Modeling Cone Optimization Problems (and more!) with COIN OS

Imre Pólik
Joint work with Gus Gassmann, Jun Ma, Kipp Martin

Lehigh University
Department of ISE

February 19, 2010
Outline

1. Problem description
2. Motivation
3. Problem instance representation
4. COIN OS
5. Examples
6. Where are we?
General cone optimization

\[
\begin{align*}
\min & \ c^T x \\
\text{subject to} & \ Ax = b \\
x & \in \mathcal{K}
\end{align*}
\]

\[
\begin{align*}
\max & \ b^T y \\
\text{subject to} & \ A^T y + s = c \\
s & \in \mathcal{K}^*
\end{align*}
\]

The cone \( \mathcal{K} \) can be

Linear: \( x \geq 0 \)

Second-order: \( x_0 \geq \|x\|_2 \)

Rotated second-order: \( x_0 x_1 \geq \|x_{2:n}\|^2, \text{ and } x_0 \geq 0 \)

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, \ldots
Semidefinite optimization

- Standard form

\[
\begin{align*}
\min \ & C \bullet X \\
\text{subject to} \ & AX = b \quad X \text{ is PSD}
\end{align*}
\]

\[
\begin{align*}
\max \ & b^T y \\
\text{subject to} \ & A^* y + S = C \quad S \text{ is PSD},
\end{align*}
\]

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

- Linear operator \( A \)

\[
AX = (A_i \bullet X)_{i=1}^m
\]

\[
A^* y = \sum_{i=1}^m A_i y_i
\]

\( \Rightarrow \text{too restrictive} \)
Motivation

- Need for a general format
- Express the special structure of the problem
- Preprocessing
- COIN-OR would need it anyways
Special forms

- Rank one, low rank $A_i$
  \[ A_i = a a^T, \quad A_i \bullet X = a^T X a \]
  - immediate savings in storage
  - can be exploited inside the algorithm
  - cannot be recovered exactly from $A_i$

- General operators
  \[ AX = AX + XA, \text{ or } \]
  \[ AX = AXB + BXA \]
  - $A = A \otimes A$ is a large Kronecker product
  - not practical for $n > 100$
  - huge savings in storage and computation
Input formats

- What’s out there
  - SDP: SeDuMi, SDPT3, SDPpack, PENSDDP, 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
A collection of cone optimization problems

- Problems/problem structures from
  - robust optimization
  - combinatorics
  - stability and control
  - polynomial optimization
  - ...

- Necessary language components
  - $a^T X a$
  - $\text{Tr}(X)$
  - $\det(X)$
  - $AXB + BXA$
  - $X^{-1}$
  - ...

- Collection to be published soon
  - Joint work with Johan Löfberg and Michael C. Grant
The COIN OR project

- Started in 2000 by IBM
- COnputational INfrastructure for Operations Research
- Open-source repository of OR related software
  - optimization
  - algorithmic differentiation
  - graph algorithms
- Transferred to a nonprofit organization
COIN Optimization Services

- Standards to represent
  - optimization problems
  - results
  - communication between clients and solvers

- Implemented by most COIN OR solvers

- Based on XML schemas
  - portability
  - web services
Original COIN OS conic constructs

- LP + cone constraints
  - (our fault)
  - very inefficient
  - all the drawbacks of existing formats
  - did 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
  - $\text{det}(X)$
  - $AX$
  - $AXB + BXA$
  - $\lambda_{\text{min}}(X)$
  - ... 

- The arguments are matrices, not $n^2$ numbers!
- Verification is easier
- Extends the OSnL library
A small SDP

\[
\begin{align*}
\min & \quad 10x_1 + 20x_2 \\
\text{s.t.} & \quad x_1 F_1 + x_2 F_2 - F_0 \text{ is positive semidefinite,}
\end{align*}
\]

where

\[
F_0 = \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 \\
0 & 0 & 3 & 0 \\
0 & 0 & 0 & 4 \\
\end{pmatrix}, \quad F_1 = \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{pmatrix}, \quad F_2 = \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 5 & 2 \\
0 & 0 & 2 & 6 \\
\end{pmatrix}
\]

Notice the $2 \times 2$ block structure.
Examples – Variables and objective

```
<variables numberOfVariables="2">
  <var lb="-INF" ub="INF" mult="2"></var>
</variables>

<objectives>
  <obj maxOrMin="min" numberOfObjCoef="2">
    <coef idx="0">10.</coef>
    <coef idx="1">20.</coef>
  </obj>
</objectives>
```
Examples – Constant matrices

<constantMatrix id="F2:2" numberOfColumns="2" numberOfRows="2">
  <elements numberOfValues="3">
    <start>
      <el>0</el>
      <el>1</el>
      <el>3</el>
    </start>
    <rowIdx>
      <el mult="2">0</el>
      <el>1</el>
    </rowIdx>
    <value>
      <el>5.</el>
      <el>2.</el>
      <el>6.</el>
    </value>
  </elements>
</constantMatrix>
Examples – Cones and constraints

<cones numberOfCones="2">
  <semidefiniteCone id="C1"
    numberOfColumns="2" numberOfRows="2"/>
  <semidefiniteCone id="C2"
    numberOfColumns="2" numberOfRows="2"/>
</cones>

Also available: (rotated) Lorentz, copositive, completely positive, nonnegative, product, intersection

<matrixConstraints numberOfMatrixCon="2">
  <matrixCon numberOfRows="1" numberOfColumns="1"
    lbMatrixID="F01" lbConeId="C1"/>
  <matrixCon numberOfRows="1" numberOfColumns="1"
    lbMatrixID="F02" lbConeId="C2"/>
</matrixConstraints>
Examples – Coefficients

<linearConstraintMatrixOperators numberOfOperators="3">

(term $x_1 F_{11}$ in the first constraint)

<scalarVarOperator varIdx="0" matrixConIdx="0" matrixID="F1:1"/>

(term $x_2 F_{21}$ in the first constraint)

<scalarVarOperator varIdx="1" matrixConIdx="0" matrixID="F2:1"/>

(term $x_2 F_{22}$ in the second constraint)

<scalarVarOperator varIdx="1" matrixConIdx="1" matrixID="F2:2"/>

</linearConstraintMatrixOperators>
Advanced constructions – Matrix programming

- General linear operators
  \[ 0.5 M_5 X_0 M_6^T \] in the first constraint

  ```xml
  <operator matrixConIdx="0" matrixVarIdx="0"
    scalarCoef="0.5" leftMatrixID="M5" rightMatrixID="M6"
    rightMatrixTranspose="true"/>
  ```

- Nonlinear functions
  \[ X_3^{-1} \] in the second constraint

  ```xml
  <nonlinearMatrixExpressions
    numberOfMatrixNonlinearExpressions="1">
    <matrixNL matrixConIdx="1">
      <matrixInverse>
        <matrix id="3"/>
      </matrixInverse>
    </matrixNL>
  </nonlinearMatrixExpressions>
  ```

- We don’t even need cones!
Summary

- **Completed**
  - collection of various cone problems
  - list of constructs needed
  - XML schemas in COIN OS
  - converter from SDPA format (others coming soon)
  - initial verification

- **In progress**
  - cosmetic changes
  - more examples
  - conversion from other formats (SDPpack, SeDuMi, SDPT3)
Future work

- Example library
  - SOCP
  - mixed integer problems

- Solver links
  - CSDP (already in COIN OR)
  - SeDuMi

- Extensive preprocessing routines
  - decomposition
  - matrix completion
  - sparsity