Conic optimization in MOSEK: Present and the future.

Erling D. Andersen
MOSEK APS
C/O Symbion Science Park
Fruebjergvej 3, Boks 16
2100 Copenhagen Ø
Company WWW: http://www.mosek.com
Email: e.d.andersen@mosek.com
WWW: http://erling.andersen.name

May 16, 2002

Outline

1. Introduction.

2. Nonlinear optimization in MOSEK.

3. Conic optimization in MOSEK.

4. Some computational results.

5. Conclusions.

Introduction to MOSEK

- An optimization software package.
- Linear, quadratic, and **convex nonlinear (functional)** optimization.
- **Conic quadratic optimization.**
- Mainly interior-point based.
- Primal simplex and BI.
- Parallel processor support.
- C/MPS/MATLAB/AMPL/GAMS interfaces.
- Windows/Linux/Solaris platforms.
- Mixed-integer capabilities for the linear and quadratic case.

Nonlinear optimization in MOSEK

$$\begin{array}{llll} (NO) & \text{min} & f(x) & \\ & \text{s.t.} & g(x) & \leq 0. \end{array}$$

Restrictions:

- Twice differentiable.
- Convexity.

Practical problems:

- Convexity check is difficult.
- Black box model.
  - Presolve is difficult.
  - Automatic scaling is difficult.

- Require use of a modelling language.
  - Coding Gradients and Hessians are cumbersome.
  - Modelling languages are slow.
  - Difficult to use in systems.
- Dual is difficult to form.
  - Dual may be nonconvex.
  - Sometimes better to solve dual.
- Possibilities for errors are large.
- Algorithmic problems too.

The NLO algorithm in MOSEK

- Based on homogeneous model for MCPs of Andersen and Ye.
- Roughly an interior point method applied to KKT.

Problem:

$$
\begin{array}{lrcl}
\text{min} & c(x) & & \\
\text{s.t.} & a_i(x) & \geq & 0, \ \ i = 1, \ldots, m.
\end{array}
$$

Lagrange function:

$$L(x, y) := c(x) - y^T a(x).$$

AY homogeneous model:

$$\begin{array}{rcl}
\tau \nabla_x L(x/\tau, y/\tau)^T & = & 0, \\
\tau a(x/\tau) & = & z, \\
-x^T \nabla_x L(x/\tau, y/\tau)^T - y^T a(x/\tau) & = & \kappa, \\
Zy & = & 0, \\
\tau\kappa & = & 0, \\
z, \tau, y, \kappa \geq 0. & &
\end{array}$$

Algorithmic idea:

- Apply Newton's method to AY model.

Observe:

- Nonlinear constraints.
  - Dual constraints are nonconvex.
- Merit function is required.
- What is the right barrier function?

Comments:

- Works well for small to medium sized problem.
  - Iteration count grows not-so-slowly with problem size.
- Sensitive to:
  - Parameter choice.
  - Merit function choice (impossible to choose).

Function based optimization

My conclusions are:

- Bad for the user.
  - Difficult to use.
- Bad for the software developer:
  - Difficult to handle the nonlinear information.
- Bad for the algorithm developer.
  - No structure to work with.
- Sum: The **black box model** functional model is bad.
- What is the alternative?

Conic optimization in MOSEK

$$
\begin{array}{llrcl}
(PCO) & \text{min} & c^T x & & \\
 & \text{s.t.} & Ax & = & b, \\
 & & x \in \mathcal{K}. & &
\end{array}
$$

where $\mathcal{K}$ is a convex cone (closed, pointed and solid).

- $\mathcal{K}$ is convex.

- Cone condition:

$$x \in \mathcal{K} \Rightarrow \alpha x \in \mathcal{K}, \forall \alpha \geq 0.$$

- Pointed:

$$\mathcal{K} \cap -\mathcal{K} = 0.$$

- Solid:

$$int \mathcal{K} \neq \emptyset.$$

Dual problem:

$$\begin{array}{llrcl} (DCO) & \max & b^T y & & \\ & \text{s.t.} & A^T y + s & = & c, \\ & & s \in \mathcal{K}^*, & & \end{array}$$

where

$$\mathcal{K}^* := \{s : \ x^T s \geq 0, \ \forall x \in \mathcal{K}\}.$$

Comments:

- Easy to form dual.

- Dual is convex.

Goal for conic optimization in MOSEK:

- Cone can be composed of several **predefined** cone types i.e.

$$\mathcal{K} = \mathcal{K}_1 \times \ldots \times \mathcal{K}_k$$

- Each cone $\mathcal{K}_i$ has a known form.

- Advantages:
  - Structure is well-defined.
  - No black box functions.
  - Easy to form dual.
  - Easy software wise. (Which variable are member of which cones).

Comments:

- (Restricted) conic optimization is promising.

- Nesterov and Nemirovski introduced the idea (?).

- But which cone types?
  - Symmetric.
  - Nonsymmetric.

Symmetric cones

- Linear

$$\mathcal{K}_l := \{x \in R : \; x \geq 0\}.$$

- Quadratic:

$$\mathcal{K}_q := \left\{ x \in R^n : \; x_1 \geq \sqrt{\sum_{j=2}^{n} x_j^2} \right\}.$$

- Semi-definite:

$$\mathcal{K}_s := \left\{ x \in R^{\frac{n(n+1)}{2}} : \; \begin{bmatrix} x_1 & \cdots & x_n \\ \vdots & \ddots & \vdots \\ x_n & \cdots & x_{\frac{n(n+1)}{2}} \end{bmatrix} \succeq 0 \right\}.$$

Let

$$X := \begin{bmatrix} x_1 & \cdots & x_n \\ \vdots & \ddots & \vdots \\ x_n & \cdots & x_{\frac{n(n+1)}{2}} \end{bmatrix}$$

then $\succeq$ means **symmetry**

$$X = X^T.$$

And **positive semi-definiteness**:

$$y^T X y \geq 0, \forall y$$

or equivalently

$$\lambda_{\text{min}}(X) \geq 0.$$

Comment:

- Nesterov and Todd presents efficient interior point primal-dual alg.

Nonsymmetric cones

MOSEK is used for:

- Geometric optimization (Duffin, Peterson, and Zener).
- Entropy optimization ($x \ln(x)$).
- Other convex problems.
  - All separable.
  - May be large but has simple structure.

What are the right cones:

- Glineur (ph.d thesis).
  - p-cone.
  - Geometric cone.
  - Separable cone:

$$t \geq \sum_j \tau g_j(x_j/\tau).$$

- Zhang develops general conic framework:

$$t \geq \tau f(x/\tau).$$

Following Glineur:

- Primal geometric cone

$$\left\{(\theta,\tau,x) : \tau\sum_j e^{-\frac{x_j}{\tau}} \le \theta\right\}$$

and dual

$$\left\{(\eta,\kappa,s) : \quad \sum_{j:s_j>0}\left(s_j \ln\left(\frac{s_j}{\eta}\right) - s_j\right) \le \kappa,\ \ \eta \ge 0\right\}.$$

- Primal log cone:

$$\left\{(\theta,\tau,x) :\ \ \tau\sum_j\left(-\frac{1}{2} - \ln(\frac{x_j}{\tau})\right) \le \theta,\ \ x,\tau \ge 0\right\}$$

and the dual cone

$$\left\{(\eta,\kappa,s) :\ \ \eta\sum_j\left(-\frac{1}{2} - \ln(\frac{-s_j}{\eta})\right) \le \kappa,\ \ s \le 0,\ \ \eta \ge 0\right\}$$

- Primal p-norm cone ($p \geq 1$):

$$\left\{(\theta, \tau, x) : \sum_j |x_j|^p \leq p\theta\tau^{p-1},\ \tau \geq 0\right\}$$

and the dual cone (1/p+1/q=1):

$$\left\{(\eta, \kappa, s) : \sum_j |s_j|^q \leq q\eta\kappa^{q-1},\ \kappa \geq 0\right\}.$$

Comments:

- 3 symmetric cones.
- 3(6) nonsymmetric cones.
- Possibly a few more relevant cones.
- Powerful set of cones.
- Algorithms.
  - Primal-dual algorithms for nonsymmetric cones are lacking.
    - Right scaling?
  - Tuncel has some ideas.
  - Zhang primal-dual algs. does not use symmetric scaling.

Presentation of conic optimization in MOSEK

- Supports quadratic cones only.
- Uses NT based primal-dual algorithm.
  - Primal and dual feasibility eq. linear.
  - Merit function is easy.
- Homogeneous model.
- Solves large-scale sparse case.
- C/MATLAB/GAMS interfaces.

Experiences:

- Robust.

- Slower iteration growth than convex solver.

- Leads to better formulated models e.g.

$$||x||^2 \leq M$$

  is modelled as

$$||x|| \leq t, \;\; t = \sqrt{M}.$$

- Accurate search direction computation is important.

  - Dense columns and big cones can cause problems.

Future of conic optimization in MOSEK

- Support for semi-definite cone.

- Support for non-symmetric cones.

  – Require primal-dual alg.

- Modelling language support for conic optimization.

Modelling language support for conic optimization

Issue:

- How to extend existing modelling languages to support conic optimization.
  - AMPL/GAMS/AIMMS.
- Presentation of a hack implemented in GAMS. Note
$$\frac{1}{x} \leq t$$
is identical to
$$v^2 \leq 2xt, \ v = \sqrt{2}.$$

```
Title Minimizing Total Average Cycle Stock (STOCKCC,SEQ=225)
$ontext
Minimizing Total Average Cycle Stock
$offtext


set nn items  /n1*n48/
    mm reorder intervals /i1*i9/;

Scalar N max total number of replenishments /100/;

Parameter Y(mm) possible number of orders /
     i1   3, i2   6, i3   9, i4  12, i5  18,
     i6  36, i7  52, i8  78, i9 156 /;

Parameter Dv (nn) demand rate times unit cost of item nn /
n1       20.04,  n2       21.72,  n3       37.92,  n4       54.12
n5       61.80,  n6       81.24,  n7       94.20,  n8      119.40
n9      171.60, n10     208.80, n11     415.27, n12     470.23
n13     1212    , n14     1393.2 , n15     1496.4 , n16     1600
n17     1702.4 , n18     1714.5 , n19     1870.5 , n20     1977.8
n21     2647.12, n22     3143.82, n23     4173    , n24     4347.7
n25     4917    , n26     5048.3 , n27     5397.2 , n28     6692.4
n29     6837.6 , n30     8003.1 , n31     8449.5 , n32     9152
n33   13236.3 , n34   13970    , n35   15327.6 , n36   16368
n37   19765    , n38   20470.3 , n39   23182.2 , n40   25026
n41   31675.6 , n42   56734.2 , n43   69040.4 , n44   75192
n45   97066.5 , n46   99803.2 , n47  105984    , n48  106465

Variable x(nn)    number of orders per unit time
         z(nn,mm) discrete orders choices
         obj objective variable;
Binary variable z;

Equations  defobj
           capacity
```

```
           defx(nn)
           defsos(nn);

defobj.. obj =e= sum(nn, 1.5*Dv(nn)/x(nn));

capacity..   sum(nn, x(nn)) =l= 3*N;

defx(nn)..   sum(mm, z(nn,mm)*Y(mm)) =e= x(nn);

defsos(nn).. sum(mm, z(nn,mm)) =e= 1;

x.lo(nn) = Y('i1');
x.up(nn) = Y('i9');

model stock /all/;
solve stock miniminzing obj using minlp;
```

```
Title Minimizing Total Average Cycle Stock (STOCKCC,SEQ=225)
$ontext
Minimizing Total Average Cycle Stock
$offtext


set nn items  /n1*n48/
    mm reorder intervals /i1*i9/;

Scalar N max total number of replenishments /100/;

Parameter Y(mm) possible number of orders /
     i1   3, i2   6, i3   9, i4  12, i5  18,
     i6  36, i7  52, i8  78, i9 156 /;

Parameter Dv (nn) demand rate times unit cost of item nn /
n1      20.04,  n2      21.72,  n3      37.92,  n4      54.12
n5      61.80,  n6      81.24,  n7      94.20,  n8     119.40
n9     171.60, n10     208.80, n11     415.27, n12     470.23
n13    1212   , n14    1393.2 , n15    1496.4 , n16    1600
n17    1702.4 , n18    1714.5 , n19    1870.5 , n20    1977.8
n21    2647.12, n22    3143.82, n23    4173   , n24    4347.7
n25    4917   , n26    5048.3 , n27    5397.2 , n28    6692.4
n29    6837.6 , n30    8003.1 , n31    8449.5 , n32    9152
n33   13236.3 , n34   13970   , n35   15327.6 , n36   16368
n37   19765   , n38   20470.3 , n39   23182.2 , n40   25026
n41   31675.6 , n42   56734.2 , n43   69040.4 , n44   75192
n45   97066.5 , n46   99803.2 , n47  105984   , n48  106465

Variable x(nn)    number of orders per unit time
         z(nn,mm) discrete orders choices
         v(nn)
         t(nn)
         obj objective variable;
Binary variable z;
```

```
Equations  defobj
           capacity
           defx(nn)
           defsos(nn),
           conicf(nn)
           conic(nn);

defobj.. obj =e= sum(nn, 1.5*Dv(nn)*t(nn));

capacity..   sum(nn, x(nn)) =l= 3*N;

defx(nn)..   sum(mm, z(nn,mm)*Y(mm)) =e= x(nn);

defsos(nn).. sum(mm, z(nn,mm)) =e= 1;

conicf(nn).. v(nn) =e= 1;

conic(nn).. x(nn) + t(nn) =c= v(nn);

x.lo(nn) = Y('i1');
x.up(nn) = Y('i9');

model stock /all/;
solve stock miniminzing obj using minlp;
```

Comments:

- What this right notation?
- What about semi-definite optimization?

Conclusions

- Conic optimization is a promising framework for NLO.

- Questions:

  - What is the primal-dual algorithm for non-symmetric cones?

  - How should modelling languages be extended?