Modeling Cone Optimization Problems with COIN OS

Imre Pólik

Joint work with Gus Gassman, Jun Ma, Kipp Martin

Lehigh University

Department of ISE

INFORMS 2009, San Diego

October 12, 2009

Outline

1. Problem description
   - Cone optimization
   - Semidefinite optimization
   - Special problems
2. Problem instance representation
   - Existing formats
   - Problem layout
   - How to best represent the problems?
3. Loose XML specification
   - Design philosophy
   - Declarations
   - Data, functions
4. What's next

General cone optimization

$$\min c^T x \qquad \max b^T y$$
$$Ax = b \qquad A^T y + s = c$$
$$x \in \mathcal{K} \qquad s \in \mathcal{K}^*$$

The cone $\mathcal{K}$ can be

Linear: $x \geq 0$

Second-order: $x_0 \geq \|x\|_2$

Rotated second-order: $x_0 x_1 \geq \|x_{2:n}\|$, and $x_0 \geq 0$

Semidefinite: x is (can be assembled into) a symmetric, positive semidefinite matrix, or a

product/intersection of these.

robust control, combinatorics, polynomial and SOS, truss-topology, materials structure, ...

Semidefinite optimization

- Standard form

$$\begin{aligned} \min\ & C \bullet X & \qquad \max\ & b^T y \\ \mathcal{A}X &= b & \mathcal{A}^* y + S &= C \\ X &\succeq 0 & S &\succeq 0, \end{aligned} \tag{P-D}$$

where $b, y \in \mathbb{R}^m$, $X, S, C \in \mathbb{R}^{n^2}$, $\mathcal{A} : \mathbb{R}^{n^2} \to \mathbb{R}^m$

- Linear operator $\mathcal{A}$

$$\mathcal{A}X = (A_i \bullet X)_{i=1}^m$$

$$\mathcal{A}^* y = \sum_{i=1}^m A_i y_i$$

$\Rightarrow$ too restrictive

Special forms

- Rank one, low rank A_i

$$A_i = aa^T,\ A_i \bullet X = a^T X a$$

  - can be exploited inside the IPM
  - cannot be recovered exactly from A_i
- General operators

$$\mathcal{A}X = AX + XA, \text{or}$$
$$\mathcal{A}X = AXB + BXA$$

  - $\mathcal{A}$ is a large Kronecker product
  - huge savings in storage and computation
  - one needs to have $\mathcal{A}^*$
- Cone intersections

Input formats

- What's out there
  - SDP: SeDuMi, SDPT3, SDPpack, PENSDP, Sparse SDPA, extensions
  - SOCP: MOSEK, LOQO, CPLEX
  - CVX, Yalmip
  - COIN-OS (first attempt)
- Common features
  - based on the standard problem form
  - not flexible
  - hard to extend

Problem layout

$x_{1:2}$	$x_{3:7} \leq 0$	$\mathrm{mat}(x_{8:16}) \succeq 0$

$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$		
$\mathcal{A}_{11}$	$\mathcal{A}_{12}$	$\mathcal{A}_{13}$	$=$	3
$\mathcal{A}_{21}$	$\mathcal{A}_{22}$	$\mathcal{A}_{23}$	$\preceq$	I
$\mathcal{A}_{31}$	$\mathcal{A}_{32}$	$\mathcal{A}_{33}$	$\geq$	0

- Declare variables and constraints
- Define the $\mathcal{C}_j, \mathcal{A}_{ij}$ mappings and the RHS
- Very similar to LP
- The basic unit is different

A collection of cone optimization problems

- Problems/problem structures from
  - robust optimization
  - combinatorics
  - stability and control
  - polynomial optimization
  - ...
- Necessary language components
  - $a^T X a$
  - $\mathrm{Tr}(X)$
  - $\det(X)$
  - $AXB + BXA$
  - X^{-1}
  - ...
- Collection to be published later
  - Joint work with Johan Löfberg and Michael C. Grant

Current COIN OS conic constructs

- LP + cone constraints
  - (our fault)
  - very inefficient
  - all the drawbacks of existing formats
  - does not allow advanced operators
- Use matrix variables instead
  - smallest unit
  - further subdivision is artificial
- Use functions of matrices
  - extend the OSnL library
- Goal: preprocessing

Declarations

- Matrix variable
  - from new/existing scalar variables
  - verification is done here
  - matrices can share variables
- Attributes
  - symmetric,
  - positive semidefinite
  - Hermitian
  - integer (MICLP!)
  - matrix size
  - bounds (interpreted according to the matrix type)
- Matrix parameters
  - to be used in new functions
  - $\det(M + X)$

Functions

- Create a library of matrix functions
  - $\det(X)$
  - AX
  - $AXB + BXA$
  - $\lambda_{\min}(X)$
  - $\ldots$
- The arguments are matrices, not n^2 numbers!
- Verification is easier
- Extends the OSnL library

Conclusions

- We have a ...
  - collection of various cone problems
  - list of constructs needed
  - loose syntax specification
- We need an ...
  - exact syntax, documentation
  - implementation into COIN OS
  - XML-parser
  - example library
  - extensible preprocessing library
  - OSsL extension