$

Each term in the right-hand sum is nonnegative:

$$z_i^T u_i + w_i t_i \geq -\|z_i\|\|u_i\| + w_i t_i \geq 0.$$

The first inequality follows from the Cauchy-Schwarz inequality. The second inequality follows from the fact that $t_i \geq \|u_i\| \geq 0$ and $w_i \geq \|z_i\| \geq 0$. Therefore $\eta(x,z,w) \geq 0$ for any feasible x, z, w, and as an immediate consequence we have $p^* \geq d^*$, *i.e.*, weak duality.

We can also reformulate part 3 of duality result (which we do not prove here) as follows: If the problem is strictly primal and dual feasible, then there exist primal and dual feasible points with zero duality gap. By examining each term (6), we see that the duality gap is zero if and only if the following conditions are satisfied:

$$\|u_i\| < t_i \implies w_i = \|z_i\| = 0, \tag{7}$$

$$\|z_i\| < w_i \implies t_i = \|u_i\| = 0, \tag{8}$$

$$\|z_i\| = w_i, \;\; \|u_i\| = t_i \implies w_i u_i = -t_i z_i. \tag{9}$$

These three conditions generalize the *complementary slackness* conditions between optimal primal and dual solutions in LP. They also yield a sufficient condition for optimality: a primal feasible point x is optimal if, for $u_i = A_i x + b_i$ and $t_i = c_i^T x + d_i$, there exist z, w, such that (7)–(9) hold. (The conditions are also necessary if the primal and dual problems are strictly feasible.)

1.3 Point and outline of the paper

The main goal of the paper is to present an overview of examples and applications of second-order cone programming. We have already mentioned that linear programming is a special case; in §2 we describe several other general convex optimization problems that can be cast as SOCPs. These problems include QP, QCQP, problems involving sums and maxima of norms, and hyperbolic constraints. In §3 we describe a wide variety of engineering applications, including examples in filter design, antenna arrays, robust estimation, and structural optimization.

A second goal of the paper is to describe an efficient primal-dual interior-point algorithm for solving SOCPs. In §4 we describe a primal-dual potential reduction method which is simple, robust, and efficient. This method is certainly not the only possible choice: most of the interior-point methods that have been developed for linear (or semidefinite) programming can be generalized (or specialized) to handle SOCPs as well. The concepts underlying other primal-dual interior-point methods for SOCP, however, are very similar to the ideas behind the method presented here. An implementation of the algorithm (in C, with calls to LAPACK, and including Matlab interface) is available via WWW or FTP [LVB97].

The main reference on interior-point methods for SOCP is the book by Nesterov and Nemirovsky [NN94]. The method we describe is the primal-dual algorithm of [NN94, §4.5] specialized to SOCP. Adler and Alizadeh [AA95] and Nemirovsky and Scheinberg [NS96] also discuss extensions of interior-point LP methods to SOCP. SOCP also fits the framework of optimization over *self-scaled* cones, for which Nesterov and Todd [NT94] have developed and analyzed a special class of primal-dual interior-point methods. Other researchers have worked on interior-point methods for special cases of SOCP. One example is convex quadratic programming; see, for example, Den Hertog [dH93], or Vanderbei [Van97]. As another example, Andersen has developed an interior-point method for minimizing a sum of norms, (which is a special case of SOCP; see §2.2), and describes extensive numerical tests in [And96]. See also Andersen and Andersen [AA97] for software for convex quadratic programs. Xue and Ye present another treatment of the minimization of a sum of norms, with applications to facility location and shortest network problems, in [XY].

One of the best known engineering applications of SOCP is truss design, which was studied by Ben-Tal and Nemirovsky [BTN95], Zowe [BBTZ94], and others. Lebret and Boyd have applied interior-point methods for SOCP to problems of antenna array weight design [Leb94, LB97]. Hansson, Boyd, Vandenberghe and Lobo [HBVL97], discuss control applications involving yield objectives.

We conclude this introduction with some general comments on the place of SOCP in convex optimization relative to other problem classes. SOCP includes several important standard classes of convex optimization problems, such as LP, QP and QCQP. On the other hand, it is itself less general than semidefinite programming (SDP), *i.e.*, the problem of minimizing a linear function over the intersection of an affine set and the cone of positive semidefinite matrices (see, *e.g.*, [VB96]). This can be seen as follows: The second order cone can be embedded in the cone of positive semidefinite matrices since

$$\|u\| \leq t \Longleftrightarrow \begin{bmatrix} tI & u \\ u^T & t \end{bmatrix} \geq 0,$$

i.e., a second-order cone constraint is equivalent to a linear matrix inequality. Using this property the SOCP (1) can be expressed as an SDP

$$\begin{array}{ll} \text{minimize} & f^T x \\ \text{subject to} & \begin{bmatrix} (c_i^T x + d_i)I & A_i x + b_i \\ (A_i x + b_i)^T & c_i^T x + d_i \end{bmatrix} \geq 0, \quad i = 1, \ldots, L. \end{array} \tag{10}$$

Solving SOCPs via SDP is not a good idea in practice, however. Interior-point methods that solve the SOCP directly have a much better worst-case complexity than an SDP method applied to (10): the number of iterations is bounded above by $O(\sqrt{L})$ for the SOCP algorithm, and by $O(\sqrt{\sum_i n_i})$ for the SDP algorithm (see [NN94]). In addition and more importantly in practice, each iteration is much faster: the amount of work per iteration is $O(n^2 \sum_i n_i)$ in the SOCP algorithm and $O(n^2 \sum_i n_i^2)$ for the SDP. The difference between these numbers is significant if the dimensions n_i of the second-order constraints are large. A separate study of (and code for) SOCP is therefore warranted.

2 Problems that can be cast as SOCPs

2.1 Quadratically constrained quadratic programming

We have already seen that an LP is readily expressed as an SOCP with 1-dimensional cones (*i.e.*, $n_i = 1$). Let us now consider the general *convex quadratically constrained quadratic program* (QCQP)

$$\begin{array}{ll} \text{minimize} & x^T P_0 x + 2q_0^T x + r_0 \\ \text{subject to} & x^T P_i x + 2q_i^T x + r_i \le 0 \quad i = 1, \ldots, p, \end{array} \tag{11}$$

where P_0, P_1, $\ldots$, $P_p \in \mathbf{R}^{n \times n}$ are symmetric and positive semidefinite. We will assume for simplicity that the matrices P_i are strictly positive definite, although the problem can be reduced to an SOCP in general. This allows us to write the QCQP (11) as

$$\begin{array}{ll} \text{minimize} & \left\| P_0^{1/2} x + P_0^{-1/2} q_0 \right\|^2 + r_0 - q_0^T P_0^{-1} q_0 \\ \text{subject to} & \left\| P_i^{1/2} x + P_i^{-1/2} q_i \right\|^2 + r_i - q_i^T P_i^{-1} q_i \le 0, \quad i = 1, \ldots, p, \end{array}$$

which can be solved via the SOCP with $p+1$ constraints of dimension $n+1$

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \|P_0^{1/2} x + P_0^{-1/2} q_0\| \le t, \\ & \|P_i^{1/2} x + P_i^{-1/2} q_i\| \le \left(q_i^T P_i^{-1} q_i - r_i \right)^{1/2}, \quad i = 1, \ldots, p, \end{array} \tag{12}$$

where $t \in \mathbf{R}$ is a new optimization variable. The optimal values of (11) and (12) are equal up to a constant and a square root. More precisely, the optimal value of (11) is equal to $p^{*2} + r_0 - q_0^T P_0^{-1} q_0$, where p^* is the optimal value of (12).

As a special case, we can solve a *convex quadratic programming problem* (QP)

$$\begin{array}{ll} \text{minimize} & x^T P_0 x + 2q_0^T x + r_0 \\ \text{subject to} & a_i^T x \le b_i, \quad i = 1, \ldots, p, \end{array}$$

($P_0 > 0$) as an SOCP with one constraint of dimension $n+1$ and p constraints of dimension one:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \|P_0^{1/2} x + P_0^{-1/2} q_0\| \le t \\ & a_i^T x \le b_i, \quad i = 1, \ldots, p, \end{array}$$

where the variables are x and t.

2.2 Sum and maximum of norms

Problems involving sums of norms are readily cast as SOCPs. Let $F_i \in \mathbf{R}^{n_i \times n}$ and $g_i \in \mathbf{R}^{n_i}$, $i = 1, \ldots, p$, be given. The unconstrained problem

$$\text{minimize} \quad \sum_{i=1}^{p} \|F_i x + g_i\|$$

can be expressed as an SOCP by introducing auxiliary variables $t_1, \ldots, t_p$:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{p} t_i \\ \text{subject to} & \|F_j x + g_j\| \leq t_j, \; j = 1, \ldots, p. \end{array}$$

The variables in this problem are $x \in \mathbf{R}^n$ and $t_i \in \mathbf{R}$. We can easily incorporate other second-order cone constraints in the problem, *e.g.*, linear inequalities on x. Specialized methods for minimizing a sum of norms and applications of this problem are discussed in [And96, ACO94, CO94, DO96].

As an interesting special case, consider the complex ℓ_1-norm approximation problem:

$$\text{minimize} \quad \|Ax - b\|_1$$

where $x \in \mathbf{C}^q$, $A \in \mathbf{C}^{p \times q}$, $b \in \mathbf{C}^p$, and the ℓ_1 norm on $\mathbf{C}^p$ is defined by $\|v\|_1 = \sum_{i=1}^{p} |v_i|$. This problem is a sum-of-norms problem, and can be expressed as an SOCP with p constraints of dimension three:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{p} t_i \\ \text{subject to} & \left\| \begin{bmatrix} \Re a_i^T & -\Im a_i^T \\ \Im a_i^T & \Re a_i^T \end{bmatrix} z + \begin{bmatrix} \Re b_i \\ \Im b_i \end{bmatrix} \right\| \leq t_i, \quad i = 1, \ldots, p. \end{array}$$

in the variables $z = [\Re x^T \; \Im x^T]^T \in \mathbf{R}^{2q}$, and t_i.

Similarly, problems involving a maximum of norms can be expressed as SOCPs: the problem

$$\text{minimize} \quad \max_{i=1,\ldots,p} \|F_i x + g_i\|$$

is equivalent to the SOCP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \|F_i x + g_i\| \leq t, \quad i = 1, \ldots, p, \end{array}$$

in the variables $x \in \mathbf{R}^n$ and $t \in \mathbf{R}$.

As a special case, we consider the complex ℓ_∞ norm approximation problem:

$$\text{minimize} \quad \|Ax - b\|_\infty$$

where x, A, and b are as above, and the ℓ_∞ norm on $\mathbf{C}^p$ is defined by $\|v\|_\infty = \max_{i=1}^{p} |v_i|$. This problem can be expressed as the SOCP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \left\| \begin{bmatrix} \Re a_i^T & -\Im a_i^T \\ \Im a_i^T & \Re a_i^T \end{bmatrix} z + \begin{bmatrix} \Re b_i \\ \Im b_i \end{bmatrix} \right\| \leq t, \quad i = 1, \ldots, p, \end{array}$$

with variables $z = [\Re x^T \; \Im x^T]^T \in \mathbf{R}^{2q}$, and $t \in \mathbf{R}$.

As an extension that includes as special cases both the maximum and sum of norms, consider the problem of minimizing the sum of the k largest norms $\|F_i x + g_i\|$, *i.e.*, the problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{k} y_{[i]} \\ \text{subject to} & \|F_i x + g_i\| = y_i, \quad i = 1, \ldots, p, \end{array} \tag{13}$$

where $y_{[i]}$ denotes the ith largest component of y, *i.e.*, $y_{[1]}, \ y_{[2]}, \ldots, y_{[p]}$ are the numbers $y_1, \ y_2, \ldots y_p$ sorted in decreasing order. It can be shown that the objective function in (13) is convex and that the problem is equivalent to the SOCP

$$\begin{array}{ll} \text{minimize} & kt + \sum_{i=1}^{p} y_i \\ \text{suject to} & \|F_i x + g_i\| \le t + y_i, \quad i = 1, \ldots, p \\ & y_i \ge 0, \quad i = 1, \ldots, p, \end{array}$$

where the variables are x, $y \in \mathbf{R}^p$, and t. (See, *e.g.*, [VBW] or [BV97] for further discussion.)

2.3 Problems with hyperbolic constraints

Another large class of convex problems can be cast as SOCPs using the following fact:

$$w^2 \le xy, \ x \ge 0, \ y \ge 0 \Longleftrightarrow \left\| \begin{bmatrix} 2w \\ x - y \end{bmatrix} \right\| \le x + y, \quad x, y \ge 0, \tag{14}$$

and, more generally, when w is a vector,

$$w^T w \le xy, \ x \ge 0, \ y \ge 0 \Longleftrightarrow \left\| \begin{bmatrix} 2w \\ x - y \end{bmatrix} \right\| \le x + y, \quad x, y \ge 0. \tag{15}$$

We refer to these constraints as *hyperbolic constraints*, since they describe half a hyperboloid.

As a first application, consider the problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{p} 1/(a_i^T x + b_i) \\ \text{subject to} & a_i^T x + b_i > 0, \quad i = 1, \ldots, p \\ & c_i^T x + d_i \ge 0, \quad i = 1, \ldots, q, \end{array}$$

which is convex since $1/(a_i^T x + b_i)$ is convex for $a_i^T x + b_i > 0$. This is the problem of maximizing the harmonic mean of some (positive) affine functions of x, over a polytope. This problem can be cast as an SOCP as follows. We first introduce new variables t_i and write the problem as one with hyperbolic constraints:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{p} t_i \\ \text{subject to} & t_i(a_i^T x + b_i) \ge 1, \quad t_i \ge 0, \quad i = 1, \ldots, p \\ & c_i^T x + d_i \ge 0, \quad i = 1, \ldots, q. \end{array}$$

By (14), this can be cast as an SOCP in x and t:

$$\begin{array}{ll}\text{minimize} & \sum_{i=1}^{p} t_i \\ \text{subject to} & \left\| \begin{bmatrix} 2 \\ a_i^T x + b_i - t_i \end{bmatrix} \right\| \leq a_i^T x + b_i + t_i, \quad i = 1, \ldots, p \\ & a_i^T x + b_i \geq 0, \quad t_i \geq 0, \quad i = 1, \ldots, p \\ & c_i^T x + d_i \geq 0, \quad i = 1, \ldots, q. \end{array}$$

As an extension, the quadratic/linear fractional problem

$$\begin{array}{ll}\text{minimize} & \sum_{i=1}^{p} \frac{\|F_i x + g_i\|^2}{a_i^T x + b_i} \\ \text{subject} & a_i^T x + b_i > 0, \quad i = 1, \ldots, p, \end{array}$$

where $F_i \in \mathbf{R}^{q_i \times n}$, $g_i \in \mathbf{R}^{q_i}$, can be cast as an SOCP by first expressing it as

$$\begin{array}{ll}\text{minimize} & \sum_{i=1}^{p} t_i \\ \text{subject to} & (F_i x + g_i)^T (F_i x + g_i) \leq t_i (a_i^T x + b_i), \quad i = 1, \ldots, p \\ & a_i^T x + b_i > 0, \quad i = 1, \ldots, p, \end{array}$$

and then applying (15).

As another example, consider the logarithmic Chebychev approximation problem,

$$\text{minimize} \quad \max_i |\log(a_i^T x) - \log(b_i)|, \tag{16}$$

where $A = [a_1 \cdots a_p]^T \in \mathbf{R}^{p \times n}$, $b \in \mathbf{R}^p$. We assume $b > 0$, and interpret $\log(a_i^T x)$ as $-\infty$ when $a_i^T x \leq 0$. The purpose of (16) is to approximately solve an overdetermined set of equations $Ax \approx b$, measuring the error by the maximum logarithmic deviation between the numbers $a_i^T x$ and b_i. To cast this problem as an SOCP, first note that

$$|\log(a_i^T x) - \log(b_i)| = \log \max(a_i^T x / b_i, b_i / a_i^T x)$$

(assuming $a_i^T x > 0$). The log-Chebychev problem (16) is therefore equivalent to minimizing $\max_i \max(a_i^T x / b_i, b_i / a_i^T x)$, or:

$$\begin{array}{ll}\text{minimize} & t \\ \text{subject to} & 1/t \leq a_i^T x / b_i \leq t, \quad i = 1, \ldots, p. \end{array}$$

This can be expressed as the SOCP

$$\begin{array}{ll}\text{minimize} & t \\ \text{subject to} & a_i^T x / b_i \leq t, \quad i = 1, \ldots, p \\ & \left\| \begin{bmatrix} 2 \\ t - a_i^T x / b_i \end{bmatrix} \right\| \leq t + a_i^T x / b_i, \quad i = 1, \ldots, p. \end{array}$$

As a final illustration of the use of hyperbolic constraints, we consider the problem of maximizing a product of nonnegative affine functions (from Nesterov and Nemirovsky [NN94, §6.2.3, p.227]):

$$\begin{array}{ll} \text{maximize} & \prod_{i=1}^{p}(a_i^T x + b_i) \\ \text{suject to} & a_i^T x + b_i \geq 0, \quad i = 1, \ldots, p. \end{array}$$

For simplicity, we consider the special case $p = 4$; the extension to other values of p is straightforward. We first reformulate the problem by introducing new variables t_1, t_2, and t_3, and by adding hyperbolic constraints:

$$\begin{array}{ll} \text{maximize} & t_3 \\ \text{subject to} & (a_1^T x + b_2)(a_2^T x + b_2) \geq t_1^2, \quad (a_3^T x + b_3)(a_4^T x + b_4) \geq t_2^2 \\ & a_1^T x + b_2 \geq 0, \quad a_2^T x + b_2 \geq 0 \\ & t_1 t_2 \geq t_3^2, \quad t_1, t_2 \geq 0. \end{array}$$

Applying (14) yields an SOCP.

2.4 Matrix-fractional problems

The next class of problems are *matrix-fractional* optimization problems of the form

$$\begin{array}{ll} \text{minimize} & (Fx+g)^T \left(P_0 + x_1 P_1 + \cdots + x_p P_p\right)^{-1} (Fx+g) \\ \text{subject to} & P_0 + x_1 P_1 + \cdots + x_p P_p > 0 \\ & x \geq 0, \end{array} \tag{17}$$

where $P_i = P_i^T \in \mathbf{R}^{n \times n}$, $F \in \mathbf{R}^{n \times p}$ and $g \in \mathbf{R}^n$, and the problem variable is $x \in \mathbf{R}^p$.

We first note that it is possible to solve this problem as an SDP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \left[\begin{array}{cc} P(x) & Fx+g \\ (Fx+g)^T & t \end{array}\right] \geq 0, \end{array}$$

where $P(x) = P_0 + x_1 P_1 + \cdots + x_p P_p$. The equivalence is readily demonstrated by using Schur complements, and holds even when the matrices P_i are indefinite. In the special case where $P_i \geq 0$, we can reformulate the matrix-fractional optimization problem more efficiently as an SOCP, as shown by Nesterov and Nemirovsky [NN94, §6.2.3, p.227]. We assume for simplicity that the matrix P_0 is nonsingular (see [NN94] for the general derivation).

We claim that (17) is equivalent to the following optimization problem in $t_0, \ldots, t_n \in \mathbf{R}$, $y_0, y_1, \ldots, y_p \in \mathbf{R}^n$, and x:

$$\begin{array}{ll} \text{minimize} & t_0 + t_1 + \cdots + t_p \\ \text{subject to} & P_0^{1/2} y_0 + P_1^{1/2} y_1 + \cdots + P_p^{1/2} y_p = Fx + g \\ & \|y_0\|^2 \leq t_0 \\ & \|y_i\|^2 \leq t_i x_i, \quad i = 1, \ldots, p \\ & t_i, \; x_i \geq 0 \;\; i = 1, \ldots, p, \end{array} \tag{18}$$

which can be cast as an SOCP using (15):

$$\begin{array}{ll} \text{minimize} & t_0 + t_1 + \cdots + t_p \\ \text{subject to} & P_0^{1/2} y_0 + \sum_{i=1}^{p} P_i^{1/2} y_i = Fx + g \\ & \left\| \begin{bmatrix} 2y_0 \\ t_0 - 1 \end{bmatrix} \right\| \le t_0 + 1, \\ & \left\| \begin{bmatrix} 2y_i \\ t_i - x_i \end{bmatrix} \right\| \le t_i + x_i, \ i = 1, \dots, p. \end{array}$$

The equivalence between (17) and (18) can be seen as follows. We first eliminate the variables t_i and reduce problem (18) to

$$\begin{array}{ll} \text{minimize} & y_0^T y_0 + y_1^T y_1 / x_1 + \cdots + y_p^T y_p / x_p \\ \text{subject to} & P_0^{1/2} y_0 + P_1^{1/2} y_1 + \cdots + P_p^{1/2} y_p = Fx + g \\ & x \ge 0 \end{array}$$

(interpreting $0/0 = 0$). Since the only constraint on y_i is the equality constraint, we can optimize over y_i by introducing a Lagrange multiplier $\lambda \in \mathbf{R}^n$ for the equality constraint, which gives us y_i in terms of u and x:

$$2y_0 = -P_0^{1/2} \lambda \ \text{ and } \ 2y_i = -x_i P_i^{1/2} \lambda, \ \ i = 1, \dots, p.$$

Next we substitute these expressions for y_i and obtain a minimization problem in λ and x:

$$\begin{array}{ll} \text{minimize} & \frac{1}{4} \lambda^T (P_0 + x_1 P_1 + \cdots + x_p P_p) \lambda \\ \text{subject to} & (P_0 + x_1 P_1 + \cdots + x_p P_p) \lambda = -2(Fx + g) \\ & x \ge 0. \end{array}$$

Finally, eliminating λ yields the matrix-fractional problem (17).

2.5 SOC-representable functions

The above examples illustrate several techniques that can be used to determine whether a convex optimization problem can be cast as an SOCP. In this section we formalize these ideas with the concept of a *second-order cone representation* of a set or function, introduced by Nesterov and Nemirovsky [NN94, §6.2.3].

We say a convex set $C \subseteq \mathbf{R}^n$ is *second-order cone representable* (abbreviated SOC-representable) if it can be represented by a number of second-order cone constraints, possibly after introducing auxiliary variables, *i.e.*, there exist $A_i \in \mathbf{R}^{(n_i - 1) \times (n+m)}$, $b_i \in \mathbf{R}^{n_i - 1}$, $c_i \in \mathbf{R}^{n+m}$, d_i, such that

$$x \in C \iff \exists y \in \mathbf{R}^m \text{ s.t. } \left\| A_i \begin{bmatrix} x \\ y \end{bmatrix} + b_i \right\| \le c_i^T \begin{bmatrix} x \\ y \end{bmatrix} + d_i, \ \ i = 1, \dots, L.$$

We say a function f is second-order cone representable if its epigraph $\{(x,t) \mid f(x) \le t\}$ has a second-order cone representation. The practical consequence is that if f and C are SOC-representable, then the convex optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in C \end{array}$$

can be cast as an SOCP and efficiently solved via interior-point methods.

We have already encountered several examples of SOC-representable functions and sets. SOC-representable functions and sets can also be combined in various ways to yield new SOC-representable functions and sets. For example, if C_1 an C_2 are SOC-representable, then it is straightforward to show that αC_1 $(\alpha \ge 0)$, $C_1 \cap C_2$ and $C_1 + C_2$ are SOC-representable. If f_1 and f_2 are SOC-representable functions, then αf_1 $(\alpha \ge 0)$, $f_1 + f_2$, and $\max\{f_1, f_2\}$ are SOC-representable.

As a less obvious example, if f_1, f_2 are concave with $f_1(x) \ge 0$, $f_2(x) \ge 0$, and $-f_1$ and $-f_2$ are SOC-representable, then $f_1 f_2$ is concave and $-f_1 f_2$ is SOC-representable. In other words the problem of maximizing the product of f_1 and f_2,

$$\begin{array}{ll} \text{maximize} & f_1(x) f_2(x) \\ \text{subject to} & f_1(x) \ge 0, \;\; f_2(x) \ge 0, \end{array}$$

can be cast as an SOCP by first expressing it as

$$\begin{array}{ll} \text{maximize} & t \\ \text{subject to} & t_1 t_2 \ge t \\ & f_1(x) \ge t_1, \;\; f_2(x) \ge t_2 \\ & t_1 \ge 0, \;\; t_2 \ge 0, \end{array}$$

and then using the SOC-representation of $-f_1$ and $-f_2$.

SOC-representable functions are closed under composition. Suppose the convex functions f_1 and f_2 are SOC-representable and f_1 is monotone nondecreasing, so the composition g given by $g(x) = f_1(f_2(x))$ is also convex. Then g is SOC-representable. To see this, note that the epigraph of g can be expressed as

$$\{(x,t) | g(x) \le t\} = \{(x,t) | \exists s \in \mathbf{R} \text{ s.t. } f_1(s) \le t, \;\; f_2(x) \le s\}$$

and the conditions $f_1(s) \le t$, $f_2(x) \le s$ can both be represented via second-order constraints.

3 Applications

3.1 Antenna array weight design

In an antenna array the outputs of several antenna elements are linearly combined to produce a composite array output. The array output has a directional pattern that depends on the relative weights or scale factors used in the combining process, and the goal of weight design is to choose the weights to achieve a desired direction pattern.

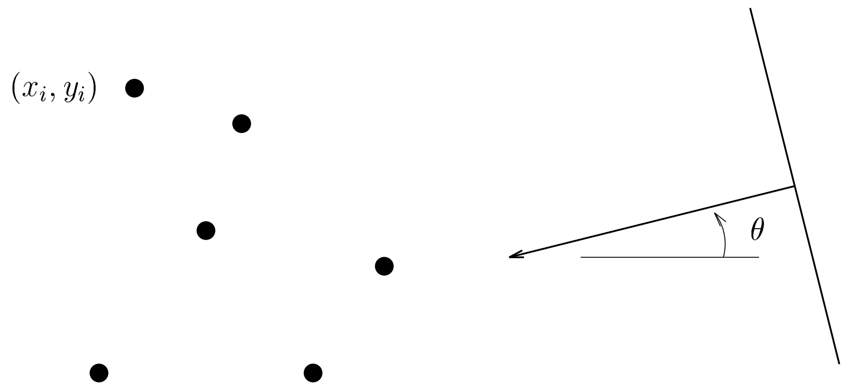

Figure 1: Antenna array.

We will consider the simplest model, an array of omnidirectional antenna elements in a plane, at positions (x_i, y_i), $i = 1, \ldots, n$ (see figure 1). A unit plane wave, of frequency ω, is incident from angle θ. We assume the wave number is one, *i.e.*, the wavelength is $\lambda = 2\pi$. This incident wave induces in the ith antenna element a signal $e^{j(x_i \cos\theta + y_i \sin\theta - \omega t)}$ (where $j = \sqrt{-1}$). This signal is demodulated (*i.e.*, multiplied by $e^{j\omega t}$) to yield the baseband signal, which is the complex number $e^{j(x_i \cos\theta + y_i \sin\theta)}$. This baseband signal is multiplied by the complex factor $w_i \in \mathbf{C}$ to yield

$$\begin{aligned} y_i(\theta) &= w_i \, e^{j(x_i \cos\theta + y_i \sin\theta)} \\ &= (w_{\mathrm{re},i} \cos\gamma_i(\theta) - w_{\mathrm{im},i} \sin\gamma_i(\theta)) + j\,(w_{\mathrm{re},i} \sin\gamma_i(\theta) + w_{\mathrm{im},i} \cos\gamma_i(\theta)), \end{aligned}$$

where $\gamma_i(\theta) = x_i \cos\theta + y_i \sin\theta$. The weights w_i are often called the antenna array coefficients or shading coefficients. The output of the array is the sum of the weighted outputs of the individual array elements:

$$y(\theta) = \sum_{i=1}^{n} y_i(\theta).$$

For a given set of weights, this combined output is a function of the angle of arrival θ of the plane wave; its magnitude is often plotted on a polar plot to show the relative sensitivity of the array to plane waves arriving from different directions. The design problem is to select weights w_i that achieve a desirable directional pattern $y(\theta)$.

The crucial property is that for any θ, $y(\theta)$ is a linear function of the weight vector w. This property is true for a very wide class of array problems, including those in 3 dimensions, with non-omnidirectional elements, and in which the elements are electromagnetically coupled. For these cases the analysis is complicated, but we still have $y(\theta) = a(\theta)w$, for some complex row vector $a(\theta)$.

As an example of a simple design problem, we might insist on the normalization $y(\theta_t) = 1$, where θ_t is called the look or target direction. We also want to make the array relatively insensitive to plane waves arriving from other directions, say, for $|\theta - \theta_t| \geq \Delta$, where 2Δ is called the *beamwidth* of the pattern.

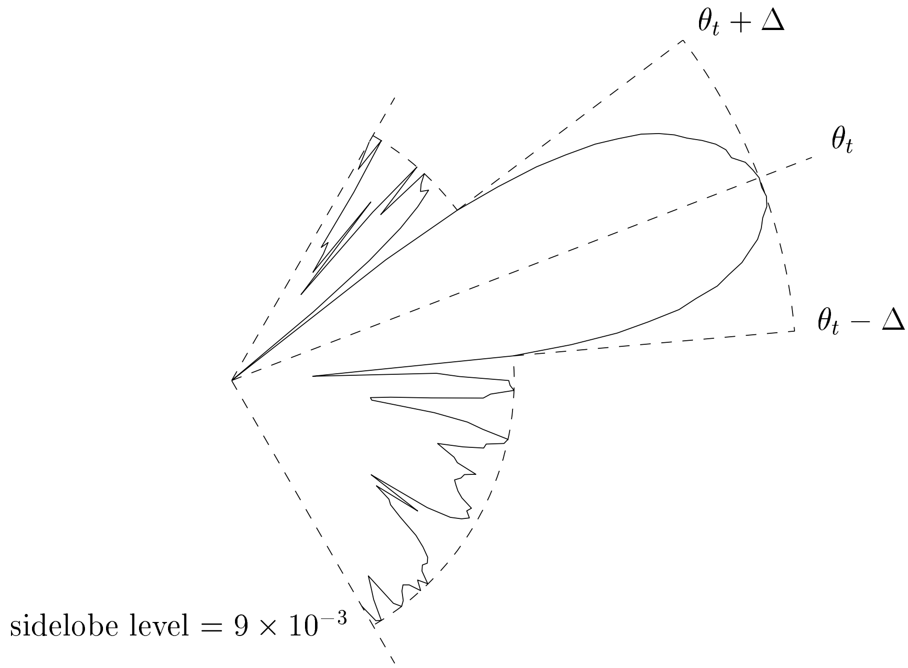

Figure 2: Radial plot of $|y(\theta_i)|^2$ on logarithmic scale, versus angle of incidence. The specifications for sidelobe level are shown in dashed line type; the corresponding optimal design is shown in solid line type. In this example $\theta_t = 40°$, $\Delta = 16°$ (*i.e.*, beamwidth is $32°$), and the sidelobe level is 9×10^{-3}.

To minimize the maximum array sensitivity outside the beam, we solve the problem

$$\begin{array}{ll} \text{minimize} & \max_{|\theta-\theta_t|>\Delta} |y(\theta)| \\ \text{subject to} & y(\theta_t) = 1. \end{array} \tag{19}$$

The square of the optimal value of this problem is called the *sidelobe level* of the array or pattern. This is illustrated in figure2, which also shows a typical optimal design.

This problem can be approximated as an SOCP by discretizing the angle θ, *e.g.*, at $\theta_1, \ldots, \theta_m$, where $m \gg n$. We assume that the target direction is one of the angles, say, $\theta_t = \theta_k$. We can express the array response or pattern as

$$\tilde{y} = Aw,$$

where $\tilde{y} \in \mathbf{C}^m$, $A \in \mathbf{C}^{m \times n}$, where

$$\tilde{y} = \begin{bmatrix} y(\theta_1) \\ \vdots \\ y(\theta_m) \end{bmatrix}, \quad A = \begin{bmatrix} a(\theta_1) \\ \vdots \\ a(\theta_m) \end{bmatrix}$$

The problem (19) can then be approximated as

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & |y(\theta_i)| \le t, \;\; \text{for } |\theta_i - \theta_k| > \Delta \\ & y(\theta_k) = 1 \end{array}$$

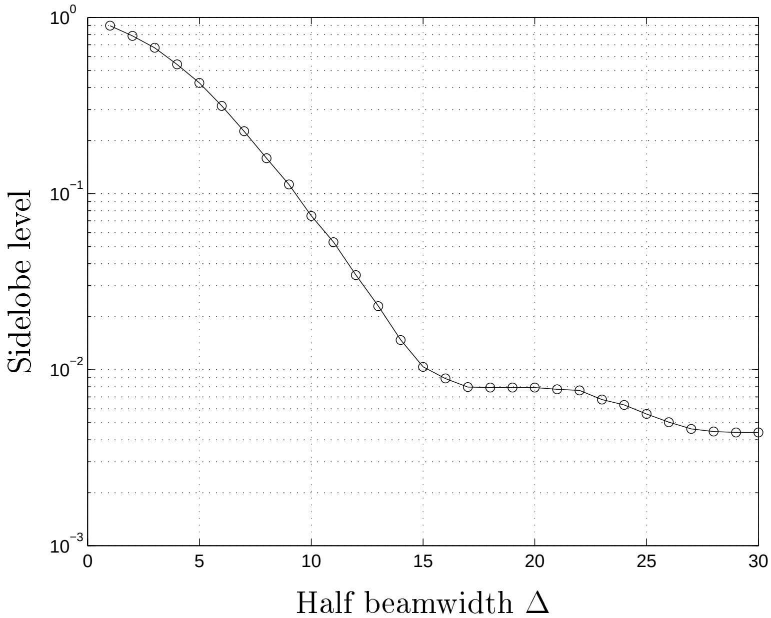

Figure 3: Optimal tradeoff curve of sidelobe level versus half-beamwidth Δ.

which becomes an SOCP when expressed in terms of real and imaginary parts of the variables and data.

This basic problem formulation can be extended in many ways. For example, we can impose a null in a direction θ_l by adding the equality constraint $y(\theta_l) = 0$. We can also add constraints on the coefficients, *e.g.*, that w is real (amplitude only shading), or that $|w_i| \leq 1$ (attenuation only shading), or we can limit the total noise power $\sigma^2 \sum_i |w_i|^2$ in y.

Numerical example

The data for this example, *i.e.*, the matrix A, was obtained from field measurements of an antenna array with eight elements, and angle of incidence θ sampled in 1° increments between -60° and $+60^\circ$. Thus, $A \in \mathbf{C}^{121\times 8}$, the problem variables are $w \in \mathbf{C}^8$, and the response or pattern is given by $\tilde{y} \in \mathbf{C}^{121}$. (For more details on the array hardware and experimental setup, see [SSO95].)

In addition to the sidelobe level and target direction normalization, a constraint on each weight was added, *i.e.*, $|w_i| \leq W_{\max}$, $i = 1, \ldots, 8$, which can be expressed as 8 SOC constraints of dimension 3. (The value of $W_{\max}$ was chosen so that some, but not all, of the weight constraints are active at the optimum.) The target direction was fixed as $\theta_t = 40^\circ$, and the sidelobe level was minimized for various beamwidths. In fact, figure 2 above shows a typical design. As a result, we obtain the (globally) optimal tradeoff curve between beamwidth and optimal sidelobe level for this array. This tradeoff curve is plotted in figure 3.

3.2 FIR filter design

We denote by $h_0, h_1, \ldots, h_{n-1} \in \mathbf{R}$ the coefficients (impulse response) of a finite impulse response (FIR) filter of length n. This means the filter output sequence or signal $y : \mathbf{Z} \to \mathbf{R}$ is related to the input $u : \mathbf{Z} \to \mathbf{R}$ via convolution:

$$y(k) = \sum_{i=0}^{n-1} h_i u(k-i).$$

The frequency response of the filter is the function $H : [0, 2\pi] \to \mathbf{C}$ defined as

$$H(\omega) = \sum_{k=0}^{n-1} h_k e^{-jk\omega},$$

where $j = \sqrt{-1}$ and ω is the (discrete-time) frequency variable.

Minimax complex transfer function design

We first consider the problem of designing a filter that approximates a desired frequency response as well as possible. We assume the desired frequency response is specified by the complex numbers H_i^{des}, $i = 1, \ldots, N$, that are the desired values of the transfer function at the frequencies ω_i, $i = 1, \ldots, N$. The design problem is to choose filter coefficients that minimize the maximum absolute deviation:

$$\text{minimize} \quad \max_{i=1,\ldots,N} \left| H(\omega_i) - H_i^{\mathrm{des}} \right|$$

over all possible coefficients h_k. This is a complex ℓ_∞-approximation problem,

$$\text{minimize} \quad \left\| \begin{bmatrix} 1 & e^{-j\omega_1} & e^{-j2\omega_1} & \cdots & e^{-j(n-1)\omega_1} \\ 1 & e^{-j\omega_2} & e^{-j2\omega_2} & \cdots & e^{-j(n-1)\omega_2} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & e^{-j\omega_N} & e^{-j2\omega_N} & \cdots & e^{-j(n-1)\omega_N} \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{n-1} \end{bmatrix} - \begin{bmatrix} H_1^{\mathrm{des}} \\ H_2^{\mathrm{des}} \\ \vdots \\ H_N^{\mathrm{des}} \end{bmatrix} \right\|_\infty$$

which can be cast as an SOCP using the results of §2.2.

Minimax linear phase lowpass filter design

As a second filter design example, we consider the special case where the filter coefficients are symmetric: $h_k = h_{n-k-1}$. For simplicity we assume n is even. The frequency response simplifies to

$$\begin{aligned} H(\omega) &= \sum_{k=0}^{n/2-1} h_k \left(e^{-jk\omega} + e^{-j(n-k-1)\omega} \right) \\ &= 2e^{-j\omega(n-1)/2} \sum_{k=0}^{n/2-1} h_k \cos((k-(n-1)/2)\omega). \end{aligned}$$

This is called a *linear phase filter* because the transfer function can be factored into a pure delay (which has linear phase),

$$e^{-j\omega(n-1)/2}$$

and real-valued term,

$$T(\omega) = 2\sum_{k=0}^{n/2-1} h_k \cos((k-(n-1)/2)\omega), \tag{20}$$

which is a trigonometric polynomial with coefficients h_i. Note that $|H(\omega)| = |T(\omega)|$.

It was observed already in the 1960s that many interesting design problems for linear phase FIR filters can be cast as LPs. We illustrate this with a simple example involving low-pass filter design, with the following specifications. In the stopband, $\omega_s \le \omega \le \pi$, we impose a minimum attenuation: $|H(\omega)| \le \alpha$. In the passband, $0 \le \omega \le \omega_p$, we want the magnitude of the transfer function to be as close as possible to one, which we achieve by minimizing the maximum deviation $||H(\omega)|-1|$. This leads to the following design problem:

$$\begin{array}{ll} \text{minimize} & \max_{0\le\omega\le\omega_p} ||H(\omega)|-1| \\ \text{subject to} & |H(\omega)| \le \beta, \;\; \omega_s \le \omega \le \pi. \end{array} \tag{21}$$

where the variables are the coefficients h_i, $i = 0, \ldots, n/2-1$, and $\omega_p < \omega_s < \pi$, and $\beta > 0$, are parameters.

In the form given, the design problem (21) is not a convex optimization problem, but it can be simplified and recast as one. First we replace $|H(\omega)|$ by $|T(\omega)|$, the trigonometric polynomial (20). Since we can change the sign of the coefficients h_i (hence, T) without affecting the problem, we can assume without loss of generality that $T(0) > 0$. The optimal value of the problem is always less than one (which is achieved by $h_i = 0$), so in fact we can assume that $T(\omega) > 0$ in the passband. This yields the following optimization problem:

$$\begin{array}{ll} \text{minimize} & \max_{0\le\omega\le\omega_p} |T(\omega)-1| \\ \text{subject to} & |T(\omega)| \le \beta, \;\; \omega_s \le \omega \le \pi. \end{array} \tag{22}$$

This problem is convex, but has semi-infinite constraints. We can form an approximation by discretizing the frequency variable ω: let ω_i, $i = 1, \ldots, N_1 - 1$, be N_1 frequencies in the passband, and ω_i, $i = N_1, \ldots, N-1$, be $N - N_1$ frequencies in the stopband. The discretized version of (22) is the LP

$$\begin{array}{lll} \text{minimize} & t & \\ \text{subject to} & 1 - t \le 2\displaystyle\sum_{k=0}^{n/2-1} h_k \cos((k-(n-1)/2)\omega_i) \le 1 + t, & i = 1, \ldots, N_1 - 1 \\ & -\beta \le \displaystyle\sum_{k=0}^{n/2-1} h_k \cos((k-(n-1)/2)\omega_i) \le \beta, & i = N_1, \ldots, N, \end{array} \tag{23}$$

with as variables $h_0, \ldots, h_{n/2-1}$. (See also the course notes [BV97].)

Bounds on the deviation from specifications between sample points can be derived, showing that the solution of the discretized problem converges to the solution of the continuous problem as the discretization interval becomes small. See, *e.g.*, [Che82] and [WBV96].

Minimax dB linear phase lowpass filter design

We now describe a variation on the design problem just considered, in which the magnitude deviation in the passband is measured on a logarithmic scale, which more accurately captures actual filter design specifications. This problem cannot be formulated as an LP, but can be cast as an SOCP.

We suppose the deviation of the transfer function magnitude from one, in the passband, is measured on a logarithmic scale, *i.e.*, we use the objective

$$\max_{0\le\omega\le\omega_p} |\log|H(\omega)| - \log 1| = \max_{0\le\omega\le\omega_p} |\log|H(\omega)||.$$

This objective is, except for a constant factor, the minimax deviation of the filter magnitude measured in decibels (dB) (which uses $20\log_{10}$ instead of log).

We can handle the resulting problem in a way similar to the minimax lowpass filter problem described above. The logarithmic deviation of T is handled using SOCP in a way similar to the log-Chebychev approximation problem of §2.3: we introduce a new variable t, and modify problem (23) as

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & 1/t \le 2\displaystyle\sum_{k=0}^{n/2-1} h_k \cos((k-(n-1)/2)\omega_i) \le t, \quad i=1,\ldots,N_1-1 \\ & -\beta \le 2\displaystyle\sum_{k=0}^{n/2-1} h_k \cos((k-(n-1)/2)\omega_i) \le \beta, \quad i=N_1,\ldots,N. \end{array} \tag{24}$$

Note that here, the objective t represents the *fractional deviation* of $|H(\omega)|$ from one, whereas in (23) t represents the *absolute deviation*. The optimal value (in dB) of the minimax dB design problem is given by $20\log_{10} t^*$, where t^* is the optimal value of (24).

After reformulating the hyperbolic constraints as second-order constraints, we obtain the SOCP:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \left\| \begin{bmatrix} 2 \\ u-t \end{bmatrix} \right\| \le u+t, \ u,t\ge 0 \\ & u \le 2\displaystyle\sum_{k=0}^{n/2-1} h_k \cos((k-(n-1)/2)\omega_i) \le t, \quad i=1,\ldots,N_1-1 \\ & -\beta \le 2\displaystyle\sum_{k=0}^{n/2-1} h_k \cos((k-(n-1)/2)\omega_i) \le \beta, \quad i=N_1,\ldots,N. \end{array} \tag{25}$$

For more on this subject, see [BB91, p.380], [OS70, §5.6]. The topic of FIR filter design using convex optimization and interior-point algorithms is pursued in much greater detail in [WBV96].

3.3 Portfolio optimization with loss risk constraints

We consider a classical portfolio problem with n assets or stocks held over one period. x_i will denote the amount of asset i held at the beginning of (and throughout) the period, and

p_i will denote the price change of asset i over the period, so the return is $r = p^T x$. The optimization variable is the portfolio vector $x \in \mathbf{R}^n$. The simplest assumptions are $x_i \geq 0$ (*i.e.*, no short positions) and $x_1 + \cdots + x_n = 1$ (*i.e.*, unit total budget).

We take a simple stochastic model for price changes: $p \in \mathbf{R}^n$ is Gaussian, with known mean $\overline{p}$ and covariance Σ. Therefore with portfolio $x \in \mathbf{R}^n$, the return r is a (scalar) Gaussian random variable with mean $\overline{r} = \overline{p}^T x$ and variance $\sigma_r = x^T \Sigma x$. The choice of portfolio x involves the (classical, Markowitz) tradeoff between return mean and variance.

Using SOCP, we can directly handle constraints that limit the risk of various levels of loss. Consider a loss risk constraint of the form

$$\mathrm{Prob}(r \leq \alpha) \leq \beta, \tag{26}$$

where α is a given unwanted return level (*e.g.*, an excessive loss) and β is a given maximum probability. This constraint can be written as

$$\mathrm{Prob}\left(\frac{r - \overline{r}}{\sqrt{\sigma_r}} \leq \frac{\alpha - \overline{r}}{\sqrt{\sigma_r}}\right) \leq \beta,$$

which in turn can be expressed as

$$\frac{\alpha - \overline{r}}{\sqrt{\sigma_r}} \leq \Phi^{-1}(\beta),$$

where

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-t^2/2}\, dt$$

is the CDF of a unit Gaussian random variable. Thus the loss risk constraint (26) can be expressed in terms of the portfolio vector x as

$$\overline{p}^T x + \Phi^{-1}(\beta) \left\| \Sigma^{1/2} x \right\| \geq \alpha. \tag{27}$$

Now, provided $\beta \leq 1/2$ (*i.e.*, $\Phi^{-1}(\beta) \leq 0$), this loss risk constraint is a second-order cone constraint. (If $\beta > 1/2$, the loss risk constraint becomes concave in x.)

The problem of maximizing the expected return subject to a bound on the loss risk (with $\beta \leq 1/2$), can therefore be cast as a simple SOCP with one second-order cone constraint:

$$\begin{array}{ll} \text{maximize} & \overline{p}^T x \\ \text{subject to} & \overline{p}^T x + \Phi^{-1}(\beta) \left\| \Sigma^{1/2} x \right\| \geq \alpha \\ & x \geq 0, \ \displaystyle\sum_{i=1}^{n} x_i = 1. \end{array}$$

There are many extensions on this simple problem. For example, we can impose several loss risk constraints, *i.e.*,

$$\mathrm{Prob}(r \leq \alpha_i) \leq \beta_i, \quad i = 1, \ldots, k,$$

(where $\beta_i \leq 1/2$), which expresses the risks (β_i) we are willing to accept for various levels of loss (α_i).

As another variation, we can handle uncertainty in the statistical model $(\bar{p}, \Sigma)$ for the price changes during the period. Suppose we have L different possible scenarios, each of which is modeled by a simple Gaussian model for the price change vector, with mean $\bar{p}_k$ and covariance Σ_k. We can then take a worst-case approach and maximize the minimum of the expected returns for the L different scenarios, subject to a constraint on the loss risk for each scenario. In other words, we solve the SOCP

$$\begin{array}{ll} \text{maximize} & \min_k \overline{p}_k^T x \\ \text{subject to} & \overline{p}_k^T x + \Phi^{-1}(\beta) \left\| \Sigma_k^{1/2} x \right\| \geq \alpha, \quad k = 1, \ldots, L \\ & x \geq 0, \quad \sum_{i=1}^n x_i = 1. \end{array}$$

Note that the constraints impose the loss risk limit under all L scenarios.

As another (standard) extension, we can allow short positions, *i.e.*, $x_i < 0$. To do this we introduce variables x_{long} and x_{short}, with

$$x_{\text{long}} \geq 0, \quad x_{\text{short}} \geq 0, \quad x = x_{\text{long}} - x_{\text{short}}, \quad \sum_{i=1}^n x_{\text{short}} \leq \eta \sum_{i=1}^n x_{\text{long}}.$$

(The last constraint limits the total short position to some fraction η of the total long position.)

3.4 Robust linear programming

We consider a linear program,

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i, \quad i = 1, \ldots, m, \end{array}$$

in which there is some uncertainty or variation in the parameters c, a_i, b_i. To simplify the exposition we will assume that c and b_i are fixed, and that a_i are known to lie in given ellipsoids:

$$a_i \in \mathcal{E}_i = \{ \overline{a}_i + P_i u \mid \|u\| \leq 1 \},$$

where $P_i = P_i^T \geq 0$. (If P_i is singular we obtain 'flat' ellipsoids, of dimension $\operatorname{rank}(P_i)$).

In a worst-case framework, we require that the constraints be satisfied for all possible values of the parameters a_i, which leads us to the *robust linear program*

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i, \text{ for all } a_i \in \mathcal{E}_i, \quad i = 1, \ldots, m. \end{array} \tag{28}$$

The robust linear constraint $a_i^T x \leq b_i$ for all $a_i \in \mathcal{E}_i$ can be expressed as

$$\max\{ a_i^T x \mid a_i \in \mathcal{E}_i \} = \overline{a}_i^T x + \|P_i x\| \leq b_i,$$

which is evidently a second-order cone constraint. Hence the robust LP (28) can be expressed as the SOCP

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \overline{a}_i^T x + \|P_i x\| \leq b_i, \quad i = 1, \ldots, m. \end{array}$$

Note that the additional norm terms act as 'regularization terms', discouraging large x in directions with considerable uncertainty in the parameters a_i.

The same problem can be considered in a statistical framework as well. Here we suppose that the parameters a_i are independent, with Gaussian distribution with mean $\overline{a}_i$ and variance Σ_i. We require that each constraint $a_i^T x \le b_i$ should hold with a probability (confidence) exceeding η, where $\eta \ge 0.5$. Exactly as in the portfolio optimization problem (§3.3), these constraints are equivalent to the second-order constraints

$$\overline{a}_i^T x - \Phi^{-1}(1-\eta)\|\Sigma_i^{1/2} x\| \le b_i,$$

so the robust LP again becomes an SOCP.

We refer to Ben-Tal and Nemirovsky [BTN96], and Oustry, El Ghaoui, and Lebret [OEL96] for a further discussion of robustness in convex optimization.

3.5 Robust least-squares

The idea of incorporating robustness to parameter variation into a problem can be extended to many problems, *e.g.*, least-squares.

Suppose we are given an overdetermined set of equations $Ax \approx b$, where $A \in \mathbf{R}^{m\times n}, b \in \mathbf{R}^m$ are subject to unknown but bounded errors δA and δb with $\|\delta A\| \le \rho,\ \|\delta b\| \le \xi$ (where the matrix norm is the spectral norm, or maximum singular value). We define the *robust least-squares solution* as the solution $\widehat{x} \in \mathbf{R}^n$ that minimizes the largest possible residual, *i.e.*, $\widehat{x}$ is the solution of

$$\text{minimize} \quad \max_{\|\delta A\|\le\rho,\ \|\delta b\|\le\xi} \|(A+\delta A)x - (b+\delta b)\|. \tag{29}$$

This is the *robust least-squares problem* introduced by El Ghaoui and Lebret [EL] and by Chandrasekaran, Golub, Gu and Sayed [CGGS96]. The objective function in problem (29) can be written in a closed form, by noting that

$$\begin{aligned}
\max_{\|\delta A\|\le\rho,\ \|\delta b\|\le\xi} \|(A+\delta A)x - (b+\delta b)\| &= \max_{\|\delta A\|\le\rho,\ \|\delta b\|\le\xi}\ \max_{\|y\|\le 1} y^T(Ax-b) + y^T\delta A x - y^T\delta b \\
&= \max_{\|z\|\le\rho}\ \max_{\|y\|\le 1} y^T(Ax-b) + z^T x + \xi \\
&= \|Ax-b\| + \rho\|x\| + \xi.
\end{aligned}$$

Problem (29) is therefore equivalent to minimizing a sum of Euclidean norms:

$$\text{minimize } \|Ax-b\| + \rho\|x\| + \xi.$$

Although this problem can be solved as an SOCP, there is a simpler solution via the SVD of A. The SOCP-formulation becomes useful as soon as we add additional constraints on x, *e.g.*, nonnegativity constraints.

A variation on this problem is to assume that the rows a_i of A are subject to independent errors, but known to lie in a given ellipsoid: $a_i \in \mathcal{E}_i$, where

$$\mathcal{E}_i = \{\overline{a}_i + P_i u \mid \|u\| \le 1\} \qquad (P_i = P_i^T > 0).$$

We obtain the robust least squares estimate x by minimizing the worst-case residual:

$$\text{minimize} \quad \max_{a_i \in \mathcal{E}_i} \left(\sum_{i=1}^n \left(a_i^T x - b_i \right)^2 \right)^{1/2}. \tag{30}$$

We first work out the objective function in a closed form:

$$\begin{aligned}
\max_{\|u\| \le 1} \left| \overline{a}_i^T x - b_i + u^T P_i x \right| &= \max_{\|u\| \le 1} \max \left\{ \overline{a}_i^T x - b_i + u^T P_i x, -\overline{a}_i^T x + b_i - u^T P_i x \right\} \\
&= \max \left\{ \overline{a}_i^T x - b_i + \|P_i x\|, -\overline{a}_i^T x + b_i + \|P_i x\| \right\} \\
&= \left| \overline{a}_i^T x - b_i \right| + \|P_i x\|.
\end{aligned}$$

Hence, the robust least-squares problem (30) can be formulated as

$$\text{minimize} \quad \left(\sum_{i=1}^n \left(\left| \overline{a}_i^T x - b_i \right| + \|P_i x\| \right)^2 \right)^{1/2}$$

which can be cast as the SOCP

$$\begin{array}{ll}
\text{minimize} & s \\
\text{subject to} & \|t\| \le s \\
& \left| \overline{a}_i^T x - b_i \right| + \|P_i x\| \le t_i, \quad i = 1, \ldots, n.
\end{array}$$

These two robust variations on the least squares problem can be extended to allow for uncertainty on b. For the first problem, suppose the errors δA and δb are bounded as $\|[\delta A \ \delta b]\| \le \rho$, Using the same analysis as above it can be shown that

$$\max_{\|[\delta A \ \ \delta b]\| \le \rho} \|(A + \delta A)x - (b + \delta b) = \|Ax - b\| + \rho \left\| \begin{bmatrix} x \\ 1 \end{bmatrix} \right\|.$$

The robust least-squares solution can therefore be found by solving

$$\text{minimize } \|Ax - b\| + \rho \left\| \begin{bmatrix} x \\ 1 \end{bmatrix} \right\|.$$

In the second problem, we can assume b_i is bounded by $b_i \in [\overline{b}_i - p_i, \overline{b}_i + p_i]$. A straightforward calculation yields

$$\text{minimize} \quad \left(\sum_{i=1}^n \left(\left| \overline{a}_i^T x - \overline{b}_i \right| + \|P_i x\| + p_i \right)^2 \right)^{1/2}$$

which can be easily cast as an SOCP.

3.6 Truss design

Ben-Tal and Bendsøe in [BTB93] and Nemirovsky in [BTN92] consider the following problem from structural optimization. A structure of k linear elastic bars connects a set of p nodes.

The task is to size the bars, *i.e.*, determine x_i, the cross-sectional areas of the bars, that yield the stiffest truss subject to constraints such as a total weight limit.

In the simplest version of the problem we consider one fixed set of externally applied nodal forces f_i, $i = 1, \ldots, p$; more complicated versions consider multiple loading scenarios. The vector of small node displacements resulting from the load forces f will be denoted d. One objective that measures stiffness of the truss is the elastic stored energy $\frac{1}{2}f^T d$, which is small if the structure is stiff. The applied forces f and displacements d are linearly related: $f = K(x)d$, where

$$K(x) \triangleq \sum_{i=1}^{k} x_i K_i$$

is called the stiffness matrix of the structure. The matrices K_i are all symmetric positive semidefinite and depend only on fixed parameters (Young's modulus, length of the bars, and geometry). To maximize the stiffness of the structure, we minimize the elastic energy, *i.e.*, $f^T K(x)^{-1} f/2$. Note that increasing any x_i will decrease this objective, *i.e.*, stiffen the structure.

We impose a constraint on the total volume (or equivalently, weight), of the structure, *i.e.*, $\sum_i l_i x_i \leq v_{\max}$, where l_i is the length of the ith bar, and $v_{\max}$ is maximum allowed volume of the (bars of the) structure. Other typical constraints include upper and lower bounds on each bar cross-sectional area, *i.e.*, $\underline{x}_i \leq x_i \leq \overline{x}_i$. For simplicity, we assume that $\underline{x}_i > 0$, and that $K(x) > 0$ for all positive values of x_i.

The optimization problem then becomes

$$\begin{array}{ll} \text{minimize} & f^T K(x)^{-1} f \\ \text{subject to} & \sum_{i=1}^{k} l_i x_i \leq v \\ & \underline{x}_i \leq x_i \leq \overline{x}_i, \quad i = 1, \ldots, k. \end{array}$$

where d and x are the variables. This problem can be cast as an SOCP since the objective has the matrix-fractional form described in §2.4.

Several extensions can be developed, *e.g.*, multiple loading scenarios. See also [ABBTZ92, BBTZ94]. For a survey and further references, see Ben-Tal and Nemirovski [BTN95].

3.7 Equilibrium of system with piecewise-linear springs

We consider a mechanical system that consists of N nodes at positions $x_1, \ldots, x_n \in \mathbf{R}^2$, with node i connected to node $i+1$, for $i = 1, \ldots, N-1$, by a nonlinear spring. The nodes x_1 and x_N are fixed at given values a and b, respectively. The tension T_i in spring i is a nonlinear function of the distance between its endpoints, *i.e.*, $\|x_i - x_{i+1}\|$:

$$T_i = k(\|x_i - x_{i+1}\| - l_0)_+ \tag{31}$$

where $z_+ = \max\{z, 0\}$. Here $k > 0$ denotes the stiffness of the springs and $l_0 > 0$ is its natural (no tension) length. In this model the springs can only produce positive tension (which would be the case if they buckled under compression). Each node has a mass of weight $w_i \geq 0$ attached to it. This is shown in figure (4).

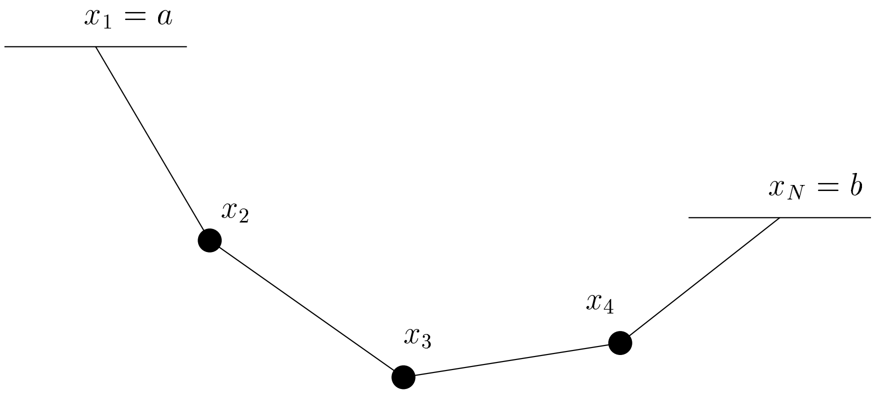

Figure 4: System of nodes (weights) connected by springs. The first and last node positions, *i.e.*, x_1 and x_N, are fixed.

The problem is to compute the equilibrium configuration of the system, *i.e.*, values of $x_1, \ldots, x_N$ such that the net force on each node is zero. This can be done by finding the minimum energy configuration, *i.e.*, solving the optimization problem

$$\begin{array}{ll} \text{minimize} & \sum_i w_i e_2^T x^i + \sum_i \phi(\|x_i - x_j\|) \\ \text{subject to} & x_1 = a, \;\; x_N = b \end{array}$$

where e_2 is the second unit vector (which points up), and $\phi(d)$ is the potential energy stored in a spring stretched to an elongation d:

$$\phi(d) = \int_0^d k(a - l_0)_+ da = (k/2)(d - l_0)_+^2.$$

This objective can be shown to be convex, hence the problem is convex. If we write it as

$$\begin{array}{ll} \text{minimize} & \sum_i w_i e_2^T x^i + (k/2) t^T t \\ \text{subject to} & \|x_i - x_{i+1}\| - l_0 \leq t_i, \;\; i = 1, \ldots, N-1 \\ & 0 \leq t_i, \;\; i = 1, \ldots, N-1 \\ & x_1 = a, \;\; x_N = b, \end{array}$$

we can substitute y for $t^T t$ and add the hyperbolic constraint

$$t^T t \leq y \Longleftrightarrow \left\| \begin{bmatrix} 2t \\ 1 - y \end{bmatrix} \right\| \leq 1 + y,$$

thereby obtaining an SOCP.

Several extensions to this problem are possible, such as considering masses in $\mathbf{R}^3$, springs connecting arbitrary nodes, or limits on extension of springs. In general, if the spring tension versus extension function is piecewise linear and increasing, the equilibrium configuration can be found via SOCP.

4 Primal-dual interior-point method

We briefly describe an efficient method for solving second-order cone problems. The method is the primal-dual potential reduction method of Nesterov and Nemirovsky [NN94, §4.5] applied to SOCP. When specialized to LP, the algorithm reduces to a variation of Ye's potential reduction method [Ye91].

The underlying ideas and concepts are similar for most other primal-dual interior-point methods, so the description in this section can serve as an introduction to other primal-dual methods for SOCP as well. For example, the algorithm we describe can use (without any other change) the symmetric primal-dual search directions developed by Nesterov and Todd for self-scaled cones [NN94].

We first introduce some new notation that will simplify the formulas considerably. We define $\bar{A}$, $\bar{b}$, X, and Z as

$$\bar{A} = \begin{bmatrix} A_1 \\ c_1^T \\ \vdots \\ A_L \\ c_L^T \end{bmatrix}, \quad \bar{b} = \begin{bmatrix} b_1 \\ d_1 \\ \vdots \\ b_L \\ d_L \end{bmatrix}, \quad X = \begin{bmatrix} u_1 \\ t_1 \\ \vdots \\ u_L \\ t_L \end{bmatrix}, \quad Z = \begin{bmatrix} z_1 \\ w_1 \\ \vdots \\ z_L \\ w_L \end{bmatrix}.$$

This allows us to write the primal and dual SOCPs (1) and (4) more compactly as

$$\begin{array}{ll} \text{minimize} & f^T x \\ \text{subject to} & X = \bar{A}x + \bar{b} \in \mathcal{C}^{n_1} \times \mathcal{C}^{n_2} \times \cdots \times \mathcal{C}^{n_L}. \end{array}$$

and

$$\begin{array}{ll} \text{maximize} & -\bar{b}^T Z \\ \text{subject to} & \bar{A}^T Z = f \\ & Z \in \mathcal{C}^{n_1} \times \mathcal{C}^{n_2} \times \cdots \times \mathcal{C}^{n_L}. \end{array}$$

Note also that in this notation the duality gap associated with a pair of primal and dual feasible points x, Z is simply

$$\eta = X^T Z.$$

4.1 Barrier for second-order cone

We define, for $u \in \mathbf{R}^{m-1}$, $t \in \mathbf{R}$,

$$\phi(u,t) = \begin{cases} -\log\left(t^2 - \|u\|^2\right) & \|u\| < t \\ \infty & \text{otherwise.} \end{cases} \tag{32}$$

The function ϕ is a *barrier function* for the second-order cone $\mathcal{C}_m$: $\phi(u,t)$ is finite if and only if $(u,t) \in \mathcal{C}_m$ (*i.e.*, $\|u\| < t$), and $\phi(u,t)$ converges to ∞ as (u,t) approaches the boundary of $\mathcal{C}_m$. It is also smooth and convex on the interior of the second-order order cone. Its first and second derivatives are given by

$$\nabla\phi(u,t) = \frac{2}{t^2 - u^T u} \begin{bmatrix} u \\ -t \end{bmatrix}$$

and

$$\nabla^2\phi(u,t) = \frac{2}{(t^2-u^Tu)^2}\left[\begin{array}{cc} (t^2-u^Tu)I+2uu^T & -2tu \\ -2tu^T & t^2+u^Tu \end{array}\right].$$

4.2 Primal-dual potential function

For strictly feasible (x, Z), we define the *primal-dual potential function* as

$$\varphi(x,Z) = (2L+\nu\sqrt{2L})\log\eta + \sum_{i=1}^{L}\Big(\phi(A_ix+b_i, c_i^Tx+d_i)+\phi(z_i,w_i)\Big) - 2L\log L \qquad (33)$$

where $\nu \geq 1$ is an algorithm parameter, and η is the duality gap (5) associated with (x, Z). The most important property of the potential function is the inequality

$$\eta(x,Z) \leq \exp\Big(\varphi(x,Z)\Big/\nu\sqrt{2L}\Big), \qquad (34)$$

which holds for all strictly feasible x, Z. Therefore, if the potential function is small, the duality gap must be small. In particular, if $\varphi \to -\infty$, then $\eta \to 0$ and (x, Z) approaches optimality.

The inequality (34) can be easily verified by noting the fact that

$$\psi(x,Z) \triangleq 2L\log\eta + \sum_{i=1}^{L}\Big(\phi(A_ix+b_i, c_i^Tx+d_i)+\phi(z_i,w_i)\Big) - 2L\log L \geq 0 \qquad (35)$$

for all strictly feasible x, Z (see the appendix). This implies $\varphi(x,Z) \geq \nu\sqrt{2L}\log(\eta(x,Z))$, and hence (34).

4.3 Primal-dual potential reduction algorithm

In a primal-dual potential reduction method, we start with strictly primal and dual x, Z, and update them in such a way that the potential function $\varphi(x, Z)$ is reduced at each iteration by at least some guaranteed amount. There exist several variations of this idea. In this section we present one such variation, the primal-dual potential reduction algorithm of Nesterov and Nemirovsky [NN94, §4.5].

At each iteration of the Nesterov and Nemirovsky method, primal and dual search directions δx, δZ are computed by solving the set of linear equations

$$\left[\begin{array}{cc} H^{-1} & \bar{A} \\ \bar{A}^T & 0 \end{array}\right]\left[\begin{array}{c} \delta Z \\ \delta x \end{array}\right] = \left[\begin{array}{c} -H^{-1}(\rho Z+g) \\ 0 \end{array}\right] \qquad (36)$$

in the variables δx, δZ, where

$$H = \left[\begin{array}{ccc} \nabla^2\phi(u_1,t_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \nabla^2\phi(u_L,t_L) \end{array}\right], \quad g = \left[\begin{array}{c} \nabla\phi(u_1,t_1) \\ \vdots \\ \nabla\phi(u_L,t_L) \end{array}\right],$$

and ρ is equal to $\rho = (2L + \nu\sqrt{2L})/\eta$. (And as before, $u_i = A_i x + b_i$ and $t_i = c_i^T x + d_i$.) Note that

$$\nabla^2 \phi(u,t)^{-1} = \frac{1}{2} \begin{bmatrix} (t^2 - u^T u)I + 2uu^T & 2tu \\ 2tu^T & t^2 + u^T u \end{bmatrix},$$

and therefore forming $H^{-1} = \mathbf{diag}\,(\nabla^2\phi(u_1,t_1)^{-1}, \ldots, \nabla^2\phi(u_L,t_L)^{-1})$ does not require a matrix inversion.

The outline of the algorithm is as follows.

Primal-dual potential reduction algorithm

given strictly feasible x, Z, a tolerance $\epsilon > 0$, and a parameter $\nu \geq 1$.

repeat

1. Find primal and dual search directions by solving (36).
2. Plane search. Find p, $q \in \mathbf{R}$ that minimize $\varphi(x + p\delta x, Z + q\delta Z)$.
3. Update $x := x + p\delta x$, $Z := Z + q\delta Z$.

until $\eta(x,Z) \leq \epsilon$.

It can be shown that at each iteration of the algorithm, the potential function decreases by at least a fixed amount:

$$\varphi(x^{(k+1)}, Z^{(k+1)}) \leq \varphi(x^{(k)}, Z^{(k)}) - \delta$$

where $\delta > 0$ does not depend on any problem data at all (including the dimensions). For a proof of this result, see [NN94, §4.5]. Combined with (34) this provides a bound on the number of iterations required to attain a given accuracy ϵ. From (34) we see that $\eta \leq \epsilon$ after at most

$$\frac{\nu\sqrt{2L}\log(\eta^{(0)}/\epsilon) + \psi(x^{(0)}, Z^{(0)})}{\delta}$$

iterations. Roughly speaking and provided the initial value of ψ is small enough, this means it takes no more than $O(\sqrt{L})$ steps to reduce the initial duality gap by a given factor.

Computationally the most demanding step in the algorithm is solving the linear system (36). This can be done by first eliminating δZ from the first equation, solving

$$\bar{A}^T H \bar{A} \delta x = -\bar{A}^T(\rho Z + g) = -\rho f - \bar{A}^T g \tag{37}$$

for δx, and then substituting to find

$$\delta Z = -\rho Z - g - H\bar{A}\delta x.$$

Since $\bar{A}^T \delta Z = 0$, the updated dual point $Z + q\delta Z$ satisfies the dual equality constraints, for any $q \in \mathbf{R}$.

An alternative is to directly solve the larger system (36) instead of (37). This may be preferable when $\bar{A}$ is very large and sparse, or when the equations (37) are badly conditoned.

We refer to the second step in the algorithm as the *plane search* since we are minimizing the potential function over the plane defined by the current points x,Z and the current primal and dual search directions. This plane search can be carried out very efficiently using some preliminary preprocessing. The objective function for the plane search is

$$\begin{aligned} f(p,q) &= \varphi(x+p\delta x, Z+q\delta Z) \\ &= (2L+\nu\sqrt{2L})\log\left(\eta(x,Z)+pZ^T\delta X+q\delta Z^T X\right) \\ &\quad -\sum_{i=1}^{L}\log\left(t_i^2-\|u_i\|^2+2p(t_i\delta t_i-u_i^T\delta u_i)+p^2(\delta t_i^2-\|\delta u_i\|^2)\right) \\ &\quad -\sum_{i=1}^{L}\log\left(w_i^2-\|z_i\|^2+2q(w_i\delta w_i-z_i^T\delta z_i)+q^2(\delta w_i^2-\|\delta z_i\|^2)\right), \end{aligned}$$

where $\delta X = \bar{A}\delta x + \bar{b}$. This function of two variables can be very efficiently minimized if we first compute the coefficients of p, q, p^2 and q^2 in the arguments of the logarithms. Once those coefficients are available, the first and second derivatives of f at any given p and q can be computed very quickly, in $O(L)$ operations, and therefore the minimum of f is readily obtained by a (safe-guarded) Newton method.

We conclude this section by pointing out the analogy between (36) and the systems of equations arising in interior-point methods for LP. We consider the primal-dual pair of LPs

$$\begin{array}{ll} \text{minimize} & f^T x \\ \text{subject to} & c_i^T x + d_i \geq 0, \quad i=1,\ldots,L \end{array}$$

and

$$\begin{array}{ll} \text{minimize} & -\sum_{i=1}^{L} d_i z_i \\ \text{subject to} & \sum_{i=1}^{L} z_i c_i = f \\ & z_i \geq 0, \quad i=1,\ldots,L, \end{array}$$

and solve them as SOCPs with $n_i = 1$, $i=1,\ldots,L$. Using the method outlined above, we obtain

$$\bar{A} = [c_1 \ \cdots \ c_L]^T, \quad \bar{b} = d,$$

and writing $X = \mathbf{diag}\left(c_1^T x + d_1, \ldots, c_L^T x + d_L\right)$, the equation (36) reduces to

$$\begin{bmatrix} \frac{1}{2}X^2 & \bar{A} \\ \bar{A}^T & 0 \end{bmatrix} \begin{bmatrix} \delta z \\ \delta x \end{bmatrix} = \begin{bmatrix} -(\rho/2)X^2 z + Xe \\ 0 \end{bmatrix}, \tag{38}$$

The factor $1/2$ in the first block can be absorbed into δz since only the direction of δz is important, not its magnitude. Also note that $\rho/2 = (L+\nu\sqrt{L})/\eta$. We therefore see that the equations (38) coincide with (one particular variation) of familiar expressions for LP.

4.4 Finding strictly feasible initial points

The algorithm of the previous section requires strictly feasible primal and dual starting points. In this section we discuss two techniques that can be used when primal and/or dual feasible points are not readily available. We first show that any given SOCP can be modified in such a way that it has an obvious dual strictly feasible solution. We then show how to compute a primal feasible point for an SOCP by solving a related problem, known as the *phase-I* problem.

Bounds on the primal variables

As a general guideline, it is easy to find strictly dual feasible points in SOCPs where the primal constraints include explicit bounds on the feasible set. Such bounds can include, for example, componentwise upper and lower bounds $l \le x \le u$, or a norm constraint $\|x\| \le R$. It can be verified that adding explicit bounds results in SOCPs with straightforward dual feasible points.

For example, suppose that we modify the SOCP (1) by adding a bound on the norm of x:

$$\begin{array}{ll} \text{minimize} & f^T x \\ \text{subject to} & \|A_i x + b_i\| \le c_i^T x + d_i, \quad i = 1, \ldots, L \\ & \|x\| \le R. \end{array} \tag{39}$$

If R is large enough, the extra constraint does not change the solution and the optimal value of the SOCP. The dual of the SOCP (39) is

$$\begin{array}{ll} \text{maximize} & -\sum_{i=1}^{L} \left(b_i^T z_i + d_i w_i \right) - R w_{L+1} \\ \text{subject to} & \sum_{i=1}^{L} \left(A_i^T z_i + c_i w_i \right) + z_{L+1} = f \\ & \|z_i\| \le w_i, \quad i = 1, \ldots, L+1. \end{array} \tag{40}$$

Strictly feasible points for (40) can be easily calculated as follows. For $i = 1, \ldots, L$, we can take any z_i and $w_i > \|z_i\|$. The variable z_{L+1} then follows from the equality constraint in (40), and for w_{L+1} we can take any number greater than $\|z_{L+1}\|$.

This idea of adding bounds on the primal variable is a variation on the big-M method in linear programming.

Phase-I method

A primal strictly feasible point can be computed by solving the SOCP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \|A_i x + b_i\| \le c_i^T x + d_i + t, \quad i = 1, \ldots, L \end{array} \tag{41}$$

in the variables x and t. If (x, t) is feasible in (41), and $t < 0$, then x satisfies $\|A_i x + b_i\| < c_i^T x$, *i.e.*, it is strictly feasible for the original SOCP (1). We can therefore find a strictly feasible x by solving (41), provided the optimal value $t^\star$ of the SOCP (41) is negative. If $t^\star > 0$, the original SOCP (1) is infeasible.

Note that it is easy to find a strictly feasible point for the SOCP (41). One possible choice is

$$x = 0, \quad t > \max_i \|b_i\| - d_i.$$

The dual of the SOCP (41) is

$$
\begin{array}{ll}
\text{maximize} & \sum_{i=1}^{L} \left(b_i^T z_i + d_i w_i \right) \\
\text{subject to} & \sum_{i=1}^{L} \left(A_i^T z_i + c_i w_i \right) = 0 \\
& \sum_{i=1}^{L} w_i = 1 \\
& \|z_i\| \leq w_i, \quad i = 1, \ldots, L.
\end{array}
\tag{42}
$$

If a strictly feasible (z, w) for (42) is available, one can solve the phase-I problem by applying the primal-dual algorithm of the previous section to the pair of problems (41), and (42). If no strictly feasible (z, w) for (42) is available, one can add an explicit bound on the primal variable as described above.

4.5 Performance in practice

A C-implementation of the potential reduction method described in §4.3 is available via the WWW[1] and numerical experiments with the algorithm are reported in the documentation of the code [LVB97]. Our experience with the method is consistent with the practical behavior observed in many similar methods for linear or semidefinite programming: the number of iterations is only weakly dependent on the problem dimensions (n, n_i, L), and typically lies between 5 and 50 for a very wide range of problem sizes.

We can therefore say that for practical purposes the cost of solving an SOCP is roughly equal to the cost of solving a modest number (5–50) of systems of the form (37). If no special structure in the problem data is exploited, the cost of solving the system is $O(n^3)$, and the cost of forming the system matrix is $O(n^2 \sum_{i=1}^{L} n_i)$. In practice, special problem structure (*e.g.*, sparsity) often allows forming the equations faster, or solving the systems (37) and (36) more efficiently.

We close this section by pointing out a few possible improvements. The most popular interior-point methods for linear programming share many of the features of the potential reduction method we presented here, but differ in three respects (see [Wri97]). First, they treat the primal and dual problems more symmetrically (for example, the diagonal matrix X^2 in (38) is replaced by XZ^{-1}). A second difference is that common interior-point methods for LP are one-phase methods that allow an infeasible starting point. Finally, the asymptotic convergence of the method is improved by the use of predictor steps. These different techniques can all be extended to SOCP. In particular, Nesterov and Todd [NT94] and Adler and Alizadeh [AA95] have developed extensions of the symmetric primal-dual LP methods to SOCP, and an implementation will be made available in the next version of SDPPACK [AHNO97].

[1] At `http://www-isl.stanford.edu/people/boyd/SOCP.html`

5 Conclusions

Second-order cone programming is a problem class that lies between linear (or quadratic) programming and semidefinite programming. Like LP and SDP, SOCPs can be solved very efficiently by primal-dual interior-point methods (and in particular, far more efficiently than by treating the SOCP as an SDP). Moreover, a wide variety of engineering problems can be formulated as second-order cone problems.

Acknowledgments

The data for the numerical antenna array weight design example described in §3.1 were kindly supplied by H. L. Southall, USAF Rome Lab., Electromagnetics & Reliability Directorate, Hanscom AFB, USA. The examples are taken from the course notes for EE364 [BV97].

We thank Michael Grant for many useful comments and suggestions.

References

[AA95] I. Adler and F. Alizadeh. Primal-dual interior point algorithms for convex quadratically constrained and semidefinite optimization problems. Technical Report RRR 46-95, RUTCOR, Rutgers University, 1995.

[AA97] E. Andersen and K. Andersen. *APOS user's manual for QCOPT ver 1.00 Beta*. EKA consulting, May 1997. `http://www.samnet.ou.dk/ eda/`.

[ABBTZ92] W. Achtziger, M. Bendsoe, A. Ben-Tal, and J. Zowe. Equivalent displacement based formulations for maximum strength truss topology design. *Impact of Computing in Science and Engineering*, 4(4):315–45, December 1992.

[ACO94] K. Andersen, E. Christiansen, and M. Overton. Computing limits loads by minimizing a sum of norms. Technical Report 30, Institut for Matematik og Datalogi, Odense Universitet, September 1994.

[AHNO97] F. Alizadeh, J. P. Haeberly, M. V. Nayakkankuppam, and M. L. Overton. SDPPACK *User's Guide, Version 0.8 Beta.* NYU, March 1997.

[And96] K. Andersen. An efficient Newton barrier method for minimizing a sum of Euclidean norms. *SIAM J. on Optimization*, 6(1):74–95, 1996.

[BB91] S. Boyd and C. Barratt. *Linear Controller Design: Limits of Performance*. Prentice-Hall, 1991.

[BBTZ94] M. Bendsoe, A. Ben-Tal, and J. Zowe. Optimization methods for truss geometry and topology design. *Structural Optimization*, 7:141–159, 1994.

[BHM96] M. Buss, H. Hashimoto, and J.B. Moore. Dextrous hand grasping force optimization. *IEEE Transactions on Robotics and Automation*, 12(3):406–18, June 1996.

[BTB93] A. Ben-Tal and M. P. Bendsøe. A new method for optimal truss topology design. *SIAM Journal on Optimization*, 3:322–358, 1993.

[BTN92] A. Ben-Tal and A. Nemirovskii. Interior point polynomial-time method for truss topology design. Technical report, Faculty of Industrial Engineering and Management, Technion Institute of Technology, Haifa, Israel, 1992.

[BTN95] A. Ben-Tal and A. Nemirovski. Optimal design of engineering structures. *Optima*, pages 4–9, October 1995.

[BTN96] A. Ben-Tal and A. Nemirovski. Robust convex optimization. Technical report, Faculty of Industrial Engineering and Management, Technion, December 1996.

[BV97] S. Boyd and L. Vandenberghe. Introduction to convex optimization with engineering applications. Course Notes, 1997. `http://www-leland.stanford.edu/class/ee364/`.

[CGGS96] S. Chandrasekaran, G. H. Golub, M. Gu, and A. H. Sayed. Efficient algorithms for least square type problems with bounded uncertainties. Technical Report SCCM-96-16, Stanford University, 1996.

[Che82] E. W. Cheney. *Introduction to Approximation Theory.* Chelsea Publishing Company, New York, second edition, 1982.

[CO90] F.-T. Cheng and D. E. Orin. Efficient algorithm for optimal force distribution — the compact-dual LP method. *IEEE Transactions on Robotics and Automation*, 6(2):178–187, 1990.

[CO94] A. R. Conn and M. L. Overton. Primal-dual interior-point method for minimizing a sum of Euclidean distances. Draft, 1994.

[dH93] D. den Hertog. *Interior Point Approach to Linear, Quadratic and Convex Programming.* Kluwer, 1993.

[DO96] D. R. Dreyer and M. L. Overton. Two heuristics for the Steiner tree problem. Technical Report 724, New York University Computer Science Department, 1996.

[EL] L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM J. on Matrix Analysis and Applications.* To appear.

[HBVL97] A. Hansson, S. Boyd, L. Vandenberghe, and M. Lobo. Optimal linear static control with moment and yield objectives. In *Proceedings of the 37th IEEE Conference on Decision and Control*, San Diego, California, 1997. Submitted for possible presentation.

[JL87] J. W. Jameson and L. J. Leifer. Automatic grasping: an optimization approach. *IEEE Transactions on Systems, Man and Cybernetics*, 17(5):806–814, 1987.

[KK90] C. A. Klein and S. Kittivatcharapong. Optimal force distribution for the legs of a walking machine with friction cone constraint. *IEEE Transactions on Robotics and Automation*, 6(1):73–85, 1990.

[LB97] H. Lebret and S. Boyd. Antenna array pattern synthesis via convex optimization. *IEEE Trans. on Signal Processing*, 45(3):526–532, March 1997.

[Leb94] H. Lebret. *Synthèse de diagrammes de réseaux d'antennes par optimisation convexe.* PhD thesis, Université de Rennes I, November 1994.

[Leb95] H. Lebret. Antenna pattern synthesis through convex optimization. In Proc. SPIE 2563 Franklin T. Luk, editor, *Advanced Signal Processing Algorithms*, pages 182–192, 1995.

[LVB97] M. S. Lobo, L. Vandenberghe, and S. Boyd. SOCP*: Software for Second-Order Cone Programming.* Information Systems Laboratory, Stanford University, 1997.

[NN94] Yu. Nesterov and A. Nemirovsky. *Interior-point polynomial methods in convex programming*, volume 13 of *Studies in Applied Mathematics.* SIAM, Philadelphia, PA, 1994.

[NS96] A. Nemirovskii and K. Scheinberg. Extension of Karmarkar's algorithm onto convex quadratically constrained quadratic problems. *Mathematical Programming*, 72:273–289, 1996.

[NT94] Yu. E. Nesterov and M. J. Todd. Self-scaled cones and interior-point methods in nonlinear programming. Technical Report 1091, Cornell University, April 1994.

[OEL96] F. Oustry, L. El Ghaoui, and H. Lebret. Robust solutions to uncertain semidefinite programs. *SIAM J. on Optimization*, 1996. Submitted.

[OS70] A. V. Oppenheim and R. W. Schafer. *Digital Signal Processing.* Prentice-Hall, Englewood Cliffs, N. J., 1970.

[SA92] P. R. Sinha and J. A. Abel. A contact stress model for multifingered grasps of rough objects. *IEEE Transactions on Robotics and Automation*, 8(1):7–22, 1992.

[SSO95] H.L. Southall, J.A. Simmers, and T.H. O'Donnell. Direction finding in phased arrays with a neural network beamformer. *IEEE Transactions on Antennas and Propagation*, 43(12):1369–74, December 1995.

[Van97] R. J. Vanderbei. *Linear Programming: Foundations and Extensions.* Kluwer Academic Publishers, 1997.

[VB96] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, March 1996.

[VBW] L. Vandenberghe, S. Boyd, and S.-P. Wu. Determinant maximization with linear matrix inequality constraints. *SIAM J. on Matrix Analysis and Applications.* To appear.

[WBV96] S.-P. Wu, S. Boyd, and L. Vandenberghe. FIR filter design via semidefinite programming and spectral factorization. In *Proc. IEEE Conf. on Decision and Control*, pages 271–276, 1996.

[Wri97] S. J. Wright. *Primal-Dual Interior-Point Methods.* SIAM, Philadelphia, 1997.

[XY] G. Xue and Y. Ye. An efficient algorithm for minimizing a sum of Euclidean norms with applications. *SIAM J. on Optimization.* To appear.

[Ye91] Y. Ye. An $O(n^3L)$ potential reduction algorithm for linear programming. *Mathematical Programming*, 50:239–258, 1991.