Convex Quadratically-Constrained Quadratic Programming .

Consider the problem

\[
\begin{align*}
\min_y & \quad f_0(y) \\
n_i(y) & \leq 0, \quad i = 1, \ldots, n,
\end{align*}
\]

where each \( f_i \) is a convex quadratic function of \( y \in \mathbb{R}^m \).

We can assume the objective is linear, so we have

\[
\begin{align*}
\max_y & \quad b^T y \\
n_i(y) & \leq 0, \quad i = 1, \ldots, n,
\end{align*}
\]

where each \( f_i(y) = y^T C_i y - d_i^T y - \epsilon_i \) with \( C_i \) psd. We can write \( C_i = G_i^T G_i \) where \( G_i \in \mathbb{R}^{r_i \times m} \).

Then

\[
\begin{align*}
f_i(y) \leq 0 & \iff d_i^T y + \epsilon_i \geq (G_i y)^T (G_i y) \\
& \iff M_i = \begin{bmatrix} d_i^T y + \epsilon_i & (G_i y)^T \\ G_i y & I \end{bmatrix} \succeq 0
\end{align*}
\]

(this follows from Schur complements, but we have to consider the two cases when \( d_i^T y + \epsilon \) is zero and when it is positive). So our problem can be formulated as

\[
\max_y \quad b^T y \\
\text{Diag} (M_1, \ldots, M_n) \succeq 0.
\]

Alternatively

\[
\begin{align*}
f_i(y) \leq 0 & \iff (d_i^T y + \epsilon_i + 1)^2 \geq (d_i^T y + \epsilon_i - 1)^2 + (2G_i y)^T (2G_i y) \\
& \iff (d_i^T y + \epsilon_i + 1) \geq \left\| \frac{d_i^T y + \epsilon_i - 1}{2G_i y} \right\|_2 \\
& \iff d_i^T y + \epsilon_i + 1 \geq \frac{1}{d_i^T y + \epsilon_i + 1} \left\| \frac{d_i^T y + \epsilon_i - 1}{2G_i y} \right\|_2^2 \\
& \iff W_i = \begin{bmatrix} d_i^T y + \epsilon_i + 1 & d_i^T y + \epsilon_i - 1 & (2G_i y)^T \\ d_i^T y + \epsilon_i - 1 & d_i^T y + \epsilon_i + 1 & 0 \\ 2G_i y & 0 & (d_i^T y + \epsilon_i + 1) I \end{bmatrix} \succeq 0
\end{align*}
\]

(some steps of this argument need to ensure that \( d_i^T y + \epsilon + 1 \) is positive, whichever the direction of the implications). So we could replace all the \( M_i \)s with \( W_i \)s in the formulation above.
However, this analysis also shows another formulation of the problem in dual conic programming form as
\[
\max_y \quad b^T y \\
\quad d_i^T y + \epsilon_i + 1 \geq \left\| \frac{d_i^T y + \epsilon_i - 1}{2G_i y} \right\|_2, \quad \text{all } i,
\]
a second-order cone programming problem, which will typically be much more efficient to solve.

In fact, we have used

Proposition 1

\[
\begin{bmatrix} \gamma & v^T \\ v & \gamma I \end{bmatrix} \succeq 0 \iff \gamma \geq \|v\|_2.
\]

**Truss topology design** is the problem of choosing where and what size rods to use in a framework to support one or more loads.

For more details in what follows, see Ben-Tal & Nemirovskii, Lectures on Modern Convex Optimization.

Suppose we put a rod in potential link \( j \) with cross-sectional area \( y_j \). This gives a stiffness matrix

\[
A(y) = \sum y_j b_j b_j^T \succeq 0
\]

and “determines” displacements \( d \) (indexed by 2 or 3 components for each free node) that can support a load vector \( f \) by

\[
A(y)d = f \quad \text{(Hooke’s Law)}.
\]

We may also have constraints \( a \leq y \leq b \) and \( l^T y \leq w \) and we also want the “maximum stiffness” or “minimum compliance” (proportional to the work done)

\[
\min f^T d.
\]

So the problem is

\[
\min \quad f^T d \\
A(y)d = f \\
l^T y \leq w, \quad a \leq y \leq b.
\]

This is nonlinear in the variables \( d \) and \( y \). If \( A(y) \succ 0 \), we could eliminate \( d \) and the first set of constraints and minimize \( f^T A(y)^{-1} f \) to get

\[
\min \quad \begin{bmatrix} f^T \\ A(y) \end{bmatrix} \succeq 0, \quad a \leq y \leq b, \quad l^T y \leq w.
\]

In fact, this works even if \( A(y) \) might be singular at the solution.
Proposition 2 Suppose $A \succeq 0$. Then

(i) $f^T d$ is the same for all solutions to $Ad = f$;

(ii) $\min_{Ad = f} f^T d = \min \frac{\eta}{f^T A} \succeq 0$.

Proof:

(i) Suppose $Ad_1 = Ad_2 = f$. Then $f^T d_1 = d_2^T A d_1 = d_1^T A d_2 = f^T d_2$.

(ii) Suppose first $f \not\in \text{Range}(A)$, so the left-hand minimum is $\infty$. Since there is no $d$ with $Ad = f$, so there exists $v$ with $Av = 0$ and $f^T v < 0$. Then

$$
\begin{bmatrix}
\beta \\
v
\end{bmatrix}^T
\begin{bmatrix}
\eta & f^T \\
f & A
\end{bmatrix}
\begin{bmatrix}
\beta \\
v
\end{bmatrix} = \eta \beta^2 + 2 f^T v \beta < 0
$$

for any $\eta$ by choosing $\beta > 0$ small enough. Thus the right-hand minimum is also $\infty$.

Now assume there is some $d$ with $Ad = f$. Then the left-hand minimum is $f^T d$ for such a $d$. Also,

$$
\begin{bmatrix}
d^T A & d^T A \\
Ad & Ad
\end{bmatrix} = \begin{bmatrix}
d^T \\
I
\end{bmatrix} A \begin{bmatrix} d & I \end{bmatrix} \succeq 0.
$$

So we can choose $\eta = d^T A = f^T d$, and the right-hand minimum is at most the left-hand minimum.

Also, if $\begin{bmatrix} \eta & f^T \\ f & A \end{bmatrix} \succeq 0$, then

$$
0 \leq \begin{bmatrix}
1 & -d \\
-f & A
\end{bmatrix}^T
\begin{bmatrix}
\eta & f^T \\
f & A
\end{bmatrix}
\begin{bmatrix}
1 \\
-d
\end{bmatrix}
= \eta - 2 f^T d + d^T A d
= \eta - f^T d.
$$

So the right-hand minimum is at least the left-hand minimum. $\square$

So the TTD problem can be formulated as

$$
\min \eta
\begin{bmatrix}
\eta & f^T \\
f & A(y)
\end{bmatrix} \succeq 0
\quad a \leq y \leq b, \quad l^T y \leq w.
$$

In fact, it can be formulated as a linear programming problem!! (See Ben-Tal & Nemirovskii.)

But in practice, the truss has to withstand different load vectors and then we got the robust TTD problem with constraints

$$
\begin{bmatrix}
\eta & f_i^T \\
f_i & A(y)
\end{bmatrix} \succeq 0, \quad i = 1, \ldots, n,
$$

not known to be reducible to linear programming (although it can in fact be reduced to a second-order cone programming problem).
Consider the problem
\[
\max \quad b^T y \\
\quad a_j^T y \leq c_j, \quad j = 1, \ldots, n,
\]
where some or all of the \((a_j; c_j)\)s are not known exactly. Change variables to \((y; -1)\) to get the constraints
\[
a_j^T y \leq 0, \quad \text{for all } a_j \in \mathcal{E}_j, \quad j = 1, \ldots, k, \\
a_j^T y \leq c_j, \quad j = k + 1, \ldots, n,
\]
where the last “certain” set of constraints forces the final component of \(y\) to be \(-1\). This is a semi-infinite set of linear constraints (infinite number of constraints in a finite number of variables).

Assume each \(\mathcal{E}_j\) is “ellipsoid-like”, i.e., of the form \(\{\bar{a}_j + G_j u_j : \|u_j\|_2 \leq 1\}\). Consider some \(j = 1, \ldots, k\). Then
\[
a_j^T y \leq 0 \quad \text{all } a_j \in \mathcal{E}_j \\
\iff \max_{a_j \in \mathcal{E}_j} a_j^T y \leq 0 \\
\iff \max_{\|u_j\| \leq 1} (\bar{a}_j + G_j u_j)^T y \leq 0 \\
\iff \bar{a}_j^T y + \max_{\|u_j\| \leq 1} u_j^T (G_j^T y) \leq 0 \\
\iff \bar{a}_j^T y + \|G_j^T y\|_2 \leq 0.
\]

So the robust linear programming problem is equivalent to the dual form conic programming problem
\[
\max \quad b^T y \\
-\bar{a}_j^T y \geq \|G_j^T y\|_2, \quad j = 1, \ldots, k, \\
\quad a_j^T y \leq c_j, \quad j = k + 1, \ldots, n.
\]
This is a SOCP (second order conic programming) problem.