From linear to conic optimization

Erling D. Andersen
http://erling.andersen.name
http://www.mosek.com

September 15, 2009
Outline

• Introduction

• Conic optimization.

• Applications of conic optimization.

• Algorithms for conic optimization.

• Some computational results.

• Literature.

• Conclusions.
Introduction

The most successful OR model:

\[(PO) \quad \min \quad c^T x \quad \text{s.t.} \quad Ax = b, \quad x \geq 0.\]

Pros:

- Wide applicability.
- Efficient and robust solution algorithms.
- “Simple” \((c, A, b)\).
- Duality theory.
Cons:

- Only linear.
- $x^2, 1/x, \ln(x), \ldots$

Nonlinear optimization

\[(NO) \quad \min \ f(x) \quad \text{s.t.} \quad g(x) \leq 0.\]

Pros:

- Very general.
Cons:

- Lack of good algorithms.

- Local versus global optimums.

- Convexity (how to check).

- Black box model.

- How to compute gradients and Hessians.

- How to handle $f$ and $g$ in software.
Summary:

- A linear model is restrictive.
- The nonlinear model is too general
- Is there a good compromise?
Conic optimization

\[(CO) \quad \min \quad c^T x \quad \text{s.t.} \quad Ax = b, \quad x \in K,\]

where $K$ is a convex cone (closed, pointed and solid).

- $K$ is convex.

- Cone condition:
  \[x \in K \Rightarrow \alpha x \in K, \forall \alpha \geq 0.\]

- Pointed:
  \[K \cap -K = 0.\]

- Solid:
  \[\text{int}K \neq \emptyset.\]
Some basic cones:

- **Linear:**
  \[ \mathcal{K}_l := \{ x \in \mathbb{R} : x \geq 0 \} \]

- **Quadratic:**
  \[ \mathcal{K}_q := \left\{ x \in \mathbb{R}^n : x_1 \geq \sqrt{\sum_{j=2}^{n} x_j^2} \right\} \]

- **Semi-definite:**
  \[ \mathcal{K}_s := \left\{ x \in \mathbb{R}^{\frac{n(n+1)}{2}} : \begin{bmatrix} x_1 & \cdots & x_n \\ \vdots & \ddots & \vdots \\ x_n & \cdots & x_{\frac{n(n+1)}{2}} \end{bmatrix} \succeq 0 \right\} \]
Let

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

then \( \succeq \) means \textbf{symmetry}

\[X = X^T.\]

And \textbf{positive semi-definiteness}:

\[y^T X y \geq 0, \forall y\]

or equivalently

\[\lambda_{\text{min}}(X) \geq 0.\]
Cone composition

Assumption:

\[ \mathcal{K} = \mathcal{K}^1 \times \ldots \times \mathcal{K}^k \]

where \( \mathcal{K}^i \) is of one of the basic cone types.

Example:

\[ \{x_1 \geq 0\} \times \{x_2 \geq \|(x_3, x_4)\|\} \]

Comments:

- Are called symmetric or self-scaled cones. (=self-dual and homogeneous cones.)

- Other cones exists but are not symmetric. Open topic.
Conic vision

• Restricted set of cones \((\leq 10)\).

• Cones are simple and easy to specify.

• Convexity is not an issue.

• A lot of structure.

• Nonlinearity is explicit.

• Gradients and Hessians are not an issue.

• Powerful algorithms exists (theory).
Conic duality

The dual cone:

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

The dual problem:

\[
(CO_D) \quad \max \quad b^T y \\
\text{s.t.} \quad A^T y + s = c, \\
\quad s \in \mathcal{K}^*.
\]

- Most (but not all) of the duality relations holds.

- \( \mathcal{K}_l, \ \mathcal{K}_q, \ \mathcal{K}_s \) are self dual i.e.

\[ \mathcal{K} = \mathcal{K}^*. \]
Applications

Conic quadratic optimization

Define the rotated quadratic cone:

\[ \mathcal{K}_r := \left\{ x \in \mathbb{R}^n : \, 2x_1 x_2 \geq \sum_{j=3}^{n} x_j^2, \, x_1, x_2 \geq 0 \right\} \]

Let

\[ x_1 = \frac{u+v}{\sqrt{2}}, \quad x_2 = \frac{u-v}{\sqrt{2}}, \]

then

\[ 2x_1 x_2 \geq \sum_{j=3}^{n} x_j^2 \iff u \geq \sqrt{v^2 + \sum_{j=3}^{n} x_j^2} \]

so the quadratic and rotated quadratic cones are equivalent. (It is easy to verify \( v \geq 0 \)).
Quadratic optimization

\[
\begin{align*}
\min & \quad 0.5\|Q^0x\|^2 + c^Tx \\
\text{s.t.} & \quad 0.5\|Q^ix\|^2 + a_i:x \leq b_i, \forall i = 1, 2, \ldots.
\end{align*}
\]

Conic quadratic equivalent:

\[
\begin{align*}
\min & \quad c^Tx + t_0 \\
\text{s.t.} & \quad t_i + a_i:x = b_i, \forall i = 1, 2, \ldots, \\
& \quad Q^ix - y^i = 0, \forall i = 0, 1, \ldots, \\
& \quad z_i = 1, \forall i = 0, 1, \ldots, \\
& \quad \|y^i\|^2 \leq 2t_iz_i, \forall i = 0, 1, \ldots.
\end{align*}
\]

Because

\[
\frac{1}{2}\|Q^ix\|^2 \leq t_i, \forall i = 0, 1, \ldots.
\]
Applications:

- Finance.

- Approximation of more general non-linear problems.

- Linear least squares.
Portfolio optimization. An application

- Select a portfolio of assets i.e. stocks, bonds, etc.

- Such that a large return with a low risk is obtained.

- Assumptions:
  - An initial portfolio is available.
  - A single period.
  - One of the assets is risk free i.e. cash.
Formal definition

Parameters:

- A portfolio can consist of $n$ traded assets numbered 1, 2, ... held over a period of time

- $w_j^0$ is the initial holding of asset $j$ where $\sum_j w_j^0 > 0$.

- $r_j$ is the return on asset $j$ assumed to be a random variable. $r$ has a known mean $\bar{r}$ and covariance $\Sigma$. 
Variables:

- $x_j$ is the amount of asset $j$ traded.
  
  - If $x_j > 0$, then the amount of asset $j$ is increased (by purchasing).
  
  - If $x_j < 0$, then the amount of asset $j$ is decreased (by selling).
Tradeoff

Observe

• Return (expected return)
  \[ E[r^T(w^0 + x)] = \bar{r}^T(w^0 + x) \]

• Risk (variance)
  \[ V[r^T(w^0 + x)] = (w^0 + x)^T \Sigma (w^0 + x) \]

• High return and a small risk i.e. small variance is desired.

• There is a trade-off between return and risk.
• Expected return and variance can be nontrivial to estimate.

• By definition $\Sigma$ is positive semi-definite and

$$\text{Std. dev.} = \left\| \Sigma^{\frac{1}{2}} (w^0 + x) \right\|$$
$$= \left\| L^T (w^0 + x) \right\|$$

where $L$ is any matrix such that

$$\Sigma = LL^T$$

i.e. for instance the Cholesky factor.

• A low rank of $\Sigma$ is advantageous from a computational point of view.
First model:
\[
\begin{align*}
\min & \quad (w^0 + x)^T \Sigma (w^0 + x) \\
\text{s.t.} & \quad \bar{r}^T (w^0 + x) = t, \\
& \quad e^T x = 0,
\end{align*}
\]
where \( e := (1, \ldots, 1)^T \).

Model:

- Minimizes the variance.

- While selecting a portfolio having an expected target return of \( t \).

- Satisfying the budget or self-financing constraint.

- Can clearly be reformulated as a CQO.
Usage:

- Solved for different values of $t$.

- Investor choose the portfolio that according to his/her preferences has the best relation between risk and return.

- Nobel prize wining Markowitz model.
Hyperbolic programming

\[
\begin{align*}
\min \quad & \sum_{j} \frac{c_{j}}{x_{j}} \\
\text{s.t.} \quad & Ax = b, \\
& x \geq 0,
\end{align*}
\]

where $c_{j} > 0$. 
Conic quadratic reformulation:

\[ \min \sum_{j} c_j t_j \]
\[ \text{s.t.} \quad Ax = b, \]
\[ z_j = \sqrt{2}, \]
\[ z_j^2 \leq 2x_j t_j, \]
\[ x \geq 0. \]

Applications:

- Equilibrium in TCP networks.
- Stratified sampling.
- Stock optimization models.
Robust linear optimization

Non robust LO:

\[
\begin{align*}
\text{min} & \quad c^T x \\
\text{s.t.} & \quad a_i : x \leq b_i, \ \forall i.
\end{align*}
\]

Assume:

\[
a_i^T \in \mathcal{E}_i := \{z : z = \bar{a}_i^T + H^i y, \ \|y\| \leq 1\}
\]

where

\[
H^i \in \mathbb{R}^{n \times l_i}.
\]

Robust version (Ben-Tal and Nemirovski):

\[
\begin{align*}
\text{min} & \quad c^T x \\
\text{s.t.} & \quad a_i : x \leq b_i, \ a_i^T \in \mathcal{E}_i, \ \forall i
\end{align*}
\]

or

\[
\begin{align*}
\text{min} & \quad c^T x \\
\text{s.t.} & \quad \bar{a}_i : x + \|(H^i)^T x\| \leq b_i, \ \forall i.
\end{align*}
\]

Is a CQO.
A statistical interpretation

Assumptions:

- $a_i$: are independent Gaussian random vectors.

- $\bar{a}_i$: is the mean and $\Sigma_i$ is the covariance matrix.

Problem:

$$\min \quad c^T x$$

s.t. $\text{Prob}(a_i; x \leq b_i) \geq p, \forall i.$
Equivalent problem:

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad \bar{a}_i : x + \Phi^{-1}(p) \left\| \Sigma_i^{1/2} x \right\| \leq b_i, \forall i.
\end{align*}
\]

where

\[
\Phi(z) := \frac{1}{2\pi} \int_{-\infty}^{z} e^{-t^2/2} dt.
\]

Is a CQO for \( p \geq 0.5 \).
Let a symmetric graph be given having edge weights

\[ w_{ij} \geq 0. \]

Find a cut or equivalent a partition of the nodes into two disjoint sets

\( (S, \bar{S}) \)

such that the sum of weights of crossing edges are maximized.

Let

\[ x_j = \begin{cases} 
1, & \text{node } j \in S, \\
-1, & \text{otherwise.} 
\end{cases} \]
Observation

\[
\text{total edge weight } - \text{ weight of crossing edges} = \frac{1}{2} \sum_j \sum_i w_{ij} x_i x_j
\]

Hence,

\[
\text{weight of cut} = \frac{1}{2} \left( \frac{1}{2} \sum_i \sum_j \left( w_{ij} - w_{ij} x_i x_j \right) \right).
\]

Max cut problem

\[
\begin{align*}
\text{max } & \quad \frac{1}{4} \sum_i \sum_j (w_{ij} \left(1 - x_i x_j \right)) \\
\text{s.t.} & \quad x_j^2 = 1.
\end{align*}
\]

Equivalent problem:

\[
\begin{align*}
\text{max } & \quad \frac{1}{4} \sum_i \sum_j (w_{ij} (1 - X_{ij})) \\
\text{s.t.} & \quad X - xx^T = 0, \\
& \quad X_{ii} = 1.
\end{align*}
\]
Relaxation:

$$\max \frac{1}{4} \sum_i \sum_j (w_{ij}(1 - X_{ij}))$$
\text{s.t.} \quad X \succeq 0, \quad X_{ii} = 1.$$

Comments:

- Very good bound. (Optimal value is within 14% of relaxation).

- Provably good heuristic can be devised.

- Major result in optimization.
• SDO can provide bounds for any quadratic optimization problem.

• Bounds are sometimes surprisingly strong.

• Potentially computational expensive. Why?

• A (highly) important technique of the future?

• http://www.stanford.edu/~boyd/
Optimality conditions

Usual duality holds (almost).

Weak duality:
\[ c^T x - b^T y = x^T s \geq 0 \]
if \((x, y, s)\) is a primal-dual feasible solution.

Strong duality holds in most cases i.e.:
\[ c^T x - b^T y = x^T s = 0 \]
if and only \((x, y, s)\) is a primal-dual optimal solution.

Potential problems!

- Duality gap can occur.

- Non-attainment:
  \[
  \min \frac{1}{x} \quad \text{st} \quad x \geq 0.
  \]
Optimality conditions:

\[
\begin{align*}
Ax &= b, \\
A^T y + s &= c, \\
-c^T x - b^T y &= 0, \\
x \in \mathcal{K}, \ s \in \mathcal{K}^*. 
\end{align*}
\]

Primal infeasibility condition:

\[
\begin{align*}
b^T y &> 0, \\
A^T y + s &= 0, \\
s \in \mathcal{K}^*. 
\end{align*}
\]

Dual infeasibility condition:

\[
\begin{align*}
c^T x &< 0, \\
Ax &= 0, \\
x \in \mathcal{K}. 
\end{align*}
\]
Algorithms

Interior-point methods

Barrier approach:

\[
\begin{aligned}
\min & \quad c^T x \\
\text{s.t.} & \quad Ax = b, \\
& \quad x \in \mathcal{K}.
\end{aligned}
\]

Let

\[
B(x^k) \to -\infty
\]

for \( x^k \) approaching the boundary of \( \mathcal{K} \).

Solve

\[
\begin{aligned}
\min & \quad c^T x - \mu B(x) \\
\text{s.t.} & \quad Ax = b,
\end{aligned}
\]

for small \( \mu > 0 \).
Barriers

Linear cone:
\[ \ln(x) \]

Quadratic cone:
\[ \ln(x_1^2 - \|x_{2:n}\|^2) = \ln(x_1) + \ln \left( x_1 - \frac{\|x_{2:n}\|^2}{x_1} \right) \]

Semi-definite cone:
\[ \ln(\det(X)) \]
Optimality conditions

Lagrange function:

\[ L(x, y) := c^T x - \mu B(x) - y^T (Ax - b). \]

First-order optimality conditions:

\[ \nabla_x L(x, y) = c - \mu \nabla B(x) - A^T y = 0, \]
\[ \nabla_y L(x, y) = -Ax + b = 0. \]

Define

\[ s := \mu \nabla B(x) \]

then

\[ A^T y + s = c, \]
\[ Ax = b, \]
\[ s = \mu \nabla B(x). \]
Study

\[ s = \mu \nabla B(x) \]

Linear case:

\[ s = \mu x^{-1} \]

Quadratic case:

\[ s = \mu X^{-1} e_1 \]

\[ X := \text{mat}(x) \text{ i.e.} \]

\[ V := \text{mat}(v) = \begin{bmatrix} v_1 & v_{2:n}^T \\ v_{2:n} & v_1 I \end{bmatrix}. \]

Semi-definite case:

\[ S = \mu X^{-1}. \]
Modified

Linear case:
\[ xs = \mu. \]

Quadratic case:
\[ Xs = \begin{bmatrix} x^T s \\ x_1 s_2:2n + s_1 x_2:2n \end{bmatrix} = \mu e_1. \]

Semi-definite case:
\[ XS = \mu. \]

Complementarity conditions

Let \( \mu = 0! \)
Primal-dual algorithms

Primal-dual optimality:
\[
A^T y + s = c, \\
A x = b, \\
X S = \mu. 
\]

**WARNING:** Sloppy notation but you get the idea!

One Newton step
\[
A^T d_y + d_s = c - A^T y^0 - s^0, \\
A d_x = b - A x^0, \\
X d_s + S d_x = -X S + \mu. 
\]

for suitable chosen \( \mu \) and starting point.
Comments:

- Newton step is not well-defined always.

- Requires (Nesterov-Todd) scaling. Exists only for symmetric cones.

- Leads to a powerful primal-dual algorithm.

- Polynomial complexity (solution may not be rational).

- Hard to generalize to nonsymmetric cones.
Numerical results

- MOSEK v5.0.0.121.

- Linux server.
<table>
<thead>
<tr>
<th>Name</th>
<th>Constraints</th>
<th>Variables</th>
<th>Constraints</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6874</td>
<td>6797</td>
<td>2818</td>
<td>4148</td>
</tr>
<tr>
<td>2</td>
<td>5868</td>
<td>9612</td>
<td>5867</td>
<td>9611</td>
</tr>
<tr>
<td>3</td>
<td>902</td>
<td>2710</td>
<td>900</td>
<td>2707</td>
</tr>
<tr>
<td>4</td>
<td>6086</td>
<td>14711</td>
<td>6086</td>
<td>14711</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>6</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>2276</td>
<td>2691</td>
<td>766</td>
<td>1274</td>
</tr>
<tr>
<td>7</td>
<td>460</td>
<td>18295</td>
<td>459</td>
<td>18294</td>
</tr>
<tr>
<td>8</td>
<td>406</td>
<td>15897</td>
<td>405</td>
<td>15896</td>
</tr>
<tr>
<td>9</td>
<td>24</td>
<td>32</td>
<td>23</td>
<td>31</td>
</tr>
<tr>
<td>10</td>
<td>5868</td>
<td>11533</td>
<td>5867</td>
<td>11532</td>
</tr>
<tr>
<td>11</td>
<td>6223</td>
<td>17766</td>
<td>301</td>
<td>11844</td>
</tr>
<tr>
<td>12</td>
<td>698</td>
<td>1049</td>
<td>698</td>
<td>1049</td>
</tr>
<tr>
<td>13</td>
<td>6224</td>
<td>35532</td>
<td>302</td>
<td>29610</td>
</tr>
<tr>
<td>14</td>
<td>402</td>
<td>11886</td>
<td>402</td>
<td>11886</td>
</tr>
<tr>
<td>15</td>
<td>50</td>
<td>34</td>
<td>13</td>
<td>30</td>
</tr>
<tr>
<td>16</td>
<td>298</td>
<td>348</td>
<td>169</td>
<td>215</td>
</tr>
<tr>
<td>17</td>
<td>14745</td>
<td>84709</td>
<td>14744</td>
<td>84708</td>
</tr>
<tr>
<td>18</td>
<td>19</td>
<td>23</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>19</td>
<td>16200</td>
<td>65885</td>
<td>16199</td>
<td>49413</td>
</tr>
<tr>
<td>20</td>
<td>1793</td>
<td>1942</td>
<td>1270</td>
<td>1422</td>
</tr>
<tr>
<td>21</td>
<td>97680</td>
<td>162001</td>
<td>46560</td>
<td>78482</td>
</tr>
<tr>
<td>22</td>
<td>64800</td>
<td>261365</td>
<td>64799</td>
<td>196023</td>
</tr>
<tr>
<td>23</td>
<td>123</td>
<td>2383</td>
<td>121</td>
<td>2379</td>
</tr>
<tr>
<td>24</td>
<td>123</td>
<td>2641</td>
<td>122</td>
<td>2638</td>
</tr>
<tr>
<td>25</td>
<td>2526</td>
<td>4977</td>
<td>1717</td>
<td>4168</td>
</tr>
<tr>
<td>26</td>
<td>8337</td>
<td>18238</td>
<td>5744</td>
<td>15645</td>
</tr>
<tr>
<td>27</td>
<td>4843</td>
<td>9744</td>
<td>3289</td>
<td>8190</td>
</tr>
<tr>
<td>28</td>
<td>18086</td>
<td>37887</td>
<td>12430</td>
<td>32231</td>
</tr>
<tr>
<td>29</td>
<td>915</td>
<td>3176</td>
<td>120</td>
<td>2379</td>
</tr>
<tr>
<td>30</td>
<td>123</td>
<td>4195</td>
<td>120</td>
<td>4190</td>
</tr>
<tr>
<td>31</td>
<td>13611</td>
<td>24445</td>
<td>13587</td>
<td>22609</td>
</tr>
</tbody>
</table>
## Optimized problem:

<table>
<thead>
<tr>
<th>Name</th>
<th>Constraints</th>
<th>Quad. cones</th>
<th>Variables</th>
<th>Cone var.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4148</td>
<td>1365</td>
<td>6843</td>
<td>4025</td>
</tr>
<tr>
<td>2</td>
<td>5868</td>
<td>1923</td>
<td>9612</td>
<td>9611</td>
</tr>
<tr>
<td>3</td>
<td>900</td>
<td>4</td>
<td>2709</td>
<td>1808</td>
</tr>
<tr>
<td>4</td>
<td>6086</td>
<td>2943</td>
<td>14711</td>
<td>14711</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>766</td>
<td>270</td>
<td>1460</td>
<td>1164</td>
</tr>
<tr>
<td>7</td>
<td>459</td>
<td>9</td>
<td>18295</td>
<td>1118</td>
</tr>
<tr>
<td>8</td>
<td>405</td>
<td>5254</td>
<td>15897</td>
<td>15897</td>
</tr>
<tr>
<td>9</td>
<td>23</td>
<td>1</td>
<td>32</td>
<td>24</td>
</tr>
<tr>
<td>10</td>
<td>5867</td>
<td>3844</td>
<td>11532</td>
<td>11532</td>
</tr>
<tr>
<td>11</td>
<td>301</td>
<td>5922</td>
<td>17766</td>
<td>17766</td>
</tr>
<tr>
<td>12</td>
<td>698</td>
<td>1</td>
<td>1049</td>
<td>350</td>
</tr>
<tr>
<td>13</td>
<td>302</td>
<td>11844</td>
<td>35532</td>
<td>35532</td>
</tr>
<tr>
<td>14</td>
<td>402</td>
<td>3962</td>
<td>11886</td>
<td>11886</td>
</tr>
<tr>
<td>15</td>
<td>14</td>
<td>5</td>
<td>33</td>
<td>25</td>
</tr>
<tr>
<td>16</td>
<td>210</td>
<td>57</td>
<td>344</td>
<td>238</td>
</tr>
<tr>
<td>17</td>
<td>14744</td>
<td>28236</td>
<td>84708</td>
<td>84708</td>
</tr>
<tr>
<td>18</td>
<td>5</td>
<td>3</td>
<td>13</td>
<td>11</td>
</tr>
<tr>
<td>19</td>
<td>16199</td>
<td>16471</td>
<td>65884</td>
<td>65884</td>
</tr>
<tr>
<td>20</td>
<td>1270</td>
<td>487</td>
<td>1422</td>
<td>1422</td>
</tr>
<tr>
<td>21</td>
<td>46560</td>
<td>32400</td>
<td>110882</td>
<td>97199</td>
</tr>
<tr>
<td>22</td>
<td>64799</td>
<td>65341</td>
<td>261364</td>
<td>261364</td>
</tr>
<tr>
<td>23</td>
<td>121</td>
<td>793</td>
<td>2379</td>
<td>2379</td>
</tr>
<tr>
<td>24</td>
<td>122</td>
<td>839</td>
<td>2639</td>
<td>2637</td>
</tr>
<tr>
<td>25</td>
<td>2525</td>
<td>1</td>
<td>4976</td>
<td>2475</td>
</tr>
<tr>
<td>26</td>
<td>8336</td>
<td>1</td>
<td>18237</td>
<td>8236</td>
</tr>
<tr>
<td>27</td>
<td>4842</td>
<td>1</td>
<td>9743</td>
<td>4742</td>
</tr>
<tr>
<td>28</td>
<td>18085</td>
<td>1</td>
<td>37886</td>
<td>17885</td>
</tr>
<tr>
<td>29</td>
<td>913</td>
<td>793</td>
<td>3172</td>
<td>2379</td>
</tr>
<tr>
<td>30</td>
<td>120</td>
<td>839</td>
<td>4191</td>
<td>4191</td>
</tr>
<tr>
<td>31</td>
<td>17208</td>
<td>3614</td>
<td>30741</td>
<td>13609</td>
</tr>
</tbody>
</table>
Accuracy and efficiency:

<table>
<thead>
<tr>
<th>Name</th>
<th>Primal obj.</th>
<th>Sig. fig.</th>
<th>Iter.</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-5.3902456253e+03</td>
<td>10</td>
<td>26</td>
<td>0.4</td>
</tr>
<tr>
<td>2</td>
<td>-2.3284506027e-01</td>
<td>9</td>
<td>11</td>
<td>1.9</td>
</tr>
<tr>
<td>3</td>
<td>-1.8019712354e-01</td>
<td>7</td>
<td>19</td>
<td>11.7</td>
</tr>
<tr>
<td>4</td>
<td>-1.0334780253e-02</td>
<td>9</td>
<td>19</td>
<td>11.6</td>
</tr>
<tr>
<td>5</td>
<td>-1.4142135621e-01</td>
<td>10</td>
<td>8</td>
<td>0.0</td>
</tr>
<tr>
<td>6</td>
<td>9.6279652799e+06</td>
<td>7</td>
<td>31</td>
<td>0.1</td>
</tr>
<tr>
<td>7</td>
<td>-4.1497838732e-05</td>
<td>9</td>
<td>22</td>
<td>80.2</td>
</tr>
<tr>
<td>8</td>
<td>-5.5745820525e-05</td>
<td>9</td>
<td>34</td>
<td>73.9</td>
</tr>
<tr>
<td>9</td>
<td>-1.0983618747e+00</td>
<td>9</td>
<td>9</td>
<td>0.0</td>
</tr>
<tr>
<td>10</td>
<td>-1.5563030613e-02</td>
<td>10</td>
<td>57</td>
<td>7.3</td>
</tr>
<tr>
<td>11</td>
<td>-3.5226365277e+00</td>
<td>13</td>
<td>17</td>
<td>29.7</td>
</tr>
<tr>
<td>12</td>
<td>1.8428687768e-04</td>
<td>9</td>
<td>11</td>
<td>0.0</td>
</tr>
<tr>
<td>13</td>
<td>-3.1812503728e+00</td>
<td>12</td>
<td>26</td>
<td>97.4</td>
</tr>
<tr>
<td>14</td>
<td>-1.0068176825e-02</td>
<td>10</td>
<td>24</td>
<td>46.2</td>
</tr>
<tr>
<td>15</td>
<td>1.3482698063e+02</td>
<td>9</td>
<td>14</td>
<td>0.0</td>
</tr>
<tr>
<td>16</td>
<td>3.3972095441e+00</td>
<td>9</td>
<td>9</td>
<td>0.0</td>
</tr>
<tr>
<td>17</td>
<td>1.9171403326e+04</td>
<td>8</td>
<td>44</td>
<td>7.1</td>
</tr>
<tr>
<td>18</td>
<td>4.0973429125e-01</td>
<td>10</td>
<td>7</td>
<td>0.0</td>
</tr>
<tr>
<td>19</td>
<td>-6.5943460525e+00</td>
<td>10</td>
<td>21</td>
<td>6.2</td>
</tr>
<tr>
<td>20</td>
<td>-2.2755393309e+01</td>
<td>9</td>
<td>18</td>
<td>0.1</td>
</tr>
<tr>
<td>21</td>
<td>-9.2771622627e-01</td>
<td>9</td>
<td>17</td>
<td>29.6</td>
</tr>
<tr>
<td>22</td>
<td>-6.6394902197e+00</td>
<td>10</td>
<td>21</td>
<td>39.3</td>
</tr>
<tr>
<td>23</td>
<td>-5.0703094413e-02</td>
<td>10</td>
<td>19</td>
<td>1.0</td>
</tr>
<tr>
<td>24</td>
<td>-1.0256950693e-01</td>
<td>9</td>
<td>11</td>
<td>0.6</td>
</tr>
<tr>
<td>25</td>
<td>7.8520415550e+00</td>
<td>7</td>
<td>23</td>
<td>0.2</td>
</tr>
<tr>
<td>26</td>
<td>2.7330945916e+01</td>
<td>6</td>
<td>73</td>
<td>2.8</td>
</tr>
<tr>
<td>27</td>
<td>6.7165032259e+01</td>
<td>9</td>
<td>29</td>
<td>0.6</td>
</tr>
<tr>
<td>28</td>
<td>5.1811966331e+01</td>
<td>8</td>
<td>37</td>
<td>4.2</td>
</tr>
<tr>
<td>29</td>
<td>-1.3012270287e+01</td>
<td>9</td>
<td>17</td>
<td>1.0</td>
</tr>
<tr>
<td>30</td>
<td>-1.6289715302e+00</td>
<td>7</td>
<td>16</td>
<td>2.0</td>
</tr>
<tr>
<td>31</td>
<td>-3.2774412605e-05</td>
<td>9</td>
<td>25</td>
<td>2.9</td>
</tr>
</tbody>
</table>
Software you can try out:

- MOSEK (see http://www.mosek.com/)
- SeDuMi (see http://sedumi.mcmaster.ca/).
- Benchmarks and more links:
  
  http://plato.asu.edu/bench.html
Conclusions

• Conic optimization is an exciting extension of LOs.

• Capable of solving large problems.

• Conic quadratic optimization is already useful (in business).

• Semi-definite optimization has great potential.