
Conic Optimization: Interior-Point Methods and Beyond

Michael J. Todd

School of Operations Research and Information Engineering,
Cornell University

www.orie.cornell.edu/~miketodd/todd.html

Optimization 2007, Porto

July 24, 2007

Outline

- Conic programming problems
- Duality
- Applications
- Interior-point algorithms
- Old/new algorithms for minimum-volume enclosing ellipsoid problem

I. Conic programming problems

- Linear programming (LP)
- Semidefinite programming (SDP)
- Second-order cone programming (SOCP)
- General conic programming problem

Linear programming:

Given $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, consider:

$$\begin{aligned} & \min_x \quad c^T x \\ \text{(P)} \quad & Ax = b, \\ & x \geq 0. \end{aligned}$$

Using the same data, we can construct the dual problem:

$$\begin{aligned} & \max_{y,s} \quad b^T y \\ \text{(D)} \quad & A^T y + s = c, \\ & s \geq 0. \end{aligned}$$

Semidefinite programming:

Given $A_i \in \mathcal{SR}^{p \times p}$ (symmetric real matrices of order p), $i = 1, \dots, m$, $b \in \mathbb{R}^m$,
 $C \in \mathcal{SR}^{p \times p}$, consider:

$$\begin{aligned} & \min_X \quad C \bullet X \\ \text{(P)} \quad & \mathcal{AX} := (A_i \bullet X)_{i=1}^m = b, \\ & X \succeq 0, \end{aligned}$$

where $S \bullet Z := \text{Trace } S^T Z = \sum_i \sum_j s_{ij} z_{ij}$ for matrices of the same dimensions,
and $X \succeq 0$ means X is symmetric and positive semidefinite (psd). (We'll also write
 $A \succeq B$ and $B \preceq A$ for $A - B \succeq 0$.) We'll write $\mathcal{SR}_+^{p \times p}$ for the cone of psd real
matrices of order p .

SDP, cont'd

Note that, instead of the **components** of the **vector** x being nonnegative, now the p **eigenvalues** of the **symmetric matrix** X are nonnegative.

Using the same data, we can construct another SDP in dual form:

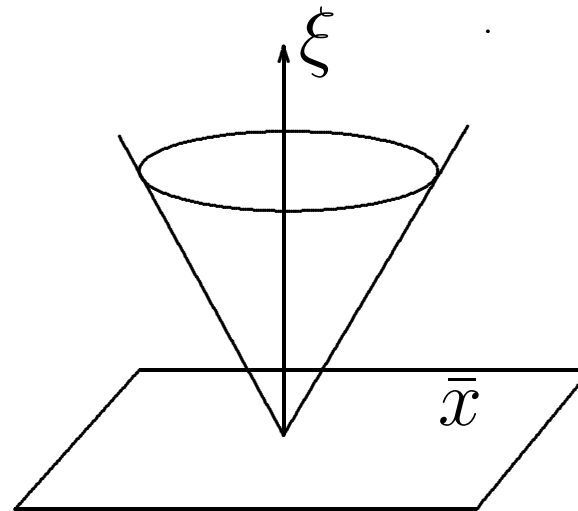
$$\begin{aligned} & \max_{y,S} && b^T y \\ \text{(D)} \quad & \mathcal{A}^* y + S := && \sum_i y_i A_i + S = C, \\ & && S \succeq 0. \end{aligned}$$

Second-order cone programming (SOCP)

Given $A_j \in \mathbb{R}^{m \times (1+n_j)}$, $c_j \in \mathbb{R}^{1+n_j}$, $j = 1, \dots, k$, and $b \in \mathbb{R}^m$, consider:

$$\begin{aligned} \text{(P)} \quad & \min_{x_1, \dots, x_k} \quad c_1^T x_1 + \dots + c_k^T x_k \\ & A_1 x_1 + \dots + A_k x_k = b, \\ & x_j \in \mathcal{S}_2^{1+n_j}, \quad j = 1, \dots, k, \end{aligned}$$

where \mathcal{S}_2^{1+q} is the **second-order cone**:



$$\{x := (\xi; \bar{x}) \in \mathbb{R}^{1+q} : \xi \geq \|\bar{x}\|_2\}$$

SOCP, cont'd

Again using the same data, we can construct a problem in dual form:

$$\begin{aligned} & \max_{y, s_1, \dots, s_k} && b^T y \\ \text{(D)} & && A_1^T y + s_1 = c_1 \\ & && \vdots \\ & && A_k^T y + s_k = c_k \\ & && s_j \in \mathcal{S}^{1+n_j}, j = 1, \dots, k. \end{aligned}$$

General conic programming problem:

Given again $A \in \mathfrak{R}^{m \times n}$, $b \in \mathfrak{R}^m$, $c \in \mathfrak{R}^n$, and a **closed convex cone** $K \subset \mathfrak{R}^n$, consider

$$\begin{aligned} & \min_x \quad \langle c, x \rangle \\ \text{(P)} \quad & Ax = b, \\ & x \in K, \end{aligned}$$

where we have written $\langle c, x \rangle$ instead of $c^T x$ to emphasize that this can be thought of as a general scalar/inner product. E.g., if our original problem is an SDP involving $X \in \mathcal{SR}^{p \times p}$, we need to embed it into \mathfrak{R}^n for some n .

Even though our problem (P) looks very much like LP, **every convex programming problem can be written in the form (P)**.

Conic problem in dual form

How do we construct the corresponding problem in dual form? We need the **dual cone**:

$$K^* = \{s \in \mathbb{R}^n : \langle s, x \rangle \geq 0 \text{ for all } x \in K\}.$$

Then we define

$$\begin{aligned} \max_{y,s} \quad & \langle b, y \rangle \\ \text{(D)} \quad & A^*y + s = c, \\ & s \in K^*. \end{aligned}$$

What is A^* ? The operator **adjoint** to A , so that for all x, y , $\langle A^*y, x \rangle = \langle Ax, y \rangle$.

If $\langle \cdot, \cdot \rangle$ is the usual dot product, $A^* = A^T$.

II. Duality

We start with the well-known one-line proof of weak duality for **LP**:

$$c^T x - b^T y = (A^T y + s)^T x - (Ax)^T y = s^T x \geq 0.$$

For **SDP**:

$$\begin{aligned} C \bullet X - b^T y &= \left(\sum y_i A_i + S \right) \bullet X - \left((A_i \bullet X)_{i=1}^m \right)^T y \\ &= S \bullet X \geq 0. \end{aligned}$$

Needs: $U \bullet V \geq 0$ for $U, V \succeq 0$.

And for **SOCP**:

$$\begin{aligned} \sum c_j^T x_j - b^T y &= \sum (A_j^T y + s_j)^T x_j - \left(\sum A_j x_j \right)^T y \\ &= \sum s_j^T x_j \geq 0. \end{aligned}$$

Needs: $u^T v \geq 0$ for $u, v \in \mathcal{S}_2^{1+q}$.

Weak duality for general conic problems

These are all special cases of weak duality for general conic programming:

If x is feasible for (P) and (y, s) for (D), then

$$\langle c, x \rangle - \langle b, y \rangle = \langle A^*y + s, x \rangle - \langle Ax, y \rangle \stackrel{(i)}{=} \langle s, x \rangle \stackrel{(ii)}{\geq} 0,$$

where (i) follows **by definition** of the adjoint operator A^* and (ii) **by definition** of the dual cone K^* .

(We need to show that $\mathcal{SR}_+^{p \times p}$ and \mathcal{S}_2^{1+q} are self-dual.)

So in all cases we have weak duality, which suggests that it is worthwhile to consider (P) and (D) together. In many cases, strong duality holds, and then it is very worthwhile!

Strong duality

Strong duality, by which we mean that both (P) and (D) have optimal solutions and there is no duality gap, **doesn't hold in general** in conic programming. We need to add a regularity condition.

We say x is a **strictly feasible** solution for (P) if it is feasible and $x \in \text{int } K$; similarly (y, s) is a strictly feasible solution for (D) if it is feasible and $s \in \text{int } K^*$.

Theorem If **both** (P) and (D) have **strictly feasible** solutions, **strong duality holds**.

Notation: $\mathcal{F}(\text{P}) := \{\text{feasible solutions of (P)}\}$ and similarly for (D).

$\mathcal{F}^0(\text{P}) := \{\text{strictly feasible solutions of (P)}\}$ and similarly for (D).

III. Applications

- matrix optimization
- quadratically constrained quadratic programming (QCQP)
- control theory
- relaxations in combinatorial optimization
- global optimization of polynomials

Matrix optimization

Suppose we have a symmetric matrix

$$A(y) := A_0 + \sum_{i=1}^m y_i A_i$$

depending affinely on $y \in \Re^m$. We wish to choose y to **minimize the maximum eigenvalue of** $A(y)$.

Note: $\lambda_{\max}(A(y)) \leq \eta$ iff all e-values of $\eta I - A(y)$ are nonnegative iff $A(y) \preceq \eta I$.

This gives

$$\begin{aligned} \max_{\eta, y} \quad & -\eta \\ & -\eta I + \sum_{i=1}^m y_i A_i \preceq -A_0, \end{aligned}$$

an SDP problem of form (D).

QCQP

Proposition (Schur complements) Suppose $B \succ 0$. Then

$$\begin{pmatrix} B & P \\ P^T & C \end{pmatrix} \succeq 0 \Leftrightarrow C - P^T B^{-1} P \succeq 0.$$

Hence the convex quadratic constraint $(Ay + b)^T (Ay + b) - c^T y - d \leq 0$ holds iff

$$\begin{pmatrix} I & Ay + b \\ (Ay + b)^T & c^T y + d \end{pmatrix} \succeq 0,$$

or alternatively iff $\sigma \geq \|\bar{s}\|_2$, $\sigma := c^T y + d + \frac{1}{4}$, $\bar{s} := (c^T y + d - \frac{1}{4}; Ay + b)$.

This allows us to model the QCQP of minimizing a convex quadratic function subject to convex quadratic inequalities as **either** an SDP **or** an SOCP.

Control theory

Suppose the state of a system is defined by $\dot{x} \in \text{conv}\{P_1, P_2, \dots, P_m\} x$.

A **sufficient** condition that $x(t)$ is **bounded for all time** is that there is $Y \succ 0$ with

$V(x) := \frac{1}{2}x^T Y x$ nonincreasing, i.e.,

$$\dot{V}(x) = \frac{1}{2}x^T (YP + P^T Y)x \leq 0$$

for all $P \in \text{conv}\{P_1, P_2, \dots, P_m\}$. This leads to

$$\max_{\eta, Y} \quad -\eta$$

$$-\eta I \quad + \quad Y \quad \preceq \quad 0,$$

$$-Y \quad \preceq \quad -I,$$

$$YP_i \quad + \quad P_i^T Y \quad \preceq \quad 0, \quad i = 1, \dots, m.$$

(Note the **block diagonal structure**.)

Relaxations in combinatorial optim'n

The **Maximum Cut Problem**: given an undirected (wlog complete) graph on $V = \{1, \dots, n\}$ with nonnegative edge weights $W = (w_{ij})$, find a cut $\delta(S) := \{\{i, j\} : i \in S, j \notin S\}$ with maximum weight.

(IP): $\max\{\frac{1}{4} \sum_i \sum_j w_{ij}(1 - x_i x_j) : x_i \in \{-1, +1\}, i = 1, \dots, n\}$.

The constraint is the same as $x_i^2 = 1$ all i . Now

$\{X : x_{ii} = 1, i = 1, \dots, n, X \succeq 0, \text{rank}(X) = 1\} = \{xx^T : x_i^2 = 1, i = 1, \dots, n\}$.

So a **relaxation** is:

$$\begin{aligned} \frac{1}{4} \sum \sum w_{ij} - \frac{1}{4} \min_X W \bullet X \\ e_i e_i^T \bullet X &= 1, i = 1, \dots, n, \\ X &\succeq 0. \end{aligned}$$

This gives a good bound and a good feasible solution (within 14%)

(Goemans and Williamson).

Global optimization of polynomials

Lastly, we just indicate the approach to global optimization of polynomials using conic programming.

Given a polynomial function θ of q variables, the **globally optimal** value of minimizing $\theta(x)$ over all $x \in \mathbb{R}^q$ is the maximum value of η such that the polynomial $p(x) \equiv \theta(x) - \eta$ is nonnegative for all x , and this is a **convex set** of polynomials (described say by all their coefficients).

This equivalence indicates that the convex cone of nonnegative polynomials **must be hard** to deal with. It can be **approximated** using SDPs; clearly if p is the **sum of squares** of polynomials then it is nonnegative (but not conversely); however, using extensions of these ideas we can approximate the optimal value as closely as desired.

V. Algorithms

We will concentrate on **interior-point** methods (IPMs), which have the theoretical advantage of **polynomial-time** complexity, while also performing very well in practice on medium-scale problems.

Assume K is solid and pointed, and $\mathcal{F}^0(P)$ and $\mathcal{F}^0(P)$ nonempty.

$F : \text{int } K \rightarrow \Re$ is a **barrier function** for K if

- F is strictly convex; and
- $x_k \rightarrow \bar{x} \in \partial K \Rightarrow F(x_k) \rightarrow +\infty$.

Similarly, let F_* be a barrier function for $\text{int } K^*$.

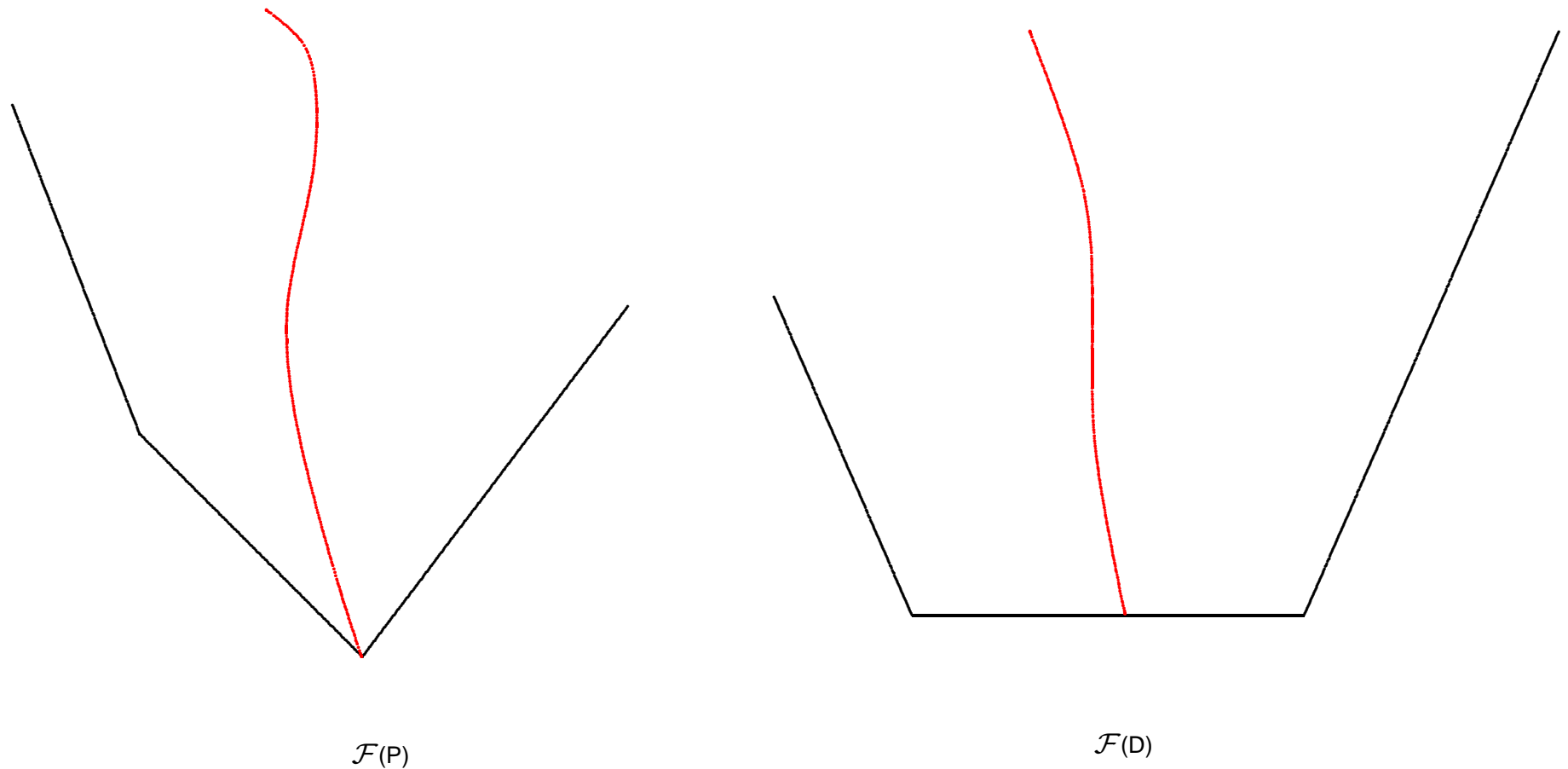
Barrier Problems: Choose $\mu > 0$ and consider

$$(\text{BP}_\mu) \quad \min \langle c, x \rangle + \mu F(x), \quad Ax = b \quad (x \in \text{int } K),$$

$$(\text{BD}_\mu) \quad \max \langle b, y \rangle - \mu F_*(s), \quad A^*y + s = c \quad (s \in \text{int } K^*).$$

Central paths

These have unique solutions $x(\mu)$ and $(y(\mu), s(\mu))$ varying smoothly with μ , forming trajectories in the feasible regions, the so-called **central paths**:



Self-concordant barriers

F is a ν -self-concordant barrier for K (Nesterov and Nemirovski) if

- F is a C^3 barrier for K ;
- For all $x \in \text{int } K$, $D^2 F(x)$ is pd; and
- For all $x \in \text{int } K$, $d \in \Re^n$,
 - (i) $|D^3 F(x)[d, d, d]| \leq 2(D^2 F(x)[d, d])^{3/2}$;
 - (ii) $|DF(x)[d]| \leq \sqrt{\nu}(D^2 F(x)[d, d])^{1/2}$.

F is ν -logarithmically homogeneous if

- For all $x \in \text{int } K$, $\tau > 0$, $F(\tau x) = F(x) - \nu \ln \tau$ (\Rightarrow (ii)).

Examples: for $K = \Re_+^n$: $F(x) := -\ln(x) := -\sum \ln(x^{(j)})$ with $\nu = n$;

for $K = \mathcal{SR}_+^{p \times p}$: $F(X) := -\ln \det X = -\sum \ln(\lambda_j(X))$ with $\nu = p$;

for $K = \mathcal{S}_2^{1+q}$: $F(\xi; \bar{x}) := -\ln(\xi^2 - \|\bar{x}\|_2^2)$ with $\nu = 2$.

Properties

Henceforth, F is a ν -LHSCB for K .

Define the **dual barrier**: $F_*(s) := \sup\{-\langle s, x \rangle - F(x)\}$.

Then F_* is a ν -LHSCB for K^* .

$$F(x) = -\ln(x) \Rightarrow F_*(s) = -\ln(s) - n;$$

$$F(X) = -\ln \det X \Rightarrow F_*(S) = -\ln \det S - p.$$

Properties: For all $x \in \text{int } K$, $\tau > 0$, $s \in \text{int } K^*$,

• $F'(\tau x) = \tau^{-1} F'(x)$, $F''(\tau x) = \tau^{-2} F''(x)$, $F''(x)x = -F'(x)$.

• $x \in \text{int } K \Rightarrow -F'(x) \in \text{int } K^*$.

• $\langle -F'(x), x \rangle = \langle s, -F'_*(s) \rangle = \nu$.

• $s = -F'(x) \Leftrightarrow x = -F'_*(s)$.

• $F''_*(-F'(x)) = [F''(x)]^{-1}$.

• $\nu \ln \langle s, x \rangle + F(x) + F_*(s) \geq \nu \ln \nu - \nu$, with equality iff $s = -\mu F'(x)$

(or $x = -\mu F'_*(s)$) for some $\mu > 0$.

Central path equations

Optimality conditions for barrier problems:

x is **optimal** for (BP_μ) iff $\exists(y, s)$ with

$$\begin{aligned}A^*y + s &= c, & s \in \text{int } K^*, \\Ax &= b, & x \in \text{int } K, \\ \mu F'(x) + s &= 0.\end{aligned}$$

Similarly, (y, s) is **optimal** for (BD_μ) iff $\exists x$ with the **same** first two equations and $x + \mu F'_*(s) = 0$.

These two sets of equations are **equivalent** if F and F_* are as above!

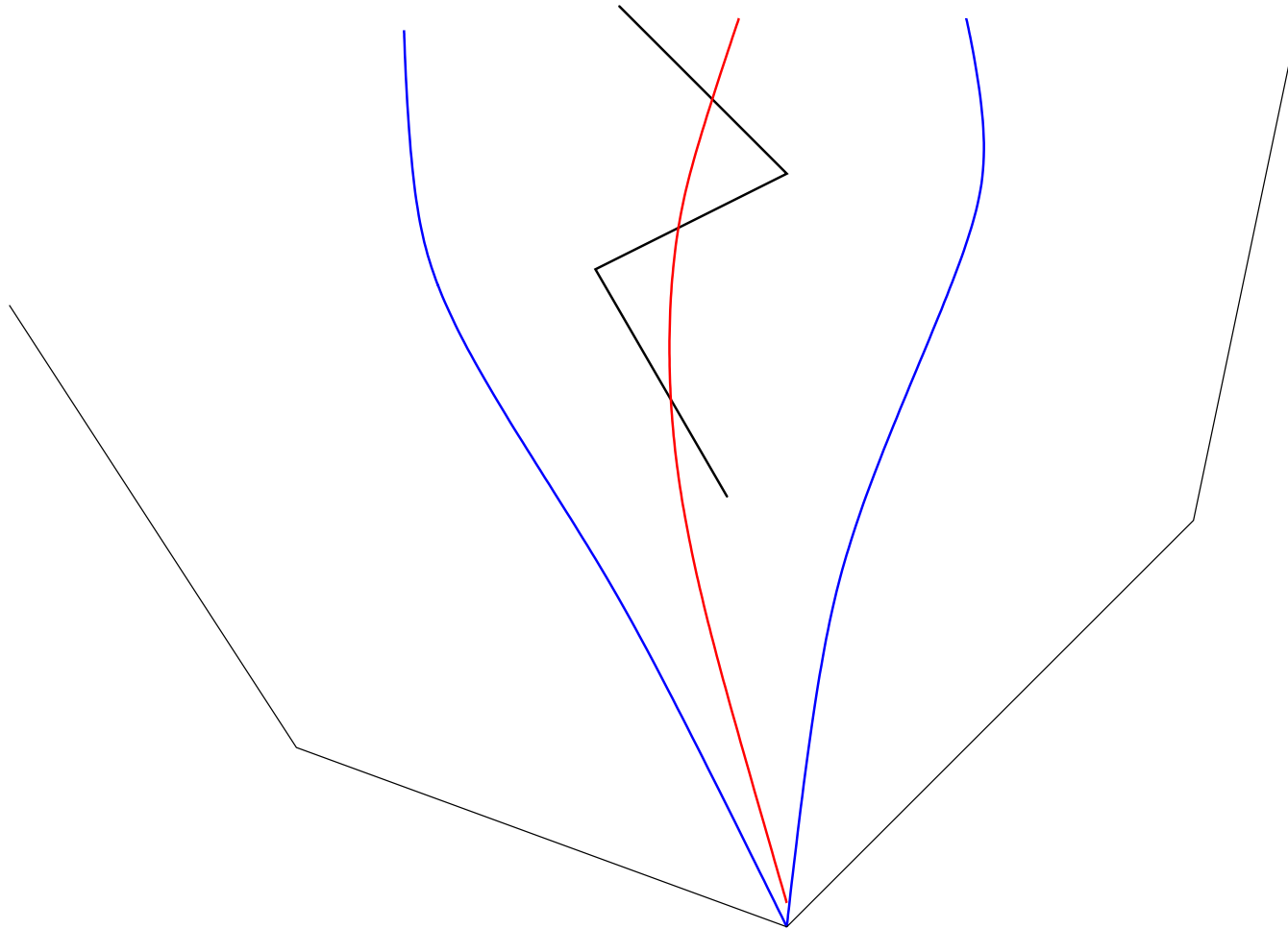
Also, if we have $x(\mu)$ solving (BP_μ) , we can easily get $(y(\mu), s(\mu))$ with **duality gap**

$$\langle s(\mu), x(\mu) \rangle = \mu \langle -F'(x(\mu), x(\mu)) \rangle = \nu\mu,$$

which **tends to zero** as $\mu \downarrow 0$ (this provides an alternative proof of **strong duality**).

Path-following algorithms

This leads to theoretically efficient path-following algorithms which use Newton's method to approximately follow the paths:



Complexity

Given a strictly feasible (x_0, y_0, s_0) close to the central path, we can produce a strictly feasible (x_k, y_k, s_k) close to the central path with

$$\langle c, x_k \rangle - \langle b, y_k \rangle = \langle s_k, x_k \rangle \leq \epsilon \langle s_0, x_0 \rangle$$

within

$$O(\nu \ln(1/\epsilon)) \quad \text{or} \quad O(\sqrt{\nu} \ln(1/\epsilon))$$

iterations. This is a primal **or** dual algorithm, unlike the primal-dual algorithms typically used for LP.

Major work per iteration: forming and factoring the sparse or dense Schur complement matrix $A[F''(x)]^{-1}A^T$ or $AF''(s)A^T$.

For LP, $A \text{Diag}(x)^2 A^T$ or $A \text{Diag}(s)^{-2} A^T$;

for SDP, $(A_i \bullet (XA_jX))$ or $(A_i \bullet (S^{-1}A_jS^{-1}))$.

Can we devise symmetric primal-dual algorithms?

Self-scaled cones

Yes, for certain cones K and barriers F . We need to find, for every $x \in \text{int } K$ and $s \in \text{int } K^*$, a **scaling point** $w \in \text{int } K$ with

$$F''(w)x = s.$$

Then $F''(w)$ approximates $\mu F''(x)$ and simultaneously $F''_*(t) := F''_*(-F'(w)) = [F''(w)]^{-1}$ approximates $\mu F''_*(s)$. Hence we find our search direction $(\Delta x, \Delta y, \Delta s)$ from

$$\begin{aligned} A^* \Delta y + \Delta s &= r_d, \\ A \Delta x &= r_p, \\ F''(w) \Delta x + \Delta s &= r_c. \end{aligned}$$

This generalizes standard primal-dual methods for LP.

Self-scaled cones, cont'd

For what cones can we find such barriers? So-called **self-scaled cones** (Nesterov-Todd), also the same as **symmetric** (homogeneous and self-dual) cones (Güler), which have been **completely characterized**. Includes LP, SDP, SOCP (and not much else).

There is another approach to defining central paths and hence algorithms, with **no barrier functions**. The idea is to generalize the characterization of LP optimality using complementary slackness, and the definition of the central path using **perturbed complementary slackness** conditions $x_j s_j = \mu$ for each j . The corresponding general structure is a **Euclidean Jordan algebra** and its **cone of squares**. These give precisely the same class of cones as above! (Faybusovich and Güler.)

The corresponding perturbed complementary slackness conditions for SDP are

$$\frac{1}{2}(XS + SX) = \mu I.$$

Performance of SDPT3-4.0-beta

Problem	m	n_l	n_q	n_s	It's	Time
arch8	174	174	–	161	23*	5.5
control11	1596	–	–	110;55	26*	214
copo23	5820	1771	–	529	19	465
equalG32	2001	–	–	2001	17	1004
filter48-socp	948	931	49	48	43*	72
maxG32	2000	–	–	2000	15	257
nb-L1	915	797	793×3	–	23	14
nql180	130,080	129,602	$32,400 \times 3$	–	39	246
pds-10	32,706	66,080	–	–	58*	158
sched-2-1-s	18,086	20,002	17,885	–	37*	55
theta6	4375	–	–	300	14	136

There are a variety of other methods for conic programming problems, which typically **sacrifice the polynomial-time complexity** of interior-point methods to get **improved efficiency** for certain large-scale problems (Helmberg-Rendl, Burer-Monteiro-Zhang).

The MVEE problem

Given points $x_1, \dots, x_m \in \mathfrak{R}^n$, we want to find the minimum-volume central ellipsoid containing them. Its dual turns out to be the D-optimal design problem from statistics. Let $X := [x_1, \dots, x_m] \in \mathfrak{R}^{n \times m}$.

$$(P) \quad \begin{aligned} \min_{H \succ 0} \quad & -\ln \det H \\ & x_i^T H x_i \leq n \quad \text{for all } i. \end{aligned}$$

$$(D) \quad \begin{aligned} \max \quad & \ln \det X \text{Diag}(u) X^T \\ & e^T u = 1, \quad u \geq 0. \end{aligned}$$

Not quite in conic form, but can be handled by SDPT3-4.0-beta, e.g.

FW-Kh, FW-KY, and WA-TY algorithms

“Barycentric coordinate descent” (Fedorov, Wynn, Frank-Wolfe) applied to (D):

Start with $u = (1/m)e$ (uniform distribution).

At each iteration, compute $\max_i x_i^T (X \text{Diag}(u) X^T)^{-1} x_i$ and stop if the max is at most $(1 + \epsilon)n$.

Else update $u \leftarrow (1 - \delta)u + \delta e_i$ for the optimal $\delta > 0$.

Analyzed by Khachiyan, and with an improved initialization, by Kumar and Yıldırım.

Modification with “away steps” (Wolfe, Atwood) analyzed by Todd and Yıldırım.

Linear convergence proved by Ahipasaoglu, Sun, and Todd.

Example: 5,000 points in dimension 500:

WA-TY takes 5453 iterations and 125 seconds;

SDPT3 takes 16 iterations and 650 seconds (and much more storage).

Summary

A **wide** range of problems from a **broad** range of applications can be modelled as conic optimization problems. There is a **beautiful theory** for such problems, which leads to **efficient algorithms** for **medium-scale** problems with **suitable cones**.

Work continues on **extensions to other cones**, and on **first-order methods** for **large-scale** problems.