On Solving SOCPs
with an
Interior-Point Method for NLP

Robert Vanderbei

and

Hande Y. Benson

INFORMS Nov 5, 2001
Miami Beach, FL

Operations Research and Financial Engineering, Princeton University

http://www.princeton.edu/∼rvdb

1 Outline

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

Theme. Modeling Matters.

1 Outline

- Introduction and Review of Problem Classes: NLP, SOCP

- Formulating SOCPs as Smooth Convex NLPs

- Applications and Computational Results

Theme. Modeling Matters.

1 Outline

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

Theme. Modeling Matters.

Outline

- Introduction and Review of Problem Classes: NLP, SOCP

- Formulating SOCPs as Smooth Convex NLPs

- Applications and Computational Results

1 Outline

- Introduction and Review of Problem Classes: NLP, SOCP

- Formulating SOCPs as Smooth Convex NLPs

- Applications and Computational Results

Theme. Modeling Matters.

2 Traditional Classes of Optimization Problems

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.

2 Traditional Classes of Optimization Problems

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.

2 Traditional Classes of Optimization Problems

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.

2 Traditional Classes of Optimization Problems

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.

2 Traditional Classes of Optimization Problems

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.

3 Second-Order Cone Programming (SOCP)

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

Here,

- f is an n-vector,
- 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.

3 Second-Order Cone Programming (SOCP)

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

Here,

- f is an n-vector,
- 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.

4 Preview of Applications

- FIR Filter Design
- Antenna Array Optimization
- Structural Optimization
- Grasping Problems
- Steiner Tree Problem
- Euclidean Multiple Facility Location
- Plastic Deformation
- Springs in Equilibrium
- Markowitz models in Finance

More later on some of these.

4 Preview of Applications

- FIR Filter Design
- Antenna Array Optimization
- Structural Optimization
- Grasping Problems
- Steiner Tree Problem
- Euclidean Multiple Facility Location
- Plastic Deformation
- Springs in Equilibrium
- Markowitz models in Finance

More later on some of these.

5 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, $x \geq 0$, 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.

5 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, $x \geq 0$, 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.

5 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, $x \geq 0$, 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.

5 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, $x \geq \mathbf{0}$, 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.

5 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, $x \geq \mathbf{0}$, 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.

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.

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.

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

8 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...

8 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...

8 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...

8 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...

9 An Example

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

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.

9 An Example

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

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.	
$t^2 - \|u\|^2 \geq 0, \quad t \geq 0$	nonconcave	equiv.	
$t - \|u\|^2/t \geq 0, \quad t > 0$	concave	equiv.	interior

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.	
$t^2 - \|u\|^2 \geq 0, \quad t \geq 0$	nonconcave	equiv.	
$t - \|u\|^2/t \geq 0, \quad t > 0$	concave	equiv.	interior

10 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.	
$t^2 - \|u\|^2 \geq 0, \quad t \geq 0$	nonconcave	equiv.	
$t - \|u\|^2/t \geq 0, \quad t > 0$	concave	equiv.	interior

10 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.	
$t^2 - \|u\|^2 \geq 0, \quad t \geq 0$	nonconcave	equiv.	
$t - \|u\|^2/t \geq 0, \quad t > 0$	concave	equiv.	interior

10 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.	
$t^2 - \|u\|^2 \geq 0, \quad t \geq 0$	nonconcave	equiv.	
$t - \|u\|^2/t \geq 0, \quad t > 0$	concave	equiv.	interior

10 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.	
$t^2 - \|u\|^2 \geq 0, \quad t \geq 0$	nonconcave	equiv.	
$t - \|u\|^2/t \geq 0, \quad t > 0$	concave	equiv.	interior

Which Formulation is Best?

Problem	socp	ϵ-pert	nonconvex	ratio
antenna [6]	5.728 (40)	9.664 (60)	34.169 (201)	3.365 (28)
emfl	318.869 (201)	29.332 (24)	277.499 (201)	61.739 (30)
fir	0.42 (37)	0.42 (37)	0.47 (42)	0.46 (40)
grasp	0.31 (201)	0.04 (37)	0.34 (201)	0.04 (29)
minsurf	14.511 (34)	7.811 (20)	109.948 (201)	7.28 (17)
springs	0.01 (17)	0.02 (17)	0.01 (15)	0.01 (15)
steiner	0.861 (201)	0.11 (27)	0.811 (201)	0.16 (41)
structure	68.829 (201)	10.645 (43)	81.186 (201)	12.898 (54)

- These problems are AMPL encodings of problems in Lobo, Vandenberghe, Boyd, and Lebret.
- “Nonconvex” refers to the “square-both-sides” reformulation.
- Numbers in parens are iteration counts.
- Red indicates failure.

Which Formulation is Best?

Problem	socp	ϵ-pert	nonconvex	ratio
antenna [6]	5.728 (40)	9.664 (60)	34.169 (201)	3.365 (28)
emfl	318.869 (201)	29.332 (24)	277.499 (201)	61.739 (30)
fir	0.42 (37)	0.42 (37)	0.47 (42)	0.46 (40)
grasp	0.31 (201)	0.04 (37)	0.34 (201)	0.04 (29)
minsurf	14.511 (34)	7.811 (20)	109.948 (201)	7.28 (17)
springs	0.01 (17)	0.02 (17)	0.01 (15)	0.01 (15)
steiner	0.861 (201)	0.11 (27)	0.811 (201)	0.16 (41)
structure	68.829 (201)	10.645 (43)	81.186 (201)	12.898 (54)

- These problems are AMPL encodings of problems in Lobo, Vandenberghe, Boyd, and Lebret.
- “Nonconvex” refers to the “square-both-sides” reformulation.
- Numbers in parens are iteration counts.
- Red indicates failure.

Which Formulation is Best?

Problem	socp	ϵ-pert	nonconvex	ratio
antenna [6]	5.728 (40)	9.664 (60)	34.169 (201)	3.365 (28)
emfl	318.869 (201)	29.332 (24)	277.499 (201)	61.739 (30)
fir	0.42 (37)	0.42 (37)	0.47 (42)	0.46 (40)
grasp	0.31 (201)	0.04 (37)	0.34 (201)	0.04 (29)
minsurf	14.511 (34)	7.811 (20)	109.948 (201)	7.28 (17)
springs	0.01 (17)	0.02 (17)	0.01 (15)	0.01 (15)
steiner	0.861 (201)	0.11 (27)	0.811 (201)	0.16 (41)
structure	68.829 (201)	10.645 (43)	81.186 (201)	12.898 (54)

- These problems are AMPL encodings of problems in Lobo, Vandenberghe, Boyd, and Lebret.
- “Nonconvex” refers to the “square-both-sides” reformulation.
- Numbers in parens are iteration counts.
- Red indicates failure.

Which Formulation is Best?

Problem	socp	ϵ-pert	nonconvex	ratio
antenna [6]	5.728 (40)	9.664 (60)	34.169 (201)	3.365 (28)
emfl	318.869 (201)	29.332 (24)	277.499 (201)	61.739 (30)
fir	0.42 (37)	0.42 (37)	0.47 (42)	0.46 (40)
grasp	0.31 (201)	0.04 (37)	0.34 (201)	0.04 (29)
minsurf	14.511 (34)	7.811 (20)	109.948 (201)	7.28 (17)
springs	0.01 (17)	0.02 (17)	0.01 (15)	0.01 (15)
steiner	0.861 (201)	0.11 (27)	0.811 (201)	0.16 (41)
structure	68.829 (201)	10.645 (43)	81.186 (201)	12.898 (54)

- These problems are AMPL encodings of problems in Lobo, Vandenberghe, Boyd, and Lebret.
- “Nonconvex” refers to the “square-both-sides” reformulation.
- Numbers in parens are iteration counts.
- Red indicates failure.

Which Formulation is Best?

Problem	socp	ϵ-pert	nonconvex	ratio
antenna [6]	5.728 (40)	9.664 (60)	34.169 (201)	3.365 (28)
emfl	318.869 (201)	29.332 (24)	277.499 (201)	61.739 (30)
fir	0.42 (37)	0.42 (37)	0.47 (42)	0.46 (40)
grasp	0.31 (201)	0.04 (37)	0.34 (201)	0.04 (29)
minsurf	14.511 (34)	7.811 (20)	109.948 (201)	7.28 (17)
springs	0.01 (17)	0.02 (17)	0.01 (15)	0.01 (15)
steiner	0.861 (201)	0.11 (27)	0.811 (201)	0.16 (41)
structure	68.829 (201)	10.645 (43)	81.186 (201)	12.898 (54)

- These problems are AMPL encodings of problems in Lobo, Vandenberghe, Boyd, and Lebret.
- “Nonconvex” refers to the “square-both-sides” reformulation.
- Numbers in parens are iteration counts.
- Red indicates failure.

Is SOCP necessary?

Problem	simplest nlp	socp
emfl	0.59 (17)	29.332 (24)
minsurf	1.883 (16)	7.28 (17)
steiner	0.2 (57)	0.11 (27)
structure	1.211 (17)	10.645 (43)
random LP (50x100)	0.44 (15)	1.041 (30)
random LP (200x500)	26.468 (18)	75.679 (42)

- LP's were converted to SOCP's using

$$x_j \geq 0, x_{j-1} \geq 0 \Leftrightarrow |x_j - x_{j-1}| \leq x_j + x_{j-1}.$$

Is SOCP necessary?

Problem	simplest nlp	socp
emfl	0.59 (17)	29.332 (24)
minsurf	1.883 (16)	7.28 (17)
steiner	0.2 (57)	0.11 (27)
structure	1.211 (17)	10.645 (43)
random LP (50x100)	0.44 (15)	1.041 (30)
random LP (200x500)	26.468 (18)	75.679 (42)

- LP's were converted to SOCP's using

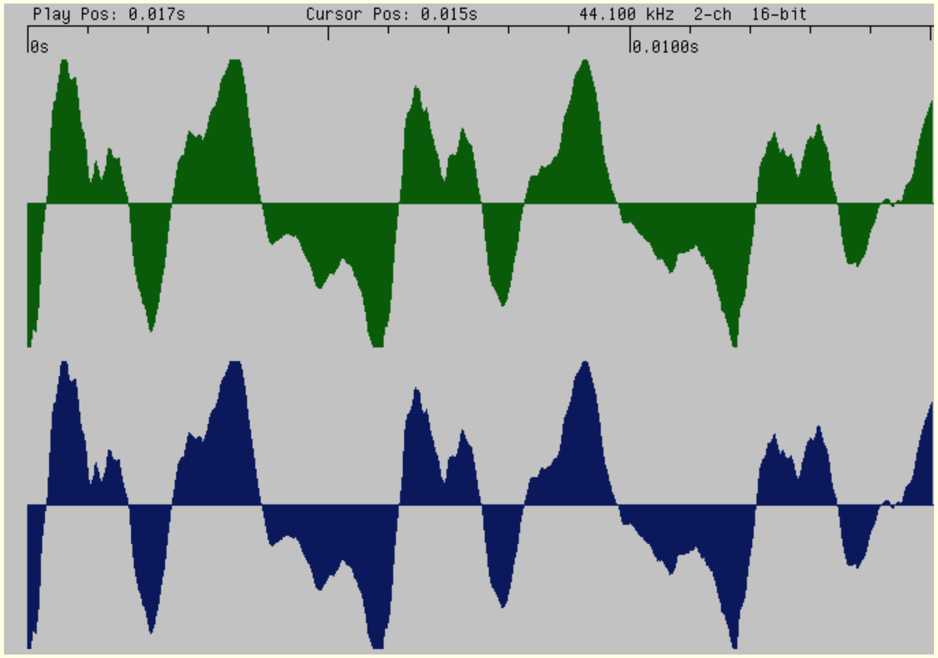
$$x_j \geq 0, x_{j-1} \geq 0 \Leftrightarrow |x_j - x_{j-1}| \leq x_j + x_{j-1}.$$

Applications and Computational Results

Mostly inspired by Lobo, Vandenberghe, Boyd, and Lebret, Applications of Second-Order Cone Programming

Finite Impulse Response (FIR) Filter Design

- Audio is stored digitally in a computer as a stream of short integers: u_k, $k \in \mathbb{Z}$.
- When the music is played, these integers are used to drive the displacement of the speaker from its resting position.
- For CD quality sound, 44100 short integers get played per second per channel.



0	-32768	8	-23681	16	12111
1	-32768	9	-18449	17	17311
2	-32768	10	-11025	18	21311
3	-30753	11	-6913	19	23055
4	-28865	12	-4337	20	23519
5	-29105	13	-1329	21	25247
6	-29201	14	1743	22	27535
7	-26513	15	6223	23	29471

FIR Filter Design—Continued

- A finite impulse response (FIR) filter takes as input a digital signal and convolves this signal with a finite set of fixed numbers $h_{-n}, \ldots, h_n$ to produce a filtered output signal:

$$y_k = \sum_{i=-n}^{n} h_i u_{k-i}.$$

- Sparing the details, the output power at frequency ν is given by

$$|H(\nu)|$$

where

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

- Similarly, the mean squared deviation from a flat frequency response over a frequency range, say $\mathcal{L} \subset [0, 1]$, is given by

$$\frac{1}{|\mathcal{L}|} \int_{\mathcal{L}} \left| H(\nu) - 1 \right|^2 d\nu$$

FIR Filter Design—Low Pass Filter

$$
\begin{array}{ll}
\text{minimize} & \rho \\[2em]
\text{subject to} & \left(\dfrac{1}{|\mathcal{L}|} \displaystyle\int_{\mathcal{L}} \left| H(\nu) - 1 \right|^2 d\nu \right)^{1/2} \le \rho \\[1em]
& \qquad\qquad\qquad\qquad |H(\nu)| \le \rho \qquad \nu \in \mathcal{H}
\end{array}
$$

where

$$
\begin{aligned}
H(\nu) &= \sum_{k=-5}^{19} h(k) e^{2\pi i k \nu}, \\
h(k) &= \text{Complex filter coefficients, i.e., } \textbf{decision variables} \\
\mathcal{L} &= [0.1, 0.5] \\
\mathcal{H} &= [0.6, 0.9]
\end{aligned}
$$

Discretizing the integral, this is an SOCP.

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.

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.

Low Pass Filter—Reformulation as QCLP

- Replace ρ with $\sqrt{\sigma}$ everywhere except the objective funtion (since the square root function is monotone):

$$\begin{array}{ll} \text{minimize} & \sigma \\ \text{subject to} & \dfrac{1}{|\mathcal{L}|} \displaystyle\int_{\mathcal{L}} \left| H(\nu) - 1 \right|^2 d\nu \le \sigma \\ & \qquad\qquad\qquad\quad |H|^2(\nu) \le \sigma \qquad \nu \in \mathcal{H} \end{array}$$

- This variant involves smooth convex quadratic constraints.
- But, squared things vary over a larger dynamic range which might lead to numerical problems.
- Tried one example, $n = 14$, frequency discretized in 2000 parts.
- SOCP variant solves in 66 iterations and 12.4 seconds.
- QCLP variant solves in 65 iterations and 11.1 seconds.
- Not much difference in this case.

Filter Design: Woofer, Midrange, Tweeter

minimize ρ

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

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

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

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

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"

$$\begin{aligned} \text{filter length: } n &= 14 \\ \text{integral discretization: } N &= 1000 \end{aligned}$$

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 demo

Specific Example: Pink Floyd's "Money"

filter length: $n = 14$

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 demo

Wide-Band Antenna Array Design

- Given: a linear array (or a 2-D grid) of radar antennae.
- An incoming signal produces a signal at each antenna.
- A linear combination of the signals is made to produce one output signal.
- Coefficients of the linear combination can be chosen to accentuate and/or attenuate the output signal's strength as a function of the input signal's source direction.
- Similar to FIR filter design (if freq of incoming signal is fixed).
- The set of antennae is analogous to the set of time delays in FIR filter design.
- The direction of the input signal is analogous to frequency in FIR filter design.
- Wide band means that we consider a range of frequencies. This adds an extra dimension to the problem (literally).

Wide-Band Antenna Array Design—Continued

$$\begin{array}{ll} \text{minimize} & \alpha \\ \text{subject to} & \iint_{(\theta,\nu)\in S} |A(\theta,\nu)|^2 d\theta d\nu \le \alpha, \\ & |A(\theta,\nu)| \le 10^{-25/20}, \quad (\theta,\nu) \in S \\ & |A(\theta,\nu)| \le 10^{-45/20}, \quad (\theta,\nu) \in S_0 \\ & \int_{\nu\in P} |A(\theta_m,\nu) - \beta_m|^2 d\nu \le 10^{-50/10}, \quad m = 1,\dots,M \end{array}$$

where

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

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

P = subset of direction/freq pairs representing passband

S = subset of direction/freq pairs representing sidelobe

S_0 = subset of sidelobe spelling NRL

$\{\theta_m\}$ = finite set of directions “covering” pass band

Specific Example

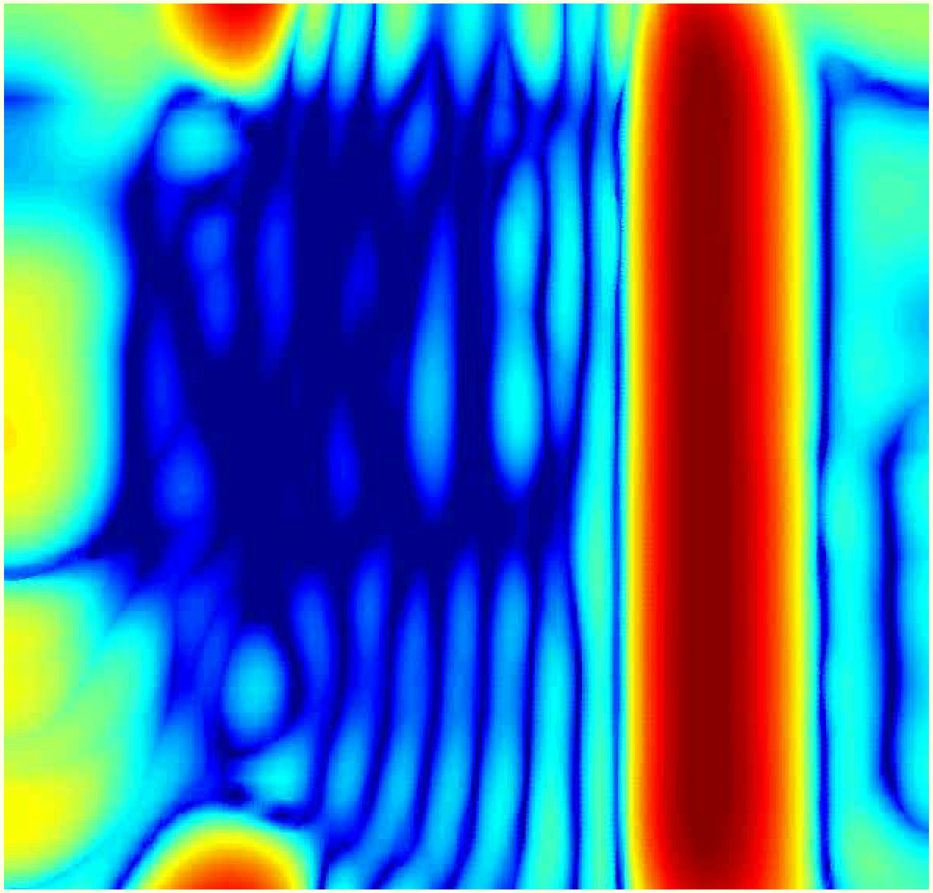
15	antennae in a linear array
21	“taps” on each array
671	Chebychev constraints to spell “NRL”

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

U.S. Naval Research Laboratory,

engr.umbc.edu/∼jeffc/pubs/papers/radar2Kds/radar2000.html

Solution and Animation



- Red represents high intensity (hot)
- Blue represents low intensity (cold)
- Horizontal axis represents angle of input signal
- Vertical axis represents freq of input signal
- This problem can be formulated either as an SOCP or as a QCLP.
- Click on image to run animation.

2-D Antenna-Array Design Problem

$$\begin{aligned} \text{minimize} \quad & \rho \\ \text{subject to} \quad & |A(p)|^2 \le \rho, \quad p \in S \\ & A(p_0) = 1, \end{aligned}$$

where

$$\begin{aligned} A(p) &= \sum_{l \in \{\text{array elements}\}} w_l e^{-2\pi i p \cdot l}, \quad p \in S \\ w_l &= \text{complex-valued } \textbf{design weight} \text{ for array element } l \\ S &= \text{subset of unit hemisphere: sidelobe directions} \\ p_0 &= \text{“look” direction} \end{aligned}$$

Specific Example: Hexagonal Lattice of 61 Elements

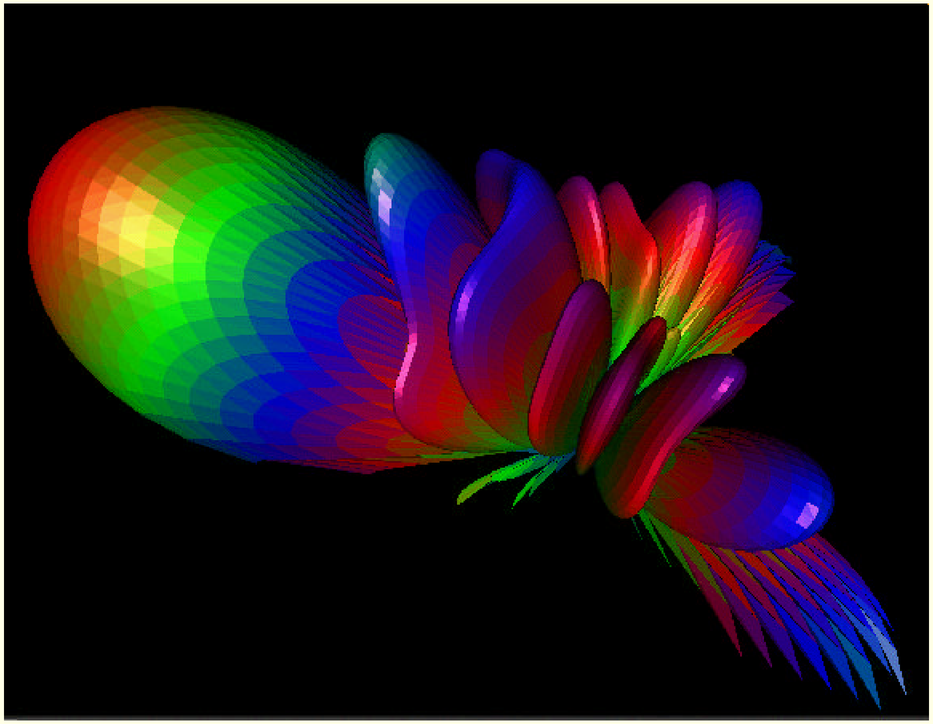
$$\rho = -20 \text{ dB} = 0.01$$

$$S = \text{889 points outside } 20^\circ \text{ from look direction}$$

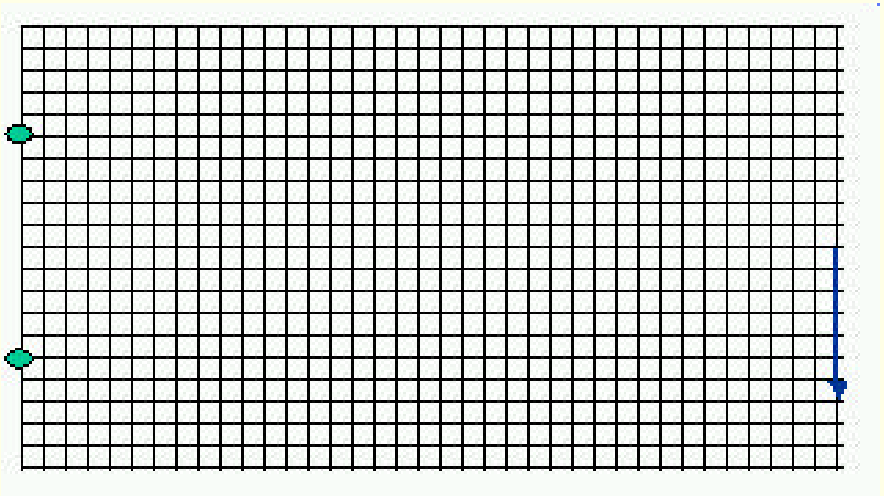
$$p_0 = 40^\circ \text{ from zenith}$$

constraints	839
variables	123
time (secs)	
LOQO	722
MINOS	> 60000
LANCELOT	55462
SNOPT	—

Solution



Structural Design



Given:

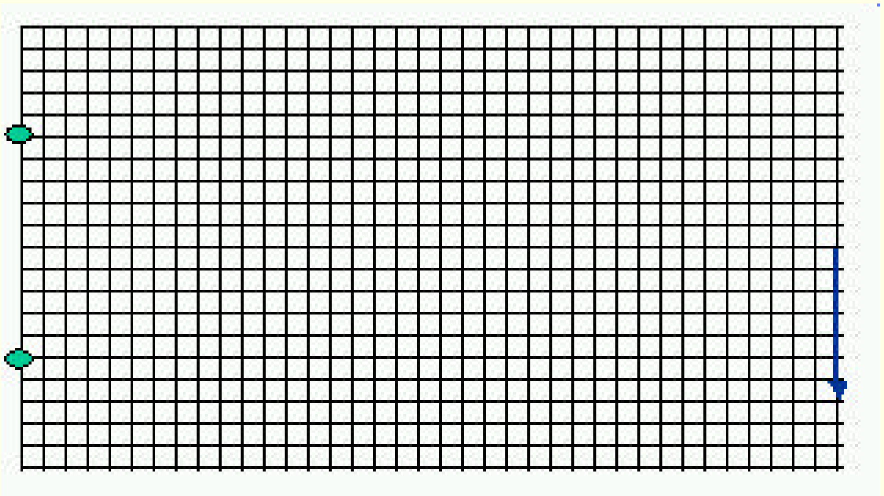
- A region of space in which to build something.
- Thing is essentially planar but with varying thickness.
- A place (or places) where the thing will be anchored.
- A place (or places) where loads will be applied.
- A certain total amount of material out of which to build the thing.

Objective: Design the thing to be as strong as possible.

Approach:

- Partition 2-D region into finite elements.
- Assign a thickness to each element.

Structural Design



Given:

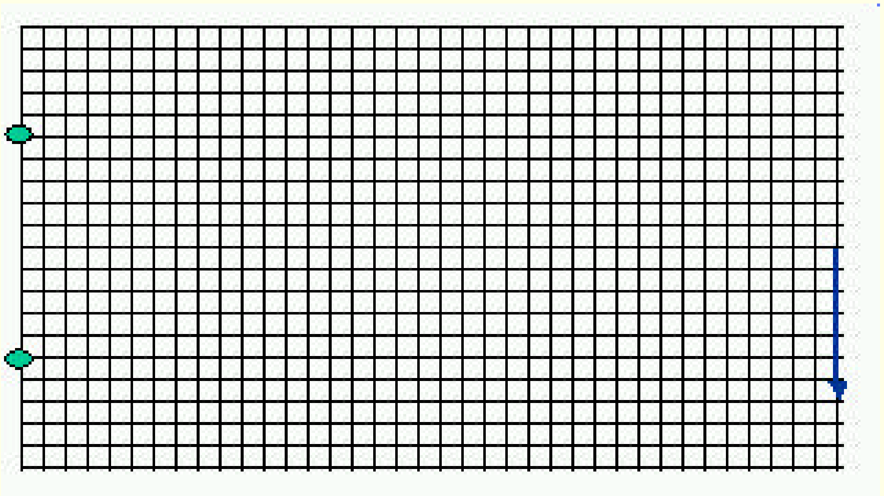
- A region of space in which to build something.
- Thing is essentially planar but with varying thickness.
- A place (or places) where the thing will be anchored.
- A place (or places) where loads will be applied.
- A certain total amount of material out of which to build the thing.

Objective: Design the thing to be as strong as possible.

Approach:

- Partition 2-D region into finite elements.
- Assign a thickness to each element.

Structural Design



Given:

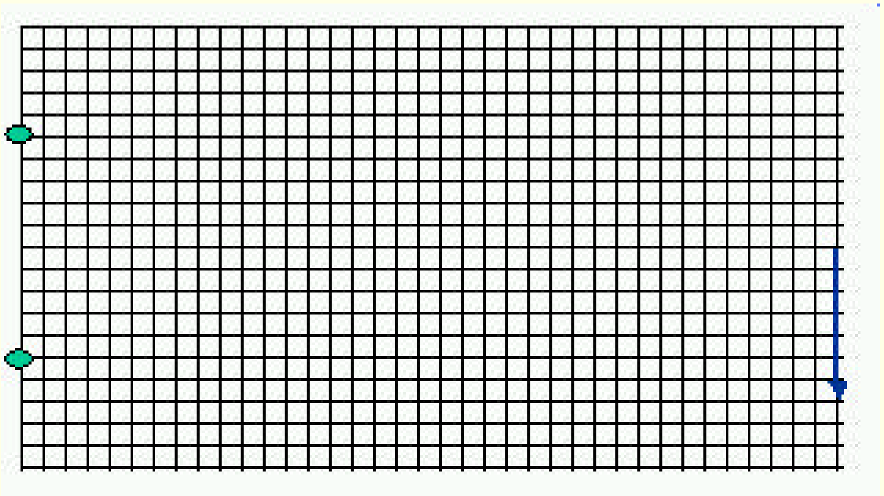
- A region of space in which to build something.
- Thing is essentially planar but with varying thickness.
- A place (or places) where the thing will be anchored.
- A place (or places) where loads will be applied.
- A certain total amount of material out of which to build the thing.

Objective: Design the thing to be as strong as possible.

Approach:

- Partition 2-D region into finite elements.
- Assign a thickness to each element.

Structural Design



Given:

- A region of space in which to build something.
- Thing is essentially planar but with varying thickness.
- A place (or places) where the thing will be anchored.
- A place (or places) where loads will be applied.
- A certain total amount of material out of which to build the thing.

Objective: Design the thing to be as strong as possible.

Approach:

- Partition 2-D region into finite elements.
- Assign a thickness to each element.

Structural Design—Continued

$$\begin{array}{ll} \text{minimize} & -p^T w \\ \text{subject to} & \dfrac{V}{A_e} w^T K_e w \le 1, \quad e \in \mathcal{E} \end{array}$$

where

p = applied load

w = node displacements; **optimization vars**

V = total volume

A_e = thickness of element e

K_e = element stiffness matrix ($\succeq 0$)

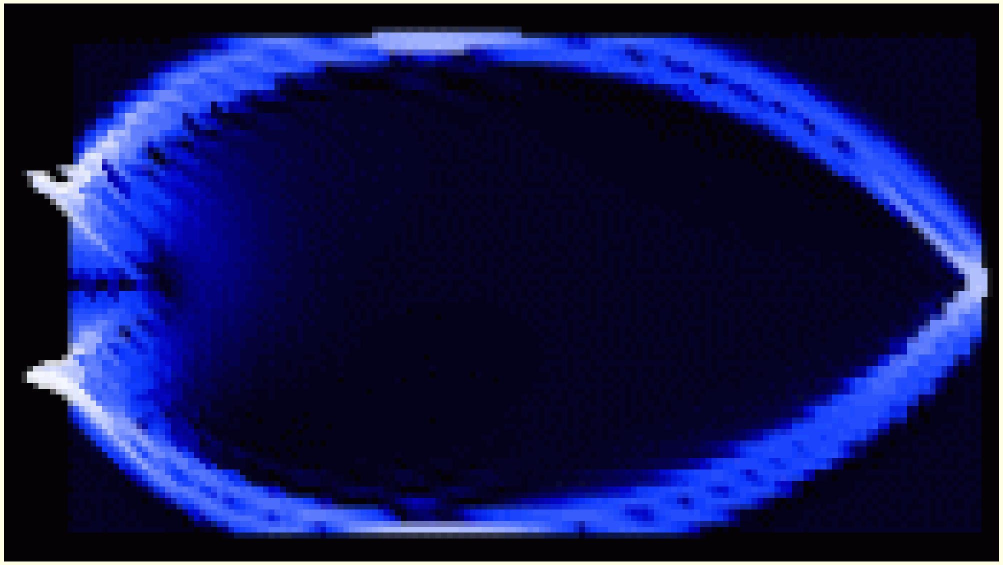
$\mathcal{E}$ = set of elements

Intrinsically a QCLP. Can be cast as an SOCP.

Specific Example: Michel Bracket

element grid	40x72	20x36	5x9
constraints	2880	720	45
variables	5965	1536	112
time (secs)			
LOQO	412	89.7	2.32
MINOS	∞	(IL)	(BS)
LANCELOT	∞	∞	15.73
SNOPT	-	(IS)	(BS)

Solution



Euclidean Multifacility Location

$$\text{minimize } \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} \|x_j - a_i\| + \sum_{j=1}^{n} \sum_{j'=1}^{j-1} v_{jj'} \|x_j - x_{j'}\|.$$

where

a_i = location of existing facilities, $i = 1, \ldots, m$

x_j = location of new facilities, $j = 1, \ldots, n$

Classification: not smooth, convex, not SOCP.

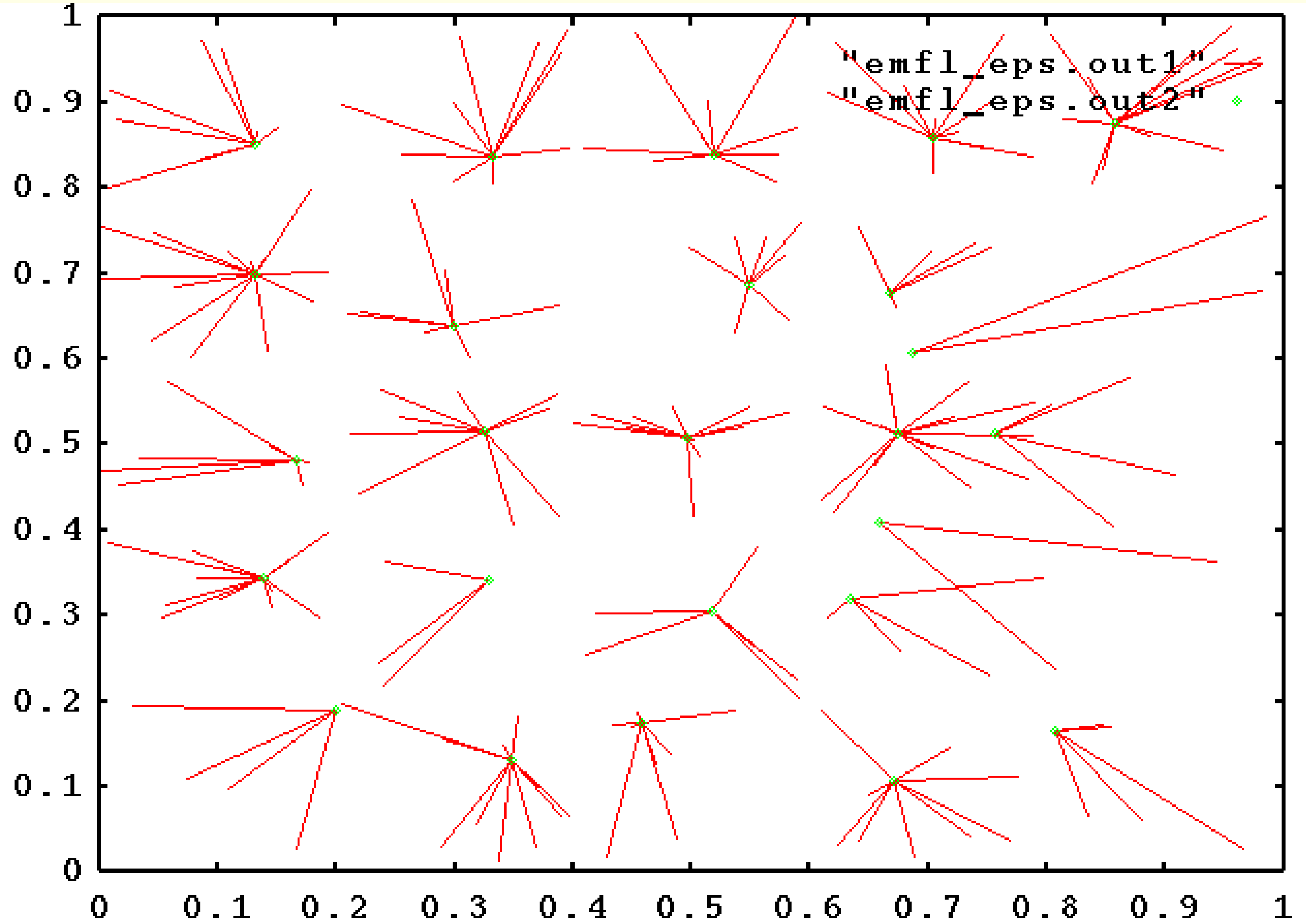
Example: Randomly Generated

$$\begin{aligned} m &= 200 \\ n &= 25 \end{aligned}$$

Used ϵ-perturbation for smoothing.

constraints	0
variables	1849
time (secs)	
LOQO	2.3
MINOS	9.7
LANCELOT	11.0
SNOPT	4.7

Solution


"emfl_eps.out1"
"emfl_eps.out2"
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1