Applications of Second Order Cone Programming

Lisa Turner

1 Introduction

Boyd and Vandenberghe (2004) define an optimisation problem as one which has the form:

\[
\text{Minimise } f_0(x), \quad \text{subject to } f_i(x) \geq b_i, \quad i = 1, \ldots, m,
\]

where \(x = (x_1, \ldots, x_n)\) is the optimisation variable, \(f_0 : \mathbb{R}^n \rightarrow \mathbb{R}\) is the objective function, \(f_i : \mathbb{R}^n \rightarrow \mathbb{R}\) for \(i = 1, \ldots, m\) are the constraint variables and \(b_1, \ldots, b_m\) are the bounds. A vector \(x^*\) is called optimal if it has the smallest possible objective value, while still satisfying all the constraints.

Karmarkar (1984) proposed his projective method for solving linear programming problems. Refinement of this method meant that solutions could be found to problems which were previously thought to be untouchable. Nesterov and Nemirovski (1994) showed that, in principle, any convex optimisation problem could be provided with a self concordant barrier and hence could be solved in polynomial time.

A significant special case of the problems which could be solved were those whose constraints were given by semidefinite cones. A Semidefinite Program (SDP) is an optimisation over the intersection of an affine set and cone of positive semidefinite matrices (Alizadeh and Goldfarb, 2001). Cone programming is discussed more in Section 3. Within semidefinite programming there is a smaller set of problems which can be modelled as Second Order Cone Programs (SOCPs), discussed more in Section 4. These have a wide range of applications, some of which are discussed in Section 5, and can still be solved efficiently using interior point methods. Lobo et al. (1998) justifies that the study of SOCPs in their own right is warranted. Software for solving SOCPs is now readily available, see Mittelmann (2012) for an overview on existing code.

2 Cones

A set \(K\) is called a cone if for every set \(x \in K\) and \(\theta \geq 0\) then \(\theta x \in K\). It is said to be convex if for every \(x_1, x_2 \in K\) and \(\theta_1, \theta_2 \geq 0\) then \(\theta_1 x_1 + \theta_2 x_2 \in K\) holds. Furthermore, the cone is pointed if it contains no line. This is equivalent to \(x \in K, -x \in K \Rightarrow x = 0\), which means that the origin is the smallest non empty face of the cones close. All of the cones in Figure 1 are convex pointed cones. They are also examples of proper cones which means they are closed, solid, pointed and convex. Proper cones have the useful property that they are invariant to linear transformations.

The cone shown in Figure 1b is a Lorentz cone, which is also known as an ice cream cone or quadratic cone, in \(\mathbb{R}^3\) and is given mathematically by the set of points which satisfy \(x_3 \geq \sqrt{x_1^2 + x_2^2}\). This is an example of a special class of convex cones called second order cones (SOC), which can be generalised to higher dimensions. The second order cone is the set of points in \(\mathbb{R}^n\) satisfying the inequality:

\[
x_n \geq \sqrt{x_1^2 + \ldots + x_{n-1}^2}.
\]
Andersen et al. (2003) define a general cone optimisation problem to be:

\[
(P) \text{ Minimise } c^T x,
\]
\[
\text{subject to } Ax = b, \ x \in K,
\]

where \( K \) is a proper cone and all other quantities have conforming dimension and \( A \) is of full row rank.

The primal solution \((P)\) given by (3) is said to be feasible if it has at least one feasible solution, which satisfies all the constraints of \((P)\).

The dual cone of \( K \) is given by \( K^* = \{ s : s^T \geq 0, \forall x \in K \} \). When a cone program is both self-dual and homogeneous, effective primal dual interior points methods exist for solving them. If a cone, \( K \), is pointed, closed and convex then it is also self-dual if \( K = K^* \), and homogeneous if for any \( x, s \in int(K) \) then:

\[
\exists B \in \mathbb{R}^{n \times m} : B(K) = K, \ Bx = s.
\]

Güler (1996) states that there are only 5 types of cones which are both homogeneous and irreducible self-dual cones. These include second order cones and the cone of semidefinite symmetric matrices. An overview of existing interior points methods for solving these types of problems is given in Nemirovski and Todd (2008).

4 Second Order Cone Programs

A second order cone program (SOCP) is a conic program over the direct product of a finite number of cones:

\[
\text{Minimise } f^T x,
\]
\[
\text{subject to } \|A_i x + b_i\| \leq c_i^T x + d_i, \ i = 1, \ldots, m.
\]

By letting \( A_i x + b_i = y_i \) and \( c_i^T x + d_i = z_i \), the constraints become standard SOCP constraints:

\[
A_i x + b_i = y_i, \ c_i^T x + d_i = z_i, \ i = 1, \ldots, m,
\]
\[
\|y_i\| \leq z_i, \ i = 1, \ldots, m.
\]

Linear programming problems along with convex problems containing hyperbolic, quadratic and norm constraints can be transformed into SOCPs (Lobo et al., 1998).
5 Applications

Second Order Cone Programs have a wide range of applications. This section will look at several examples within different areas of science where SOCPs are used. For more examples of problems which can be modelled as SOCPs, see Lobo et al. (1998) and Kuo and Mittelman (2004) where the following examples can be found.

5.1 Robust Linear Programming

Robust optimisation can be used when there is some uncertainty within the optimisation problem. When the data, rather than being known exactly, belongs to a given uncertainty set $U$ and the constraints must hold for all values of $U$ in an optimisation problem, this is known as a robust optimisation problem. Examples of types of uncertainty that exist can be found in Ben-Tal and Nemiroviski (1998). Often, the type of uncertainty set, $U$, is an ellipsoid, or an intersection of finitely many ellipsoids. The justification for choosing an ellipsoid is given in Ben-Tal and Nemiroviski (1998) and can often provide reasonable approximations to more complicated uncertainty sets. This report will only look at the robust counterpart of an uncertain Linear Program (LP). Due to the wide practical use of linear programming, its robust counterpart is of primary interest. With ellipsoidal or $\cap$-ellipsoidal uncertainty this is an SOCP. Robust counterparts exist to many other types of convex optimisation problems including robust quadratic programming, with ellipsoidal uncertainty, which can be modelled as an SDP (Ben-Tal and Nemiroviski, 1998).

The LP:

$$\begin{align*}
\text{Minimise} \; c^T x, \\
\text{subject to} \; a_i x \geq b_i, \; i = 1, \ldots, m,
\end{align*}$$  \tag{5}

can be modelled as a robust LP when there is some form of ellipsoidal uncertainty in the parameters $c, a_i, b_i$. The simplified problem, given in Lobo et al. (1998) where $c$ and $b_i$ are fixed is considered here. Assume that $a_i$ is known to lie in given ellipsoids:

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

where $P = P^T$ is a positive semidefinite matrix. In the worst case, the constraint must be satisfied for all possible values of the constraint parameters which leads to the robust LP:

$$\begin{align*}
\text{Minimise} \; c^T x, \\
\text{subject to} \; a_i^T x \geq b_i, \; \forall a_i \in \mathcal{E}, \; i = 1, \ldots, m.
\end{align*}$$  \tag{6}

This constraint can be expressed as $\max \{ a_i^T x \mid a_i \in \mathcal{E} \} = \bar{a}_i^T x + \|P_i x\| \leq b_i$ which is a second order cone constraint, and hence the robust linear programming problem is an SOCP:

$$\begin{align*}
\text{Minimise} \; c^T x, \\
\text{subject to} \; \|P_i x\| \leq b_i - \bar{a}_i^T x.
\end{align*}$$  \tag{7}

This formulation discourages a large $x$ when there is large uncertainty in the parameter $a_i$.

The robust linear programming problem can be modified such that there is a probability exceeding $\eta$, with $\eta \geq 0.5$, that each constraint should hold. This can still be modelled as an SOCP. The probability constraint is $P(a_i^T x \leq b_i) \geq \eta$, which can be written, with $u = a_i^T x$ to simplify notation, as:

$$P \left( \frac{u - \bar{u}}{\sqrt{\sigma}} \leq \frac{b_i - \bar{u}}{\sqrt{\sigma}} \right) \geq \eta.$$  \tag{8}
If the parameters are Gaussian random vectors, with mean $\bar{a}_i$ and covariance matrix $\Sigma_i$, then $\frac{u - \bar{a}}{\sqrt{\sigma}}$ is a standard normal variable and hence (8) is the same as $\Phi\left(\frac{b_i - \bar{a}}{\sqrt{\sigma}}\right) \geq \eta$. Applying the probit function and rearranging gives the constraint:

$$\tilde{u} + \Phi^{-1}(\eta)\sqrt{\sigma} \leq b_i,$$

and finally, re-expressing in terms of $\tilde{a}_i$ and $\Sigma_i$ gives a SOC constraint, as long as $\eta \geq 0.5$. Therefore, the robust LP, where uncertain constraints must be met with a certain probability, can be written as the following SOCP:

$$\begin{align*}
\text{Minimise} \quad c^T x, \\
\text{subject to} \quad \tilde{a}_i^T x + \Phi^{-1}(\eta) \left\| \Sigma_i^{1/2} x \right\| \leq b_i, \quad i = 1, \ldots, m.
\end{align*}$$

A special case of this type of robust optimisation is used for portfolio selection where the amount of money available is $M$ and there are $n$ stocks in which to invest. The idea is to minimise risk while still returning at least some given amount of money.

5.2 The Facility Location Problem

Within Management Science, a good example of where SOCPs are used is the facility location problem. Suppose a company wants to build a new facility among existing facilities, so as to minimise the weighted Euclidean distances between new and existing facilities. Assume there are $m$ existing facilities and $d = 2$ is the number of dimensions, although this can easily be extended to more dimensions. Let the co-ordinates of the $i^{th}$ existing facility be given by $(a_{i1}, a_{i2})$ and the weight between the $i^{th}$ facility and the new facility be given by $w_i$. The co-ordinates of the new facility are given by $(x_1, x_2)$. This problem can be modelled using the following optimisation problem:

$$\begin{align*}
\text{Minimise} \quad m \sum_{i=1} w_i \sqrt{2 \sum_{j=1} (x_j - a_{ij})^2}. \\
\text{subject to} \quad \sqrt{2 \sum_{j=1} (x_j - a_{ij})^2} \leq t_i, \quad i = 1, \ldots, m.
\end{align*}$$

To transform (10) to a standard SOCP problem, let $u_{ij} = x_j - a_{ij}$ for $i = 1, \ldots, m$ and $j = 1, 2$, which represents the difference in distance between an existing facility and the new facility in the $j$ direction and add the constraints:

$$\sqrt{2 \sum_{j=1} (x_j - a_{ij})^2} \leq t_i, \quad i = 1, \ldots, m.$$

These ensure the weighted Euclidean distance is minimised. Furthermore define $x_j$ by adding the constraints:

$$u_{11} + a_{11} = u_{i1} + a_{i1}, \quad u_{12} + a_{12} = u_{i2} + a_{i2}, \quad \text{for } i = 2, \ldots, m.$$

The facility location problem can then be modelled as the SOCP:

$$\begin{align*}
\text{Minimise} \quad m \sum_{i=1} w_i t_i, \\
\text{subject to} \quad u_{ij} - u_{ij} = a_{ij} - a_{ij}, \quad i = 1, \ldots, m, \quad j = 1, 2, \\
\sqrt{2 \sum_{j=1} u_{ij}^2} \leq t_i, \quad i = 1, \ldots, m, \\
t_i \geq 0, \quad i = 1, \ldots, m.
\end{align*}$$
The problem of modelling \( n \) new facilities, among \( m \) existing facilities can also be modelled as an SOCP, where \( a \) and \( w \) are as above and the objective function becomes:

\[
\text{Minimise } \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} \sqrt{\sum_{k=1}^{2} (x_{jk} - a_{ik})^2} + \sum_{j=1}^{n} \sum_{b=1}^{j} v_{jb} \sqrt{\sum_{k=1}^{2} (x_{jk} - x_{bk})^2},
\]

(12)

where \( v_{jb} \) is the distance between new and new facilities.

To convert this to an SOCP, define the new variable \( s_{ij} \geq 0 \) for \( i = 1, \ldots, m \) and \( j = 1, \ldots, n \) as the distance between existing and new facilities and \( t_{jb} \geq 0 \) for \( j = 1, \ldots, n \) and \( b < j \) as the distance between two new facilities. Furthermore, define:

- \( x_{jk} - a_{ik} = u_{ijk} \) for \( i = 1, \ldots, m, \ j = 1, \ldots, n \) and \( k = 1, 2 \).
- \( x_{jk} - x_{bk} = z_{jbk} \) for \( j = 1, \ldots, n, \ b < j \) and \( k = 1, 2 \).

Then the facility location problem with \( n \) new facilities can be modelled as the SOCP:

\[
\text{Minimise } \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} s_{ij} + \sum_{j=1}^{n} \sum_{b=1}^{j} v_{jb} t_{jb},
\]

subject to

\[
\sqrt{\sum_{k=1}^{2} u_{ijk}^2} \leq s_{ij} , \ i = 1, \ldots, m , \ j = 1, \ldots, n,
\]

\[
\sqrt{\sum_{k=1}^{2} z_{jbk}^2} \leq t_{jb} , \ j = 1, \ldots, m , \ b < j,
\]

\[
u_{1jk} - u_{ijk} = a_{ik} - a_{ik} , \ i = 1, \ldots, m , \ j = 1, \ldots, n , \ k = 1, 2,
\]

\[
z_{j1k} - z_{jbk} = x_{bk} - x_{1k} , \ j = 1, \ldots, m , \ b < j , \ k = 1, 2.
\]

5.3 Equilibrium of a System of Piecewise Linear Springs

There is a wide range of applications for optimisation using second order cone programming in engineering. One of these describes an optimisation problem of the mechanical system consisting of a hanging chain with \( N \) links where the first and last node are fixed. At each node, there is a weight \( m_j \) hanging from it and the aim is to minimise the energy in the system of piecewise linear springs. This is given by the objective function which is the minimum of the potential energy in the spring plus the sum of the gravitational potential energy at each node:

\[
\text{Minimise } \sum_{j=1}^{N} m_j g y_j + \frac{k}{2} \| t \|.
\]

(14)

The constraints are:

\[
\|(x_j, y_j) - (x_{j-1}, y_{j-1})\| - l_0 \leq t_j , \ j = 1, \ldots, N,
\]

\[
(x_0, y_0) = (a_1, a_2), \ (x_N, y_N) = (b_1, b_2),
\]

\[
t \geq 0,
\]

where \( g \) is acceleration due to gravity, \( k \) is the stiffness constant found from Hookes law, \( l_0 \) is the rest length of each spring and \( t_j \) is the upper bound on the springs energy of the \( j^{th} \) spring. To write this as an SOCP, define \( \| t \| = 2uv \) where \( u = 1 \) and \( v \geq 0 \), and \( s_j = t_j + l_0 \). Furthermore, define \( e_j = x_j - x_{j-1} \) and \( f_j = y_j - y_{j-1} \) for \( j = 1, \ldots, N \) as the difference between the \( j^{th} \) and \((j - 1)^{th}\) node in the \( x \) and
Then \( \sum_j e_j = b_1 - a_1 \) and \( \sum_j f_j = b_2 - a_2 \) are simply the differences between the first and last node in the \( x \) and \( y \) directions. The objective function can then be rewritten by firstly considering the position of the nodes in the \( y \) direction in terms of \( f_j \) and then rearranging to give:

\[
\sum_j m_j g y_j + kv = g \sum_j m_j \left( \sum_{i=1}^j f_i + a_2 \right) + kv = g \sum_{i=1}^N \left( \sum_{j=1}^N m_j \right) f_j + g \sum_{j=1}^N a_2 + kv.
\]

Hence, the problem can be written as an SOCP in the following form:

\[
\text{Minimise } g \sum_{i=1}^N \left( \sum_{j=1}^N m_j \right) f_j + g \sum_{j=1}^N a_2 + kv, \tag{15}
\]

subject to

\[
\sum_j e_j = b_1 - a_1 , \quad \sum_j f_j = b_2 - a_2 ,
\]

\[
s_j - t_j = l_0 , \quad j = 1, \ldots, N ,
\]

\[
\sqrt{e_j^2 + f_j^2} \leq s_j , \quad j = 1, \ldots, N ,
\]

\[
\|t\|_2^2 \leq 2uv , \quad v \geq 0 , \quad u = 1 .
\]

Many other problems in engineering can be modelled as SOCPs, including in FIR filtering, grasping force optimisation and antenna weight design (Lobo et al., 1998).

6 Conclusion

This report has introduced cone programming and in particular SOCPs. Effective interior points methods are available for solving SOCPs because their constraints are second order cones which are both self-dual and homogeneous. As a result, they have been applied to a wide range of scientific areas. The second half of this report gave several motivating examples of how problems can be formulated as SOCPs within a range of scientific areas showing how widely applicable they are, and hence their importance within both conic programming and optimisation programming as a whole.

References


