Polyhedral approximation of the second-order cone: Computational experiments

François Glineur

Francois.Glineur@fpms.ac.be

Faculté Polytechnique de Mons

Delft University of Technology

HPOPT 99

Rotterdam, June 17, 1999

Outline

Introduction

- ◇ Problem definition

Approximation

- ◇ Principle
- ◇ Implementation

Test Problems

- ◇ Truss topology design
- ◇ Convex quadratic optimization
- ◇ Computational results
- ◇ Accuracy

Concluding remarks

Introduction

Second-Order Cone Optimization

Second-order (Lorentz) cone $\mathbb{L}^n$:

$$\mathbb{L}^n := \left\{ (x_0, x) \in \mathbb{R}_+ \times \mathbb{R}^n \mid \sum_{i=1}^n x_i^2 \leq x_0^2 \right\}$$

General SOCO problem :

$$\min c^T x \quad \text{s.t.} \quad \begin{cases} b^l \leq Ax \leq b^u, \ x^l \leq x \leq x^u, \\ x_{I_k} \in \mathbb{L}^{n_k} \ \forall k \end{cases}$$

with $x \in \mathbb{R}^n$, $I_k \subseteq \{1, \cdots, n\}$, $\#I_k = n_k + 1$

Convex Conic Program, includes

- ◇ Linear optimization
- ◇ (Convex) Quadratic optimization
- ◇ (Convex) Quadratically constrained quadratic optimization

Many applications (robust LO, engineering, portfolio, etc.)

Observation

SOCO software $\ll$ LO software w.r.t. capability (size) and availability (commercial packages)

Idea: try to *approximate* SOCO with LO, i.e. approximate $\mathbb{L}^n$ with a polyhedral cone Π^n

Accuracy: Π^n is an α-approximation of $\mathbb{L}^n$ iff

$$\{(x_0, x) \mid \|x\| \leq x_0\} \subseteq \Pi^n \subseteq \{(x_0, x) \mid \|x\| \leq \alpha x_0\}$$

Goal: Size of approximation must not increase too much when n increases or $\alpha \to 1$

Direct approach: Circumscribed polyhedral cone $\Rightarrow$ number of facets exponential in n

Ben-Tal & Nemirovski:

- ⋄ Replace a $\mathbb{L}^n$ cone by $(n-1)$ $\mathbb{L}^2$ cones (introducing additional variables)
- ⋄ Approximate each $\mathbb{L}^2$ cone with the *projection* of a polyhedral cone

Reference: *On polyhedral approximations of the second-order cone*, A. Ben-Tal and A. Nemirovsky, technical report, Technion Israel Institute of Technology, 1998

Approximation of $\mathbb{L}^n$

Decomposition

Let n even. We have $x \in \mathbb{L}^n$ iff $\exists \alpha, \beta$ s.t.

$$(\alpha, x_A) \in \mathbb{L}^{\frac{n}{2}}, (\beta, x_B) \in \mathbb{L}^{\frac{n}{2}}, (x_0, \alpha, \beta) \in \mathbb{L}^2$$

with $A = \{1, \ldots, \frac{n}{2}\}$ and $B = \{\frac{n}{2}, \ldots, n\}$
(due to $\|x\|^2 = \|x_A\|^2 + \|x_B\|^2 \leq \alpha^2 + \beta^2 \leq x_0^2$)

Let $n = 2^m$. Proceeding *recursively*, we end up with $(n-1)$ $\mathbb{L}^2$ cones and $(n-2)$ additional variables (m generations of cones/variables). Easy generalization for any n.

If α_i-approximation is used for the $\mathbb{L}^2$ cones of i^{th} generation, resulting cone has $\alpha = \prod_{i=1}^m \alpha_i$

Approximation of $\mathbb{L}^2$

$x \in \mathbb{L}^2 \Leftrightarrow (\frac{x_1}{x_0}, \frac{x_2}{x_0}) \in B^2 = \big\{(x, y) \mid x^2 + y^2 \leq 1\big\}$

Approximate the unit disc B^2 in $\mathbb{R}^2$ and *homogenize* the resulting polytope to make an approximating polyhedral cone.

First idea: regular polygon with q inequalities

Gives quality $\alpha = \cos(\frac{\pi}{q})^{-1} \approx 1 + \frac{\pi^2}{2q^2}$

$\Rightarrow$ too expensive ($q > 2000$ for $\alpha = 1 + 10^{-6}$)

Better idea: use additional variables and project i.e.

$$x \in \Pi^2 \Leftrightarrow \exists u \in \mathbb{R}^m \mid (x, u) \in P$$

where P is a polytope in $\mathbb{R}^{3+m}$

Construction of P

Consider the following system

$$\left\{ \begin{array}{rcl} \alpha_{i+1} & = & \alpha_i \cos \frac{\pi}{2^i} + \beta_i \sin \frac{\pi}{2^i} \\ \beta_{i+1} & = & \left| -\alpha_i \sin \frac{\pi}{2^i} + \beta_i \cos \frac{\pi}{2^i} \right| \end{array} \right. , \; 0 \leq i < q$$

$$\left\{ \begin{array}{rcl} \beta_q & \leq & 2 \sin \frac{\pi}{2^q} \\ 1 & = & \alpha_q \cos \frac{\pi}{2^q} + \beta_q \sin \frac{\pi}{2^q} \end{array} \right.$$

where $q \geq 1$ is a positive parameter.

Its projection on (α_0, β_0) is the boundary of a regular 2^q-gon

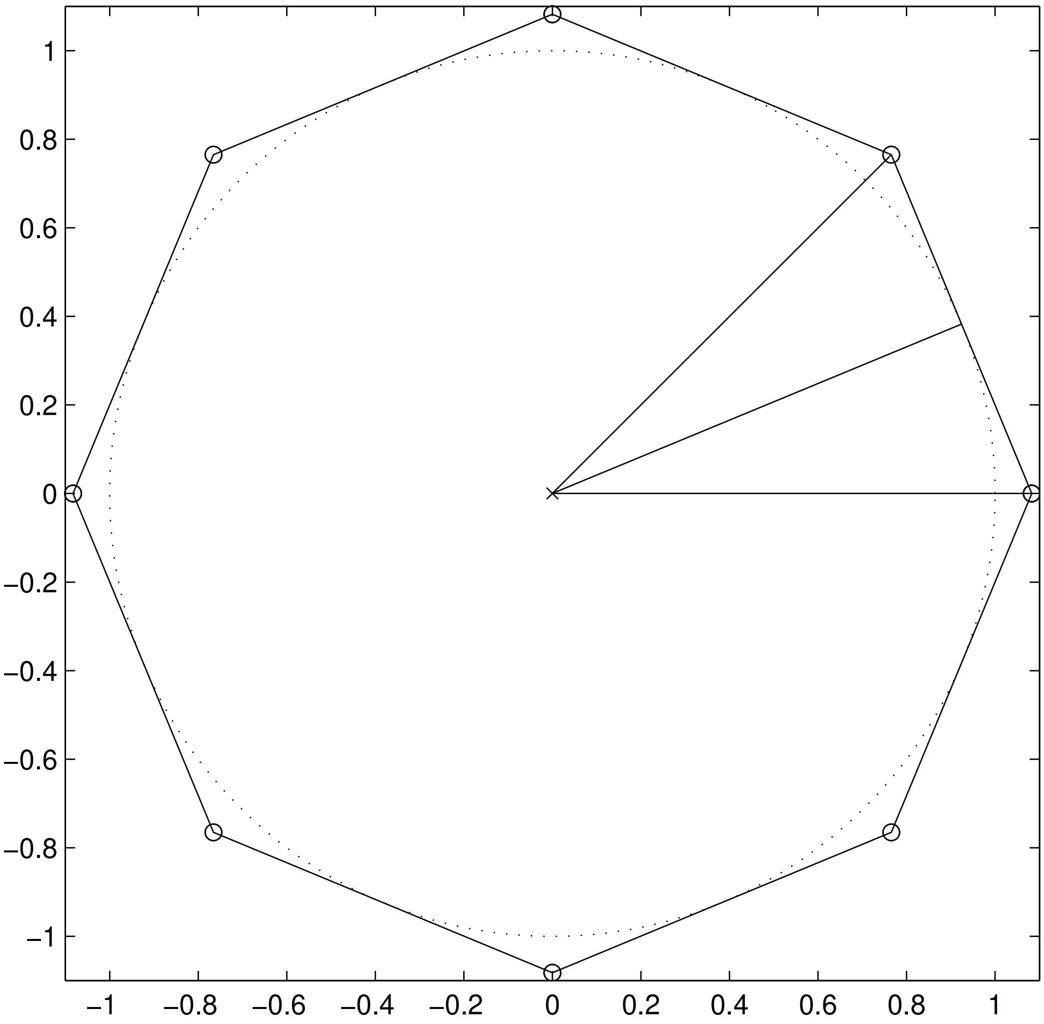


Figure for $q = 3$

Take the convex hull and homogenize to get P_q

$(a = |b| \quad \xrightarrow{\text{convex hull}} \quad a \geq |b|)$

Accuracy: $\alpha = \left(\cos \frac{\pi}{2^q}\right)^{-1} \approx 1 + \frac{\pi^2}{2^{2q+1}}$

(e.g. $\alpha = 1 + 10^{-6}$ with $q = 12$)

Size: $2q + 1$ inequalities, $2q$ additional variables (small improvement over original construction)

Thus $P_q := \{\ (x_0, \alpha_0, \beta_0, \dots, \alpha_q, \beta_q)\ \}$ s.t.

$$\left\{\begin{array}{rcl} \alpha_{i+1} & = & \alpha_i \cos \frac{\pi}{2^i} + \beta_i \sin \frac{\pi}{2^i} \\ \beta_{i+1} & \geq & \left| \alpha_i \sin \frac{\pi}{2^i} - \beta_i \cos \frac{\pi}{2^i} \right| \end{array}\right. \text{ for } i = 0, \dots q$$

$$\left\{\begin{array}{rcl} \beta_q & \leq & 2 \sin \frac{\pi}{2^q} \\ x_0 & = & \alpha_q \cos \frac{\pi}{2^q} + \beta_q \sin \frac{\pi}{2^q} \end{array}\right.$$

where α_0 and β_0 are identified to x_1 and x_2

Combining the cones

For each generation, choose quality q_i such that

$$\prod_{i=1}^{m} \left(\cos \frac{\pi}{2^{q_i}} \right)^{-1} \leq \alpha$$

while minimizing total size $\to P_\alpha^n$; leads to total number of inequalities and additional variables bounded by $O(1) n \log \frac{2}{\alpha - 1}$ (optimal)

Finally SOCO is approximated by

$$\min c^T x \quad \text{s.t.} \quad \left\{\begin{array}{l} b^l \leq Ax \leq b^u,\ x^l \leq x \leq x^u, \\ (x_{I_k}, u_{J_k}) \in P_\alpha^n\ \forall k \end{array}\right.$$

Relaxation arbitrarily close, but guarantee ?

Implementation

- ◇ MATLAB 5 on a 233 MHz Pentium PC
- ◇ LO and CQO problems solved with the interior-point MOSEK optimizer (E. Andersen)
- ◇ SOCO problems solved with the interior-point SeDuMi optimizer (J. Sturm)

We implemented the following procedures

- ◇ `ApxLor2(q)`: generates a polyhedral approximation of $\mathbb{L}^2$ with `q` rotation steps
- ◇ `ApxLorN(n, a)`: generates a polyhedral approximation of $\mathbb{L}^{\mathtt{n}}$ with quality `a`. Recursive, uses `ApxLor2` and a cache of already computed approximations
- ◇ `ApxLorLP(soc, a)`: generates an linear approximation of the problem `soc` with quality `a`. Each second-order constraint is replaced by its linear approximation given by `ApxLorLP`

All q_i's chosen equal $\Rightarrow q = \left\lceil \log_2 \frac{\pi}{\arccos \alpha^{-1/m}} \right\rceil$

with total size

$$O(1) n \log \frac{\log_2 n}{\alpha - 1}$$

Test problems

Multi-load truss topology design

$$\min \sum_{i=1}^{n} \sigma_i \text{ s.t. } \begin{cases} \|(q_{i1}, \dots, q_{ik})\|_2 \leq \sigma_i,\ 1 \leq i \leq n \\ \sum_{i=1}^{n} q_{il} b_i = f_l,\ 1 \leq l \leq k \end{cases}$$

n = number of bars, k = number of loads, norm minimization problem

Dual can be simplified $\Rightarrow$ QCO problem

α-approximation guarantees relative error ϵ on approximated objective such that $\epsilon \leq \frac{\alpha-1}{\alpha}$

$(p^* \geq p^*(\alpha) \geq \frac{\sum_{i=1}^{n} \|q_i^*\|_2}{\alpha} \geq \frac{p^*}{\alpha})$

Computational results

Using random generator by A. Nemirovski

Comparison between

- ◇ Solving the dual using QCO
- ◇ Solving the primal using SOCO up to ϵ accuracy
- ◇ Solving the approximated primal using LO up to ϵ accuracy

Solving the QCO dual always faster

$\rightarrow$ CPU times relative to QCO dual reported

Problem $p - q - k$ is with $p \times q$ nodal grid ($\approx \frac{1}{2}(pq)^2$ bars) and k loading scenarios

Observations

- ◇ Many small second-order cones
- ◇ Expected accuracy attained, often more than required

Observations

- ◇ Approximated primal slower, especially for larger cones
- ◇ Difference between SOCO primal and QCO dual less important for larger problems
- ◇ Increase of CPU time for smaller ϵ larger for LO than for SOCO

	$\epsilon = 10^{-1}$		$\epsilon = 10^{-4}$		$\epsilon = 10^{-8}$	
Prob.	SO	LO	SO	LO	SO	LO
2-2-4	3.0	2.2	4.0	4.6	5.4	8.0
2-2-8	2.2	3.7	3.0	7.0	4.4	14.6
2-2-16	1.6	6.4	2.2	14.4	3.3	29.4
2-2-32	2.3	13.2	3.1	29.6	4.9	60.7
4-4-3	1.6	7.5	2.3	21.5	2.8	42.4
4-4-4	1.7	18.5	3.0	56.2	4.0	130.6
6-6-4	1.6	12.7	2.0	56.5	2.7	124.3
8-8-2	1.2	4.7	1.6	19.2	1.9	38.2

Convex quadratic problems

$$\min \frac{1}{2} x^T Q x + c^T x + c_0 \quad \text{s.t.} \quad \begin{cases} l_c \leq Ax \leq u_c \\ l_x \leq x \leq u_x \end{cases}$$

where Q is p.s.d.

SOCO model: Let $L^T L = Q$ and $y \in \mathbb{R}^n$, $\alpha, \beta \in \mathbb{R}$ additional variables

$$\min \frac{\alpha + \beta}{2} + c^T x + c_0 \quad \text{s.t.} \quad \begin{cases} l_c \leq Ax \leq u_c \\ l_x \leq x \leq u_x \\ y = Lx \\ \beta = \alpha + 1 \\ (\beta, y, \alpha) \in \mathbb{L}^{n+1} \end{cases}$$

Equivalence because $(\beta, y, \alpha) \in \mathbb{L}^{n+1} \Leftrightarrow \|y\|^2 + \alpha^2 \leq \beta^2 \Leftrightarrow x^T L^T L x \leq \alpha + \beta$

Computational results

Using CQO test set (Maros, Mészáros)

	Size	QP	Approximated QP		
Problem	m/n		10^{-1}	10^{-4}	10^{-8}
`cvxqp1_s`	50/100	.16	2.8	9.7	20.2
`dual1`	1/85	.27	2.64	6.4	10.8
`qstandat`	359/1075	.38	3.52	16.7	50.4
`dualc1`	215/9	.16	.33	.55	.82
`primal1`	85/325	.33	10.5	34.6	66.7

Observation

Approximation slower, even for $\epsilon = 10^{-1}$

But: For problem `dualc1`, actual relative errors of approximated problems are $9.1\ 10^{-2}$, $1.4\ 10^{-4}$ and $1.7\ 10^{-7}$!

Accuracy analysis

To model $\|x\|^2 \leq t$ we use

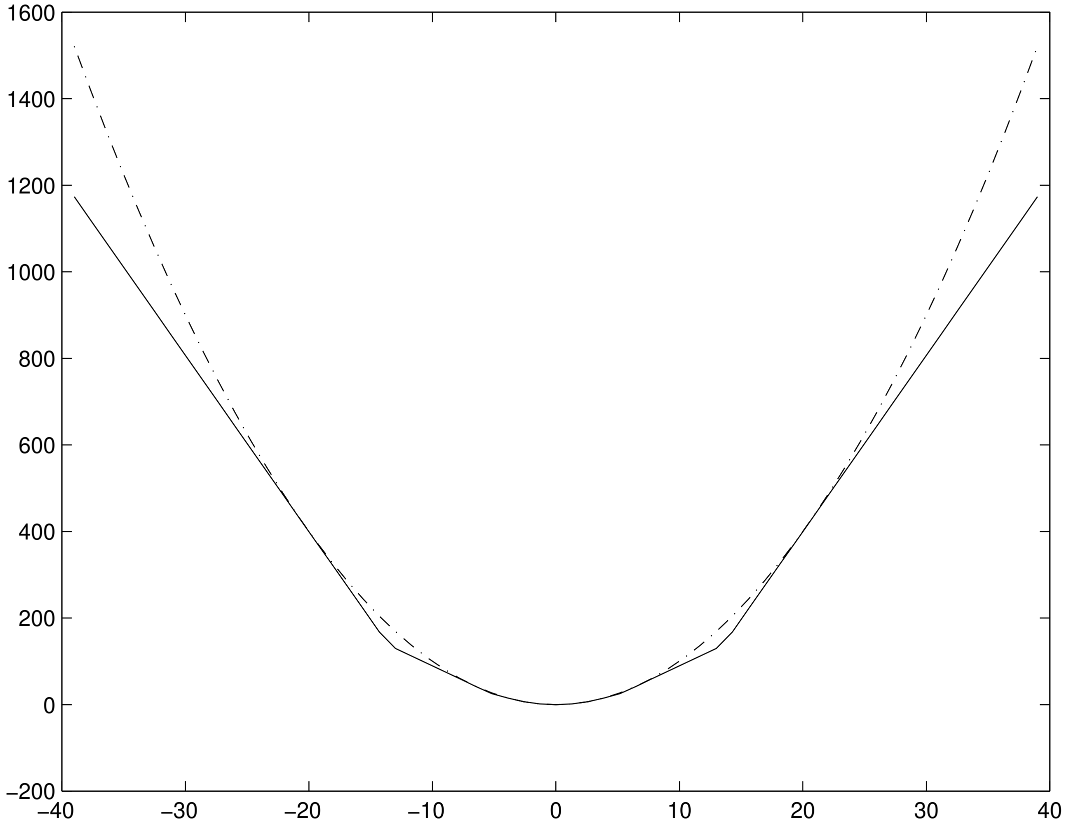
$$\|x\|^2 \leq t \Leftrightarrow (\frac{t+1}{2}, x, \frac{t-1}{2}) \in \mathbb{L}^{n+1}$$

But

$$\begin{aligned} & (\frac{t+1}{2}, x, \frac{t-1}{2}) \in \Pi^{n+1} \\ \Leftrightarrow \quad & x^2 + \frac{(t-1)^2}{4} \leq \alpha^2 \frac{(t+1)^2}{4} \\ \Leftrightarrow \quad & \|x\|^2 \leq t + (\alpha^2 - 1)\frac{(t+1)^2}{4} \end{aligned}$$

The ratio $\frac{\|x\|^2}{t}$ can be arbitrarily large

$$\frac{\|x\|^2}{t} \leq 1 + (\alpha^2 - 1)\frac{(t+1)^2}{4t} \geq \alpha^2$$



Workaround for QP

- $\diamond$ For pure QP use $x^T Q x = \|Lx\|^2 \leq t^2$
- $\diamond$ When Q is p.d. use
  $\frac{1}{2} x^T Q x + c^T x + c_0 = \|Lx + l\|^2 + C$
- $\diamond$ When Q is p.s.d. ?

Concluding remarks

- ◇ Clever scheme to approximate $\mathbb{L}^n$ with a polyhedral cone
- ◇ Computationally less efficient than dedicated solvers
- ◇ Same TTD problem 7-8-5 with CPLEX takes 100 sec. with barrier and 8200 sec. with primal simplex
- ◇ Cannot always guarantee maximum relative error for the objective in the approximated problem