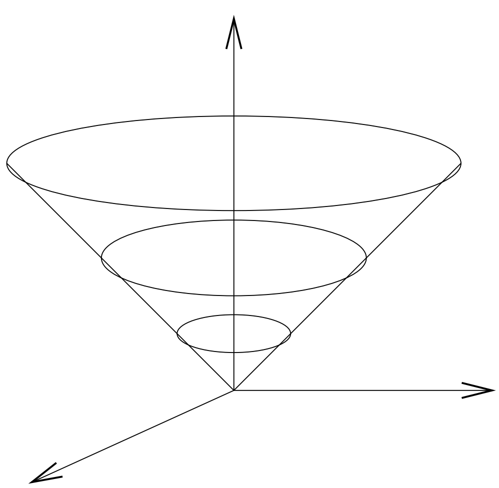
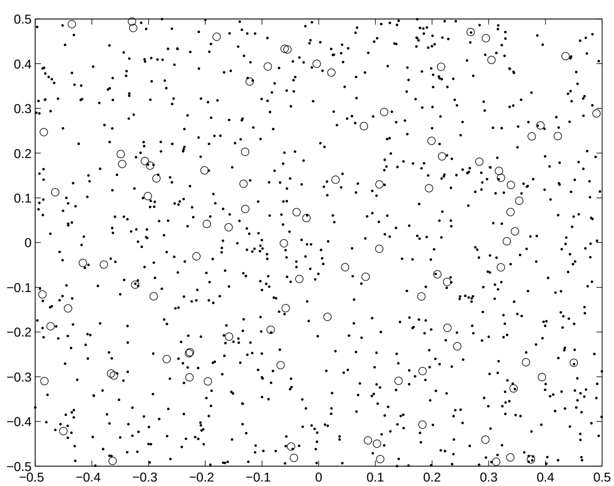
SOCP Relaxation in Euclid. Geometry Optimization

Paul Tseng

Univ. Washington, Seattle

May 2004, Waterloo Workshop





Talk Outline:

- Problem Description
- SDP and SOCP relaxation
- Properties of SOCP relaxation
- Solving SOCP relaxation
- Conclusions

Problem:

- n pts in $\Re^d$ $(d = 2, 3)$.
- Know first m pts ('anchor')

$$a_1, ..., a_m$$

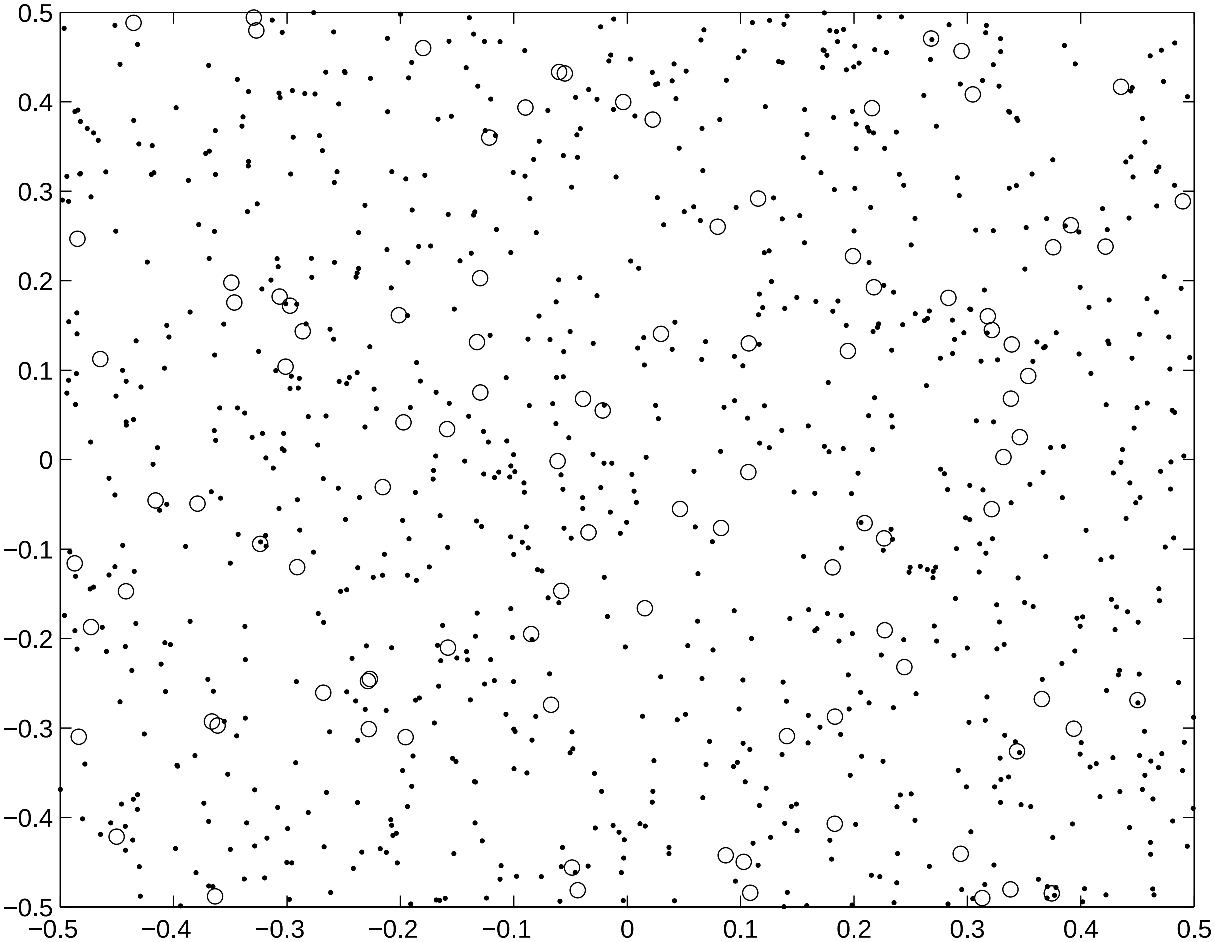
and Eucl. dist. estimate for pairs of 'neighboring' pts

$$d_{ij} \geq 0 \quad \forall (i,j) \in \mathcal{A}$$

with $\mathcal{A} \subseteq \{(i,j) : 1 \leq i < j \leq n\}$.

- Estimate remaining $n - m$ pts.

(More generally, upper/lower bounds on dist.)



History? Graph realization, position estimation in wireless sensor network,...

Optimization Formulation:

$$\boxed{\begin{aligned} v_{\mathrm{opt}} \stackrel{\mathrm{def}}{=} \min_{x_{m+1},\ldots,x_n} & \sum_{\substack{i \le m < j \\ (i,j) \in \mathcal{A}}} \left| \|a_i - x_j\|^2 - d_{ij}^2 \right| \\ & + \sum_{\substack{m < i < j \\ (i,j) \in \mathcal{A}}} \left| \|x_i - x_j\|^2 - d_{ij}^2 \right| \end{aligned}}$$

Objective function is nonconvex.

Use a convex program relaxation.

SDP Relaxation (Biswas,Ye '03):

$$A \stackrel{\text{def}}{=} [a_1 \ \cdots \ a_m], \qquad X \stackrel{\text{def}}{=} [x_{m+1} \ \cdots \ x_n].$$

Then

$$\begin{aligned}
\|a_i - x_j\|^2 &= \|a_i\|^2 - 2a_i^T x_j + \|x_j\|^2 \\
&= (e_i - e_j)^T \begin{bmatrix} A^T \\ X^T \end{bmatrix} \begin{bmatrix} A & X \end{bmatrix} (e_i - e_j) \\
&= B_{ij}^T \begin{bmatrix} I \\ X^T \end{bmatrix} \begin{bmatrix} I & X \end{bmatrix} B_{ij} \\
&= \left\langle B_{ij} B_{ij}^T, \begin{bmatrix} I & X \\ X^T & X^T X \end{bmatrix} \right\rangle_F
\end{aligned}$$

with $B_{ij} \stackrel{\text{def}}{=} \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix} (e_i - e_j)$.

<u>Fact</u>:

$$\begin{bmatrix} I & X \\ X^T & Y \end{bmatrix} \succeq 0 \text{ has rank } d \quad \Longleftrightarrow \quad Y = X^T X$$

Thus

$$
\begin{aligned}
\upsilon_{\text{opt}} = & \min_{X,Y} \sum_{(i,j)\in\mathcal{A}} \left| \left\langle B_{ij} B_{ij}^T, Z \right\rangle_F - d_{ij}^2 \right| \\
& \text{s.t. } Z = \begin{bmatrix} I & X \\ X^T & Y \end{bmatrix} \succeq 0, \ \text{rank} Z = d
\end{aligned}
$$

$$
\boxed{
\begin{aligned}
\upsilon_{\text{sdp}} \stackrel{\text{def}}{=} & \min_{X,Y} \sum_{(i,j)\in\mathcal{A}} \left| \left\langle B_{ij} B_{ij}^T, Z \right\rangle_F - d_{ij}^2 \right| \\
& \text{s.t. } Z = \begin{bmatrix} I & X \\ X^T & Y \end{bmatrix} \succeq 0
\end{aligned}
}
$$

- Biswas and Ye gave probabilistic interpretation of SDP soln, and proposed a distributed (domain partitioning) method for solving SDP when $n > 100$.

Second-order cone program (SOCP) is easier to solve than SDP.

Q: Is SOCP relaxation a good approximation? Or a mixed SDP-SOCP relaxation?

Q: How to solve SOCP relaxation?

SOCP Relaxation:

$$
\begin{aligned}
v_{\mathrm{opt}} = \min_{x_{m+1},\ldots,x_n,y_{ij}} & \sum_{(i,j)\in\mathcal{A}} |y_{ij} - d_{ij}^2| \\
\text{s.t.} \quad & y_{ij} = \|a_i - x_j\|^2 \;\; \forall i \le m < j,\; (i,j)\in\mathcal{A} \\
& y_{ij} = \|x_i - x_j\|^2 \;\; \forall m < i < j,\; (i,j)\in\mathcal{A}
\end{aligned}
$$

$$
\boxed{
\begin{aligned}
v_{\mathrm{socp}} \stackrel{\mathrm{def}}{=} \min_{x_{m+1},\ldots,x_n,y_{ij}} & \sum_{(i,j)\in\mathcal{A}} |y_{ij} - d_{ij}^2| \\
\text{s.t.} \quad & y_{ij} \ge \|a_i - x_j\|^2 \;\; \forall i \le m < j,\; (i,j)\in\mathcal{A} \\
& y_{ij} \ge \|x_i - x_j\|^2 \;\; \forall m < i < j,\; (i,j)\in\mathcal{A}
\end{aligned}
}
$$

$$
\begin{aligned}
& y \ge \|x\|^2 \\
\Longleftrightarrow\ & (y + \tfrac{1}{2})^2 \ge (y - \tfrac{1}{2})^2 + \|x\|^2 \\
\Longleftrightarrow\ & y + \tfrac{1}{2} \ge \|(y - \tfrac{1}{2}, x)\|
\end{aligned}
$$

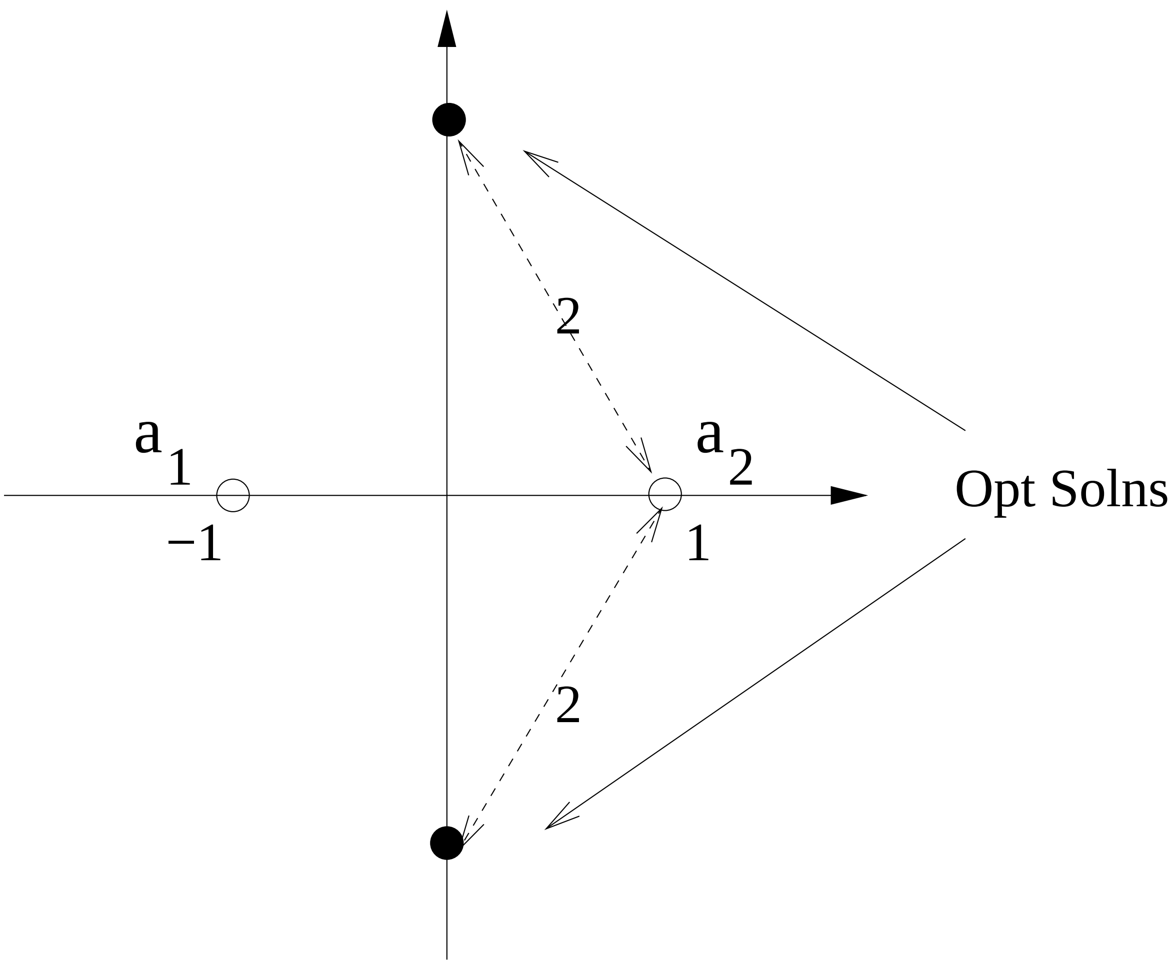
(also Doherty,Pister,El Ghaoui '03)

Properties of SDP, SOCP Relaxations:

$d = 2$, $n = 3$, $m = 2$, $d_{13} = d_{23} = 2$

Problem:

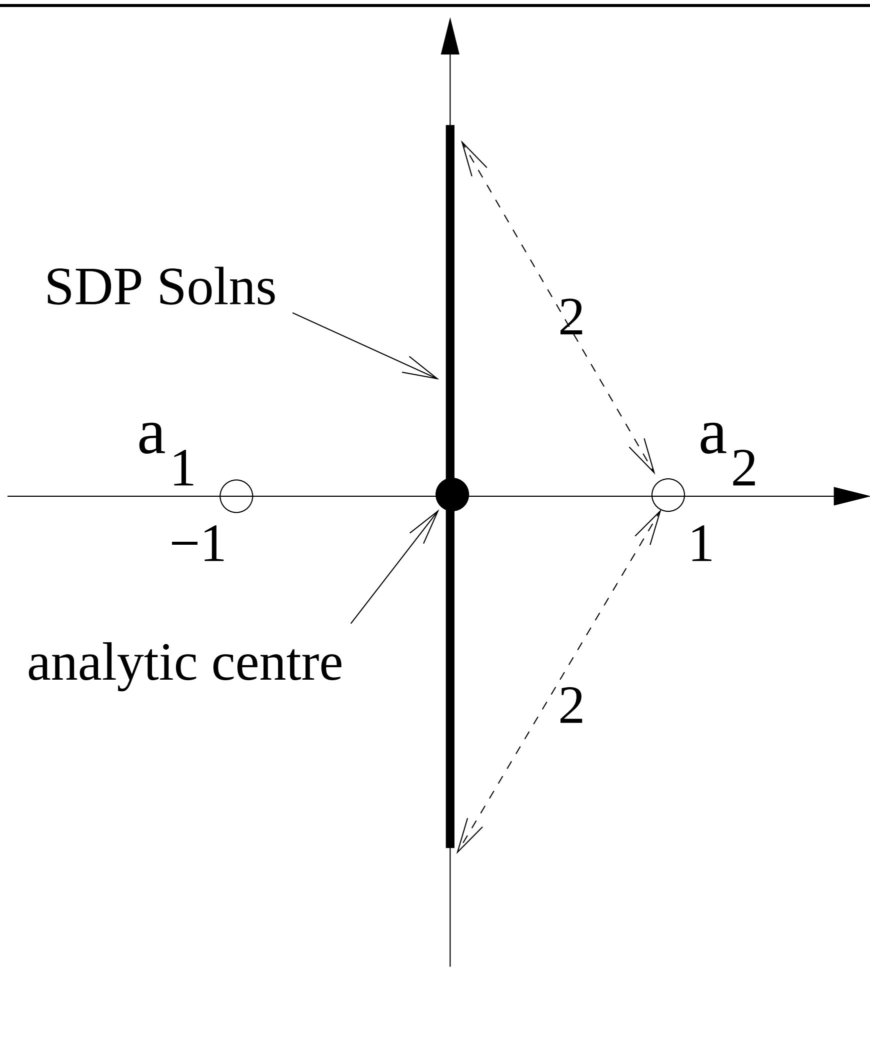
$$\boxed{\begin{aligned} 0 = \min_{x=(\alpha,\beta)\in\Re^2} & |(1-\alpha)^2 + \beta^2 - 4| \\ & + |(-1-\alpha)^2 + \beta^2 - 4| \end{aligned}}$$



SDP Relaxation:

$$0 = \min_{\substack{x=(\alpha,\beta)\in\Re^2 \\ y\in\Re}} |y - 2\alpha - 3| + |y + 2\alpha - 3|$$

$$\text{s.t.} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \beta \\ 0 & \beta & y \end{bmatrix} \succeq 0$$



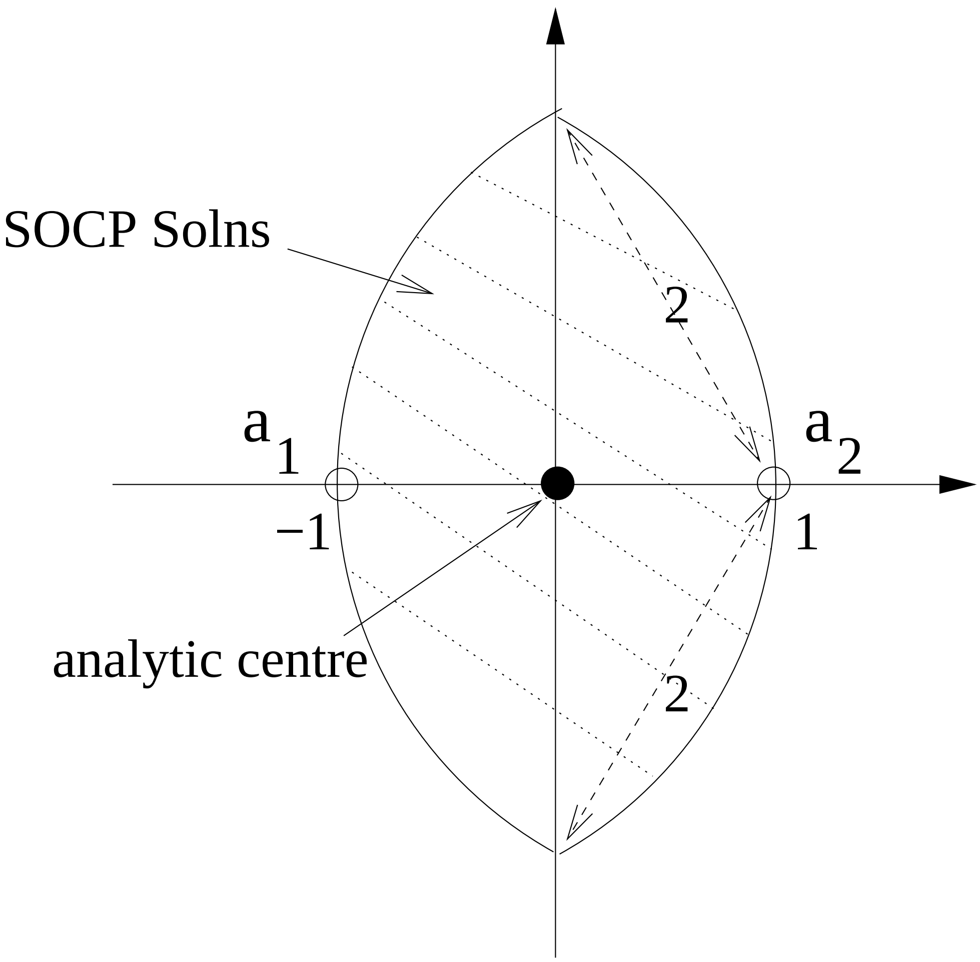
If solve SDP by IP method, then likely get analy. centre.

SOCP Relaxation:

$$0 = \min_{\substack{x=(\alpha,\beta)\in\Re^2 \\ y,z\in\Re}} |y-4| + |z-4|$$

$$\text{s.t.} \quad y \geq (1-\alpha)^2 + \beta^2,$$

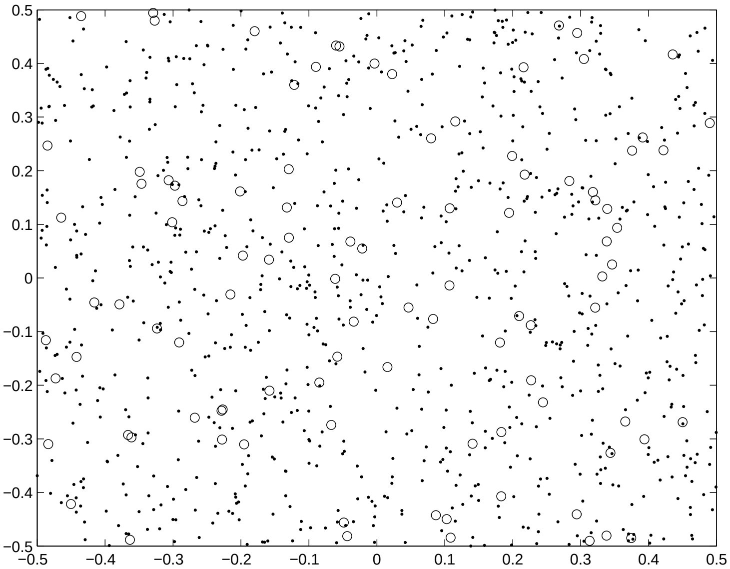
$$z \geq (-1-\alpha)^2 + \beta^2$$



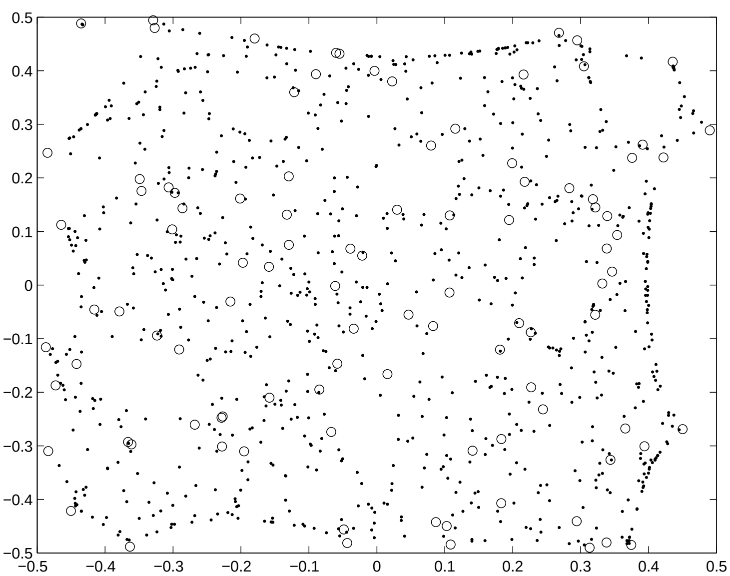
If solve SOCP by IP method, then likely get analy. centre.

Fact 1. If $\{x_i, y_{ij}\}_{m<i\leq n,(i,j)\in\mathcal{A}}$ is the analytic centre soln of SOCP, then

$$x_i \in \operatorname{conv}\{x_j, a_k\}_{(i,k),(i,j)\in\mathcal{A}} \quad \forall i$$



Opt soln ($m = 100$, $n = 1000$, nhbrs if dist$< .06$)



SOCP soln found by IP method (SeDuMi)

Fact 2. If $v_{\text{opt}} = 0$ and $\{x_i, y_{ij}\}_{m<i\leq n,(i,j)\in\mathcal{A}}$ is a maximally strictly feasible soln (e.g., analytic centre) of SOCP, then for each i,

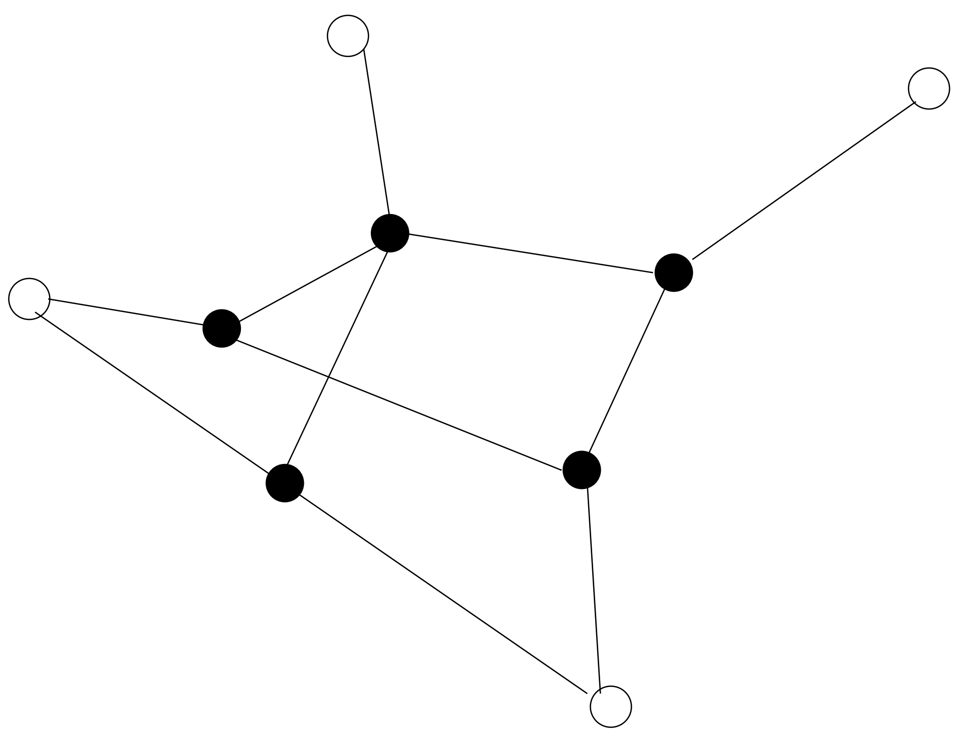
$$\begin{array}{rl} & x_i \text{ appears in every optimal soln} \\ \iff & \|x_i - x_j\|^2 = d_{ij}^2 \quad \forall j : (i,j) \in \mathcal{A} \end{array}$$

Conjectures:

1. If $v_{\text{opt}} = 0$ and, for some $\mathcal{N} \subseteq \{m+1, ..., n\}$ and some optimal soln $\{x_i, y_{ij}\}_{m<i\leq n,(i,j)\in\mathcal{A}}$, we have

$$x_i \in \text{conv}\,\{x_j, a_k\}_{(i,k),(i,j)\in\mathcal{A},j\in\mathcal{N}} \quad \forall i \in \mathcal{N},$$

then $\{x_i\}_{i\in\mathcal{N}}$ appears in every SOCP (SDP) soln.



Mixed SDP-SOCP Relaxation:

Partition $\mathcal{A}$ into $\mathcal{A}_1$, $\mathcal{A}_2$.

$$
\begin{array}{ll}
\min & \displaystyle\sum_{(i,j)\in\mathcal{A}_1} |\left\langle B_{ij}B_{ij}^T, Z\right\rangle_F - d_{ij}^2| \\
& \displaystyle + \sum_{\substack{i\le m<j \\ (i,j)\in\mathcal{A}_2}} |y_{ij} - d_{ij}^2| + \sum_{\substack{m<i<j \\ (i,j)\in\mathcal{A}_2}} |y_{ij} - d_{ij}^2| \\
\text{s.t.} & Z = \begin{bmatrix} I & X \\ X^T & Y \end{bmatrix} \succeq 0 \\
& X = [x_i]_{(i,\cdot)\in\mathcal{A}_1} \\
& y_{ij} \ge \|a_i - x_j\|^2 \ \forall i\le m<j,\ (i,j)\in\mathcal{A}_2 \\
& y_{ij} \ge \|x_i - x_j\|^2 \ \forall m<i<j,\ (i,j)\in\mathcal{A}_2
\end{array}
$$

Easier to solve than SDP? As good a relaxation?

Solving SOCP Relaxation:

$$\min_{x_{m+1},\ldots,x_n,y_{ij}} \sum_{(i,j)\in\mathcal{A}} |y_{ij} - d_{ij}^2|$$

$$\begin{aligned} \text{s.t.} \quad & y_{ij} \geq \|a_i - x_j\|^2 \ \forall i \leq m < j, \ (i,j)\in\mathcal{A} \\ & y_{ij} \geq \|x_i - x_j\|^2 \ \forall m < i < j, \ (i,j)\in\mathcal{A} \end{aligned}$$

Put into standard form

$$\min \quad c^T x \quad \text{s.t.} \quad Ax = b, \ x \in K:$$

$$\min \quad \sum_{(i,j)\in\mathcal{A}} u_{ij} + v_{ij}$$

$$\begin{aligned} \text{s.t.} \quad & y_{ij} - u_{ij} + v_{ij} = d_{ij}^2 \ \forall (i,j)\in\mathcal{A} \\ & a_i - x_j - w_{ij} = 0 \ \forall i \leq m < j, \ (i,j)\in\mathcal{A} \\ & x_i - x_j - w_{ij} = 0 \ \forall m < i < j, \ (i,j)\in\mathcal{A} \\ & \alpha_{ij} = 1 \ \forall (i,j)\in\mathcal{A} \\ & y_{ij}\alpha_{ij} \geq \|w_j\|^2 \ \forall (i,j)\in\mathcal{A} \end{aligned}$$

Simulation:

- Uniformly generate $\tilde{x}_1, ..., \tilde{x}_n$ in $[-.5, .5]^2$, $m = n/10$, two pts are nhbrs if dist$<$ .06. Set $\quad d_{ij} = \|\tilde{x}_i - \tilde{x}_j\| \quad$ (Biswas, Ye '03)
- Solve SOCP using SeDuMi 1.05 (Sturm '01), which implements a predictor-corrector IP method. Time is on a Dec Alpha. Err $= \sum_i \|x_i^{\text{SeDumi}} - \tilde{x}_i\|^2$.

n	dim(A)	secs	Err
1000	21472×33908	330	.48
2000	84440×130060	12548	.057

Solve SOCP faster? (Ongoing work...)

Idea 1: Adapt the domain partitioning approach of Biswas-Ye, using SOCP instead of SDP.

- Overlapping subdomains to reduce boundary effect.
- Upon solving SOCP on a subdomain, turn x_i into an anchor if it has at least 2 nhbrs in the same subdomain and it satisfies

$$y_{ij} \leq \|x_i - x_j\|^2 + \text{tol}$$

for all neighbors j (anchor or not) in the same subdomain.
- Postprocessing: For those (i, j) with

$$\|x_i - x_j\|^2 > d_{ij}^2 + \text{tol},$$

readjust x_i, x_j.

Idea 2: Similar to Idea 1, but use mixed SDP-SOCP relaxation instead.

- SDP for those $(i, j) \in \mathcal{A}$ such that either i or j corresponds to an anchor near the subdomain boundary.
- SOCP for the remaining $(i, j) \in \mathcal{A}$ (either i or j is an anchor in the interior of subdomain).

Idea 3:

$$\min_{y \geq z} |y - d^2| = \max\{0, z - d^2\}$$

So SOCP relaxation:

$$\min_{x_{m+1},\ldots,x_n} \sum_{\substack{i \leq m < j \\ (i,j) \in \mathcal{A}}} \max\{0, \|a_i - x_j\|^2 - d_{ij}^2\} + \sum_{\substack{m < i < j \\ (i,j) \in \mathcal{A}}} \max\{0, \|x_i - x_j\|^2 - d_{ij}^2\}$$

This is an unconstrained (nondiff) convex program.

- Smooth approximation:

$$\max\{0, t\} \;\approx\; \mu h(t/\mu) \;\; (\mu > 0)$$

h smooth convex, $\lim_{t \to -\infty} h(t) = \lim_{t \to \infty} h(t) - t = 0$. We use $h(t) = ((t^2 + 4)^{1/2} + t)/2$ (CHKS).

SOCP approximation:

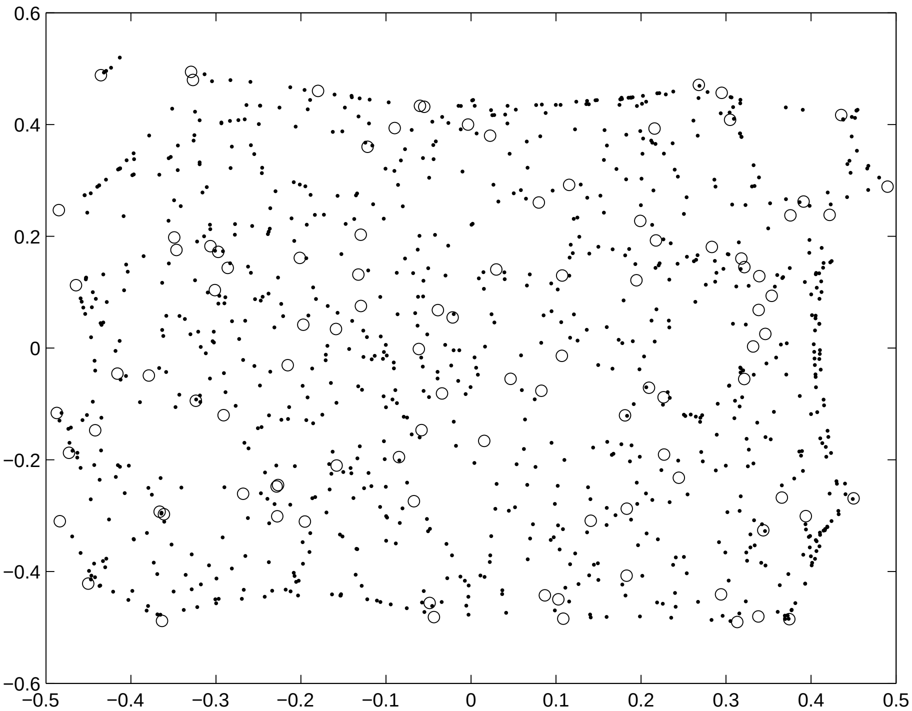
$$
\begin{aligned}
\min \quad & f_\mu(x_{m+1},..,x_n) \\
& \stackrel{\text{def}}{=} \sum_{\substack{i\le m<j\\(i,j)\in\mathcal{A}}} \mu h\left(\frac{\|a_i - x_j\|^2 - d_{ij}^2}{\mu}\right) \\
& \quad + \sum_{\substack{m<i<j\\(i,j)\in\mathcal{A}}} \mu h\left(\frac{\|x_i - x_j\|^2 - d_{ij}^2}{\mu}\right)
\end{aligned}
$$

Solve the smooth approximation using block coordinate descent (BCD):

- If $\|\nabla_{x_i} f_\mu\| = \Omega(\mu)$, then update x_i by moving it along the Newton direction $-[\nabla^2_{x_i x_i} f_\mu]^{-1}\nabla_{x_i} f_\mu$, and re-iterate.
- Decrease μ when $\|\nabla_{x_i} f_\mu\| = O(\mu)\ \forall i$.

$\mu^{\text{init}} = 1e-3$. $\mu^{\text{end}} = 2e-6$. Decrease μ by a factor of 5. Code in Matlab. Time is on a Dec Alpha.

n	secs	Err
1000	373	.48
2000	2090	.052



SOCP soln found by BCD method

$(m = 10, n = 1000)$

- Better time and soln accuracy than using SeDuMi?
- Computation easily distributes.
- Code in Fortran (instead of Matlab) to improve time?

Idea 4:

Interpret SOCP relaxation as the dual of a d-commodity convex network flow problem.

$$v_{\mathrm{socp}} = \min_{\substack{x_{m+1},\ldots,x_n \\ x_1=a_1,\ldots,x_m=a_m}} \sum_{(i,j)\in\mathcal{A}} f_{ij}(x_i - x_j),$$

where $f_{ij}(t) \stackrel{\mathrm{def}}{=} \max\{0, \|t\|^2 - d_{ij}^2\}$.

- For $d = 1$, this can be solved efficiently using ϵ-relaxation method (a generalization of preflow-push for nonlinear single commodity flow) (Bertsekas,Polymenakos,T '97). Computation easily distributes.

- Q: For $d = 2$, can the ϵ-relaxation method be extended?

Conclusions

- SOCP relaxation may be useful in solving optimization problems involving Euclidean geometry.

- Better understanding of properties of SOCP soln is needed.

- Fast methods for solving SOCP? Exploiting special structures of SOCP?