Interior-Point Methods for Second-Order-Cone and Semidefinite Programming

Robert J. Vanderbei

Joint work with H. Yurttan Benson
Real-world examples by J.O. Coleman, Naval Research Lab

Outline

- Introduction and Review of Problem Classes: LP, NLP, SOCP, SDP
- Formulating SOCPs as Smooth Convex NLPs
- Formulating SDPs as Smooth Convex NLPs
- Applications and Computational Results

Introduction and Review of Problem Classes: LP, NLP, SOCP, SDP

Traditional Families of Optimization Problems

Linear Programming (LP)

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0. \end{array}$$

Smooth Convex Nonlinear Programming (NLP)

$$\begin{array}{lll} \text{minimize} & f(x) & \\ \text{subject to} & h_i(x) = 0, & i \in \mathcal{E}, \\ & h_i(x) \geq 0, & i \in \mathcal{I}. \end{array}$$

We assume that

- h_i's in equality constraints are affine;
- h_i's in inequality constraints are concave;
- f is convex;
- All are twice continuously differentiable.

Two New Families of Optimization Problems

Second-Order Cone Programming (SOCP)

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

Here, A_i is a $k_i \times n$ matrix, b_i is a k_i-vector, c_i is an n-vector, and d_i is a scalar.

Semidefinite Programming (SDP)

$$\begin{array}{ll} \text{minimize} & f(X) \\ \text{subject to} & h_i(X) = 0, \qquad i \in \mathcal{E}, \\ & \qquad X \succeq 0. \end{array}$$

Here, X is an $n \times n$ symmetric matrix, h_i is affine, and $X \succeq 0$ means that X is to be positive semidefinite.

Preview of Applications

Second-Order Cone Programming

- FIR Filter Design
- Antenna Array Optimization
- Structural Optimization
- Grasping Problems

Semidefinite Programming

- SDP relaxations of combinatorial optimization problems such as MAX-CUT have been shown to be better than LP relaxations
- Eigenvalue optimization: maximize the minimum eigenvalue

More later on some of these.

Interior-Point Algorithms

- Interior-point methods were first developed in the mid 80's for LP.

- Later they were extended to NLP, SOCP, and SDP.

- Extension to NLP follows closely the LP case. That is, $\geq$ is treated the same in both cases. The nonnegative-orthant cone, $\boldsymbol{x} \geq \mathbf{0}$, plays a fundamental role.

- For SOCP, a different cone is introduced, the Lorentz cone, and algorithms are derived using this cone in place of the nonnegative orthant cone.

- Similarly for SDP, a different cone, the cone of positive semidefinite matrices, plays a fundamental role in algorithm development.

Our Aim—A Large Enclosing Superclass

- Recently, SOCP and SDP have been unified under the banner of Conic Programming and software has appeared to solve problems from the union of the SOCP and SDP problem classes.

- The aim of our work is to show that SOCP and SDP can be included under the banner of NLP and solved with “generic” NLP software. This is a much more general setting.

Formulating SOCPs as Smooth Convex NLPs

SOCP as NLP

For SOCP,

$$h_i(x) = c_i^T x + d_i - \|A_i x + b_i\|$$

is concave but not differentiable on

$$\{x : A_i x + b_i = 0\} .$$

Nondifferentiability should not be a problem unless it happens at optimality...

An Example

$$\begin{array}{ll} \text{minimize} & ax_1 + x_2 \qquad\qquad (-1 < a < 1) \\ \text{subject to} & |x_1| \leq x_2, \end{array}$$

Clearly, $(x_1^*, x_2^*) = (0, 0)$.

Dual feasibility:

$$\begin{bmatrix} a \\ 1 \end{bmatrix} + \begin{bmatrix} \frac{d|x_1|}{dx_1} \\ -1 \end{bmatrix} y = 0.$$

An interior-point method must pick the correct value for $\frac{d|x_1|}{dx_1}$ when $x_1 = 0$:

$$\left. \frac{d|x_1|}{dx_1} \right|_{x_1=0} = -a.$$

Not possible a priori.

Smooth Alternative Formulations

Constraint formulation:

$$\phi(A_i x + b_i, c_i^T x + d_i) \geq 0, \quad i = 1, \ldots, m$$

where

$$\phi(u, t) = t - \|u\|.$$

Not differentiable at $u = 0$.

Smooth alternatives:

$t - \sqrt{\epsilon^2 + \sum_i u_i^2} \geq 0,$	concave	**not equiv.**	
$t^2 - \|u\|^2 \geq 0, \; t \geq 0$	**nonconcave**	equiv.	
$t - \|u\|^2/t \geq 0, \; t > 0$	concave	equiv.	**strict interior**

Third suggestion is due to E.D. Andersen

Formulating SDPs as Smooth Convex NLPs

Formulating SDPs as NLPs

How to express

$$\begin{array}{lll} \text{minimize} & f(X) & \\ \text{subject to} & h_i(X) = 0, & i \in \mathcal{E}, \\ & X \succeq 0. & \end{array}$$

as

$$\begin{array}{lll} \text{minimize} & f(x) & \\ \text{subject to} & h_i(x) = 0, & i \in \mathcal{E}, \\ & h_i(x) \geq 0, & i \in \mathcal{I}. \end{array}$$

Characterizations of Semidefiniteness

The Definition of SDP—Semiinfinite LP

$$\xi^T X \xi \geq 0 \qquad \forall \xi \in \mathbb{R}^n.$$

Involves an infinite number of constraints.

Smallest Eigenvalue—Nonsmooth Convex NLP

$$\lambda_{\min}(X) \geq 0.$$

The smallest eigenvalue as a function of the matrix X is concave, but it is not smooth as the following example shows:

$$X = \begin{bmatrix} x_1 & y \\ y & x_2 \end{bmatrix}.$$

Eigenvalues are:

$$\begin{aligned} \lambda_1(X) &= \left((x_1 + x_2) - \sqrt{(x_1 - x_2)^2 + 4y^2} \right) /2 \\ \lambda_2(X) &= \left((x_1 + x_2) + \sqrt{(x_1 - x_2)^2 + 4y^2} \right) /2. \end{aligned}$$

Characterizations of Semidefiniteness—Continued

Factorization—Smooth Convex NLP

For $j = 1, \ldots, n$, let

$$d_j(X) = \begin{cases} D_{jj}, & \text{where } X = LDL^T, \qquad X \succeq 0, \\ -\infty, & \text{else.} \end{cases}$$

then

$$X \succeq 0 \qquad \text{if and only if} \qquad d_j(X) \geq 0 \qquad \forall j.$$

Theorem 1. *The functions d_j are*

(1) *concave,*
(2) *continuous over the positive semidefinite matrices, and*
(3) *infinitely differentiable over the positive definite matrices.*

First and Second Derivatives—Explicit Formulae

Fix j and partition X:

$$X = \begin{matrix} & & j & \\ & \left[\begin{array}{c|c|c} Z & y & * \\ \hline y^T & x & * \\ \hline * & * & * \end{array}\right. & & \end{matrix} \quad X = \begin{array}{c} \\ j \\ \\ \end{array} \left[\begin{array}{c|c|c} Z & y & * \\ \hline y^T & x & * \\ \hline * & * & * \end{array}\right].$$

Assume that $X \succeq \mathbf{0}$. Then Z is nonsingular and

$$d_j(X) = x - y^T Z^{-1} y.$$

Derivatives of d_j are easily obtained from derivatives of

$$f(y, Z) = y^T Z^{-1} y.$$

And they are...

Derivatives (Fletcher'85)

The derivatives of $f(y, Z)$ are:

$$\begin{aligned}
\frac{\partial f}{\partial y_i} &= y^T Z^{-1} e_i + e_i^T Z^{-1} y, \\
\frac{\partial^2 f}{\partial y_i \partial y_j} &= e_j^T Z^{-1} e_i + e_i^T Z^{-1} e_j, \\
\frac{\partial f}{\partial z_{ij}} &= -y^T Z^{-1} e_i e_j^T Z^{-1} y, \\
\frac{\partial^2 f}{\partial z_{ij} \partial z_{kl}} &= y^T Z^{-1} e_k e_l^T Z^{-1} e_i e_j^T Z^{-1} y + y^T Z^{-1} e_i e_j^T Z^{-1} e_k e_l^T Z^{-1} y, \\
\frac{\partial^2 f}{\partial y_i \partial z_{kl}} &= -y^T Z^{-1} e_k e_l^T Z^{-1} e_i - e_i^T Z^{-1} e_k e_l^T Z^{-1} y.
\end{aligned}$$

From these it is easy to check that f is convex and hence that d_j is concave.

LOQO—A Specific Interior-Point Algorithm for NLP

The problem:

$$\begin{array}{lll} \text{minimize} & f(x) & \\ \text{subject to} & h_i(x) = 0, & i \in \mathcal{E}, \\ & h_i(x) \geq 0, & i \in \mathcal{I}. \end{array}$$

is represented internally by LOQO as:

$$\begin{array}{lrl} \text{minimize} & f(x) & \\ \text{subject to} & h_i(x) - w_i = 0, & i \in \mathcal{E}, \\ & h_i(x) - w_i = 0, & i \in \mathcal{I}, \\ & x_j - g_j + t_j = 0, & j = 1, \ldots, n, \\ & w_i + p_i = 0, & i \in \mathcal{E}, \\ & w_i \geq 0, & i \in \mathcal{E} \cup \mathcal{I}, \\ & p_i \geq 0, & i \in \mathcal{E}, \\ & g_j \geq 0, & j = 1, \ldots, n, \\ & t_j \geq 0, & j = 1, \ldots, n. \end{array}$$

Complementarity

Each nonnegative variable has a complementary dual variable:

$$\begin{array}{ccc} w_i & \longleftrightarrow & y_i, \\ p_i & \longleftrightarrow & q_i, \\ g_i & \longleftrightarrow & z_i, \\ t_i & \longleftrightarrow & s_i. \end{array}$$

Using LOQO to solve SDPs

LOQO preserves positivity of each nonnegative variable.

It does not preserve positivity of h_i.

In fact, without further precaution, h_i does indeed go negative.

This is bad for SDP!

Step Shortening

Steps lengths can be shortened to guarantee that $\boldsymbol{h_i}$'s remain positive.

But then the algorithm could "jam".

Next theorem shows that simple jamming does not occur.

To be precise, we consider a point $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{w}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{p}}, \bar{\boldsymbol{q}}, \bar{\boldsymbol{g}}, \bar{\boldsymbol{z}}, \bar{\boldsymbol{t}}, \bar{\boldsymbol{s}})$ satisfying the following conditions:

(1) Nonnegativity: $\bar{\boldsymbol{w}} \geq \mathbf{0}$, $\bar{\boldsymbol{y}} \geq \mathbf{0}$, $\bar{\boldsymbol{p}} \geq \mathbf{0}$, $\bar{\boldsymbol{q}} \geq \mathbf{0}$, $\bar{\boldsymbol{g}} \geq \mathbf{0}$, $\bar{\boldsymbol{z}} \geq \mathbf{0}$, $\bar{\boldsymbol{t}} \geq \mathbf{0}$, and $\bar{\boldsymbol{s}} \geq \mathbf{0}$.

(2) Complementary variable positivity:

$$\bar{\boldsymbol{w}} + \bar{\boldsymbol{y}} > \mathbf{0}, \quad \bar{\boldsymbol{p}} + \bar{\boldsymbol{q}} > \mathbf{0}, \quad \bar{\boldsymbol{g}} + \bar{\boldsymbol{z}} > \mathbf{0}, \quad \text{and} \quad \bar{\boldsymbol{t}} + \bar{\boldsymbol{s}} > \mathbf{0}.$$

(3) Free variable positivity: $\bar{\boldsymbol{g}} + \bar{\boldsymbol{t}} > \mathbf{0}$.

Nonjamming Theorem

We are interested in a point where some of the $\bar{w}_i$'s vanish. Let

$$\mathcal{U} = \{i \;:\; \bar{w}_i = 0\}$$

and let $\mathcal{B}$ denote the other indices. Write $A = (\nabla h^T)^T$ (and other matrices and vectors) in block form according to this partition

$$A = \begin{bmatrix} B \\ U \end{bmatrix}$$

Matrix A and other quantities are functions of the current point. We use the same letter with a bar over it to denote the value of these objects at the point $(\bar{x}, \ldots, \bar{s})$.

Theorem 2. *If the point* $(\bar{x}, \ldots, \bar{s})$ *satisfies conditions* (1)–(3), *and* $\bar{U}$ *has full row rank, then the step direction* Δw *for* w *has a continuous extension to this point and* $\Delta w_{\mathcal{U}} = \mu Y_{\mathcal{U}}^{-1} e_{\mathcal{U}} > 0$ *there.*

An Example

We used LOQO to solve the following simple problem:

$$\begin{array}{ll} \text{minimize} & \sum_{j=1}^{3} X_{jj} \\ \text{subject to} & X_{12} = 1 \\ & X_{13} = 1.5 \\ & X_{23} = 2 \\ & \quad X \succeq 0. \end{array}$$

Without step shortening, LOQO fails.

With step shortening, it solves in less than 20 iterations.

Applications and Computational Results

Finite Impulse Response (FIR) Filter Design–Low Pass Filter

$$\text{minimize} \quad \rho$$

$$\text{subject to} \quad \left(\frac{1}{|L|}\int_L (H(\nu)-1)^2\,d\nu\right)^{1/2} \le \rho$$

$$\left(\frac{1}{|H|}\int_H H^2(\nu)d\nu\right)^{1/2} \le \rho$$

where

$$H(\nu) = \sum_{k=-5}^{19} h(k)e^{2\pi i k\nu},$$

$h(k)$ = Complex filter coefficients, i.e., **decision variables**

Specific Example

constraints	1880
variables	1648
time (iterations)	
LOQO	17.7 (33)
SEDUMI(Sturm)	46.1 (18)

Ref: J.O. Coleman and D.P. Scholnik,

U.S. Naval Research Laboratory,

MWSCAS99 paper available: engr.umbc.edu/∼jeffc/pubs/abstracts/mwscas99socp.html

Click here for an animation.

FIR Filter Design–Woofer, Midrange, Tweeter

minimize ρ

subject to $\displaystyle\int_0^1 (H_w(\nu) + H_m(\nu) + H_t(\nu) - 1)^2 \, d\nu \le \epsilon$

$$\left(\frac{1}{|W|}\int_W H_w^2(\nu)d\nu\right)^{1/2} \le \rho \qquad W = [.2, .8]$$

$$\left(\frac{1}{|M|}\int_M H_m^2(\nu)d\nu\right)^{1/2} \le \rho \qquad M = [.4, .6] \cup [.9$$

$$\left(\frac{1}{|T|}\int_T H_t^2(\nu)d\nu\right)^{1/2} \le \rho \qquad T = [.7, .3]$$

where

$$H_i(\nu) = h_i(0) + 2\sum_{k=1}^{n-1} h_i(k)\cos(2\pi k\nu), \quad i = w, m, t$$

$h_i(k)$ = filter coefficients, i.e., **decision variables**

Specific Example: Pink Floyd's "Money"

$$\text{filter length: } n = 14$$
$$\text{integral discretization: } N = 1000$$

constraints	4
variables	43
time (secs)	
LOQO	79
MINOS	164
LANCELOT	3401
SNOPT	35

Ref: J.O. Coleman,

U.S. Naval Research Laboratory,

CISS98 paper available: engr.umbc.edu/∼jeffc/pubs/abstracts/ciss98.html

Click here for a demo

Wide-Band Antenna Array Weight Design–SOCP

$$
\begin{array}{ll}
\text{minimize} & \alpha \\
\text{subject to} & \iint_{(\mu,\nu)\in S} |A(\mu,\nu)|^2 d\mu d\nu \leq \alpha, \\
& |A(\mu,\nu)| \leq 10^{-25/20}, \quad (\mu,\nu) \in S \\
& |A(\mu,\nu)| \leq 10^{-45/20}, \quad (\mu,\nu) \in S_0 \\
& \int_{\nu\in P} |A(\mu_m,\nu) - \beta_m|^2 d\nu \leq 10^{-50/10}, \quad m = 1,\ldots,M \\
& \sum_m \beta_m = M \\
& \sum_k \sum_n |c_{kn}|^2 \leq \rho
\end{array}
$$

where

$$
A(\mu,\nu) = \sum_k \sum_n c_{kn} e^{-2\pi i(k\mu+n\nu)}
$$

c_{kn} = complex-valued **design weight** for array element k at freq tap n

S = subset of direction/freq pairs representing sidelobe

S_0 = subset of sidelobe spelling NRL

μ_m = finite set of directions “covering” pass band

ρ = Noise bound

Specific Example

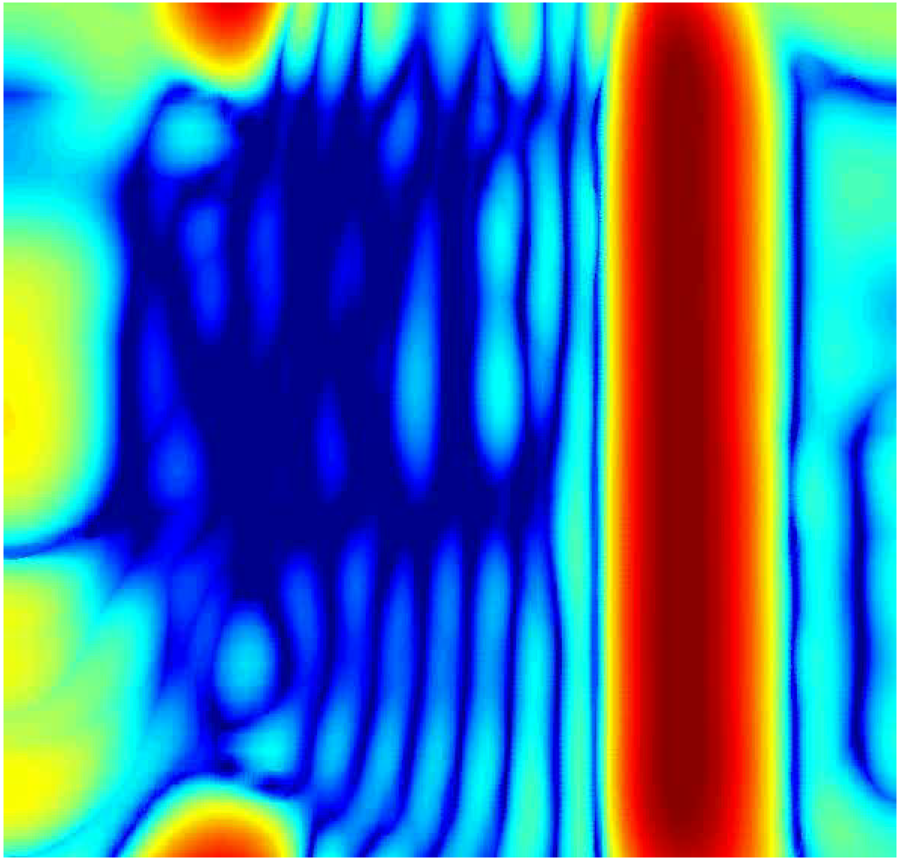
15 antennae in a linear array
21 “taps” on each array
671 Chebychev constraints to spell “NRL”

constraints	6230
variables	624
time (iterations)	
LOQO	1049 (48)
SEDUMI(Sturm)	573 (27)

Ref: D.P. Scholnik and J.O. Coleman,

U.S. Naval Research Laboratory,

RADAR2000 paper available: engr.umbc.edu/∼jeffc/pubs/papers/radar2Kds/radar2000.html



Max-Cut Problem–SDP

$$\begin{array}{ll} \text{minimize} & C \cdot X - e^T C \text{Diag}(X) \\ \text{subject to} & X_{jj} = 1/4, \qquad j = 1, \ldots, n, \\ & X \succeq 0. \end{array}$$

where C is a symmetric incidence matrix; i.e., $C_{ij} = 1$ if $\{i, j\}$ is an arc and zero otherwise.

Specific example: C is a 15×15 arc incidence matrix for a randomly generated graph with 1.5 arcs/node on average.

LOQO solves this problem in 27 iterations and 43.0 seconds.

Max-Min Eigenvalue–SDP

$$\begin{array}{lll} \text{minimize} & -C \cdot X & \\ \text{subject to} & X_{kk} = 1, & k = 1, \ldots, n, \\ & X \succeq 0. & \end{array}$$

Specific example: C is a 10×10 random symmetric matrix.

LOQO solves the problem in 27 iterations and 43.1 seconds.

Exploiting Sparsity

The computational results shown for SDP don't properly exploit sparsity.
For SDP, one way to exploit sparsity is to solve the dual problem:

$$\begin{array}{ll} \text{maximize} & \sum_k b_k y_k \\ \text{subject to} & \sum_k A_{ij}^{(k)} y_k - Z_{ij} = C_{ij}, \qquad i,j = 1,2,\ldots,n \\ & Z \succeq 0. \end{array}$$

Typically, C and each of the $A^{(k)}$ are sparse matrices.

The equality constraints then imply that Z must have a sparsity pattern that is the union of the sparsity patterns of these other matrices.

Therefore, Z is often sparse too, and its sparsity pattern is known from the beginning.
This sparsity is easy to exploit, although we haven't done that yet. Also, derivative calculatioins require Z^{-1} too.