What is Mosek up to

January 15, 2019

Erling D. Andersen

www.mosek.com

Mosek

- A software package.
- Solves large-scale sparse linear, quadratic and conic optimization problems.
- Stand-alone as well as embedded.
- Version 1 release in 1999.
- Version 8 is released Fall 2016.
- Version 9 Spring 2019.

For details about interfaces, trials, academic license etc. see

`https://mosek.com`.

Generic conic optimization problem

Primal form

$$
\begin{aligned}
\text{minimize} \quad & \sum_k (c^k)^T x^k \\
\text{subject to} \quad & \sum_k A^k x^k = b, \\
& x^k \in \mathcal{K}^k, \qquad \forall k,
\end{aligned}
$$

where

- $c^k \in \mathbb{R}^{n^k}$,
- $A^k \in \mathbb{R}^{m \times n^k}$,
- $b \in \mathbb{R}^m$,
- $\mathcal{K}^k$ are convex cones.

The 5 cones

MOSEK v9 will support the 5 cone types:

- Linear.
- Quadratic.
- Semidefinite.
- **Exponential**.
- **Power**.

- Almost all convex problems appearing in practice can be formulated using those 5 cones.
- See my blog post from 2010 about a lunch with Stephen Boyd at Stanford:
  - `http://erlingdandersen.blogspot.com/2010/11/which-cones-are-needed-to-represent.html`
- Until now we simply did not have a satisfactory algorithm handling the nonsymmetric cones.

The power cone

The power cone:

$$\mathcal{K}_{pow}(\alpha) := \left\{ (x, z) : \prod_{j=1}^{n} x_j^{|\alpha_j|} \geq \|z\|^{\sum_{j=1}^{n} |\alpha_j|}, x \geq 0 \right\}.$$

Examples ($\alpha \in (0, 1)$):

$$\begin{aligned}
(t, 1, x) \in \mathcal{K}_{pow}(\alpha, 1-\alpha) &\Leftrightarrow t \geq |x|^{1/\alpha}, t \geq 0, \\
(x, 1, t) \in \mathcal{K}_{pow}(\alpha, 1-\alpha) &\Leftrightarrow x^{\alpha} \geq |t|, x \geq 0, \\
(x, t) \in \mathcal{K}_{pow}(e) &\Leftrightarrow \left(\prod_{j=1}^{n} x_j \right)^{1/n} \geq |t|, x \geq 0.
\end{aligned}$$

More examples that can modelled using the power cone from Chares [1]:

- p-norm:

$$t \geq \|x\|_p .$$

- l_p cone:

$$\left\{ (x, t, s) : \sum_{j=1}^{n} \left(\frac{1}{p_i} \left(\frac{|x_j|}{t} \right)^{p_j} \right) \leq \frac{s}{t}, \, t \geq 0 \right\}$$

  where $p > 0$.

Dual power cone

- Is self-dual using a redefined inner-product.
- But is not homogeneous.
- Hence, the power cone is nonsymmetric.

The exponential cone

The exponential cone

$$\begin{aligned}\mathcal{K}_{\exp} \quad := \quad & \{(x_1, x_2, x_3) : \; x_1 \geq x_2 e^{\frac{x_3}{x_2}}, \; x_2 \geq 0\} \\ & \cup \{(x_1, x_2, x_3) : \; x_1 \geq 0, x_2 = 0, \; x_3 \leq 0\}\end{aligned}$$

Applications:

$$\begin{aligned}(t, 1, x) \in \mathcal{K}_{\exp} \quad &\Leftrightarrow \quad t \geq e^x, \\ (t, 1, \ln(a)x) \in \mathcal{K}_{\exp} \quad &\Leftrightarrow \quad t \geq a^x, \\ (x, 1, t) \in \mathcal{K}_{\exp} \quad &\Leftrightarrow \quad t \leq \ln(x), \\ (1, x, t) \in \mathcal{K}_{\exp} \quad &\Leftrightarrow \quad t \leq -x \ln(x), \\ (y, x, -t) \in \mathcal{K}_{\exp} \quad &\Leftrightarrow \quad t \geq x \ln(x/y), \text{(relative entropy)}.\end{aligned}$$

Summary

- Has been extended to handle 3 dimensional power and exponential cones.
- Reuse the presolve, the efficient linear algebra from the existing conic optimizer. One code path!
- Algorithm based on work of: Tuncel [5], Myklebust and T. [2].
- Related work: Skajaa and Ye [4], Serrano [3].
- Future: Will add the n dimensional power cone and p norm cones.

Mixed integer conic optimizer

- Has been extended to handle to the nonsymmetric cones.
- Work-in-progress: Outer approximation algorithm for solution of the relaxations.

Exponential/power cone optimization

- Hardware: Intel based server. (Xeon Gold 6126 2.6 GHz, 12 core)
- MOSEK: Version 9.0.69.beta.
- Threads: 8 threads is used in test to simulate a typical user environment.
- All timing results t are in wall clock seconds.
- Test problems: Public (e.g `cblib.zib.de`) and customer supplied.

Exponential/power cone optimization

Optimized problems

Name	# con.	# cone	# var.	# mat. var.
task_dopt3	1600	26	376	2
task_dopt16	1600	26	376	2
entolib_a_bd	26	4695	14085	0
entolib_ento2	26	4695	14085	0
task_dopt10	1600	26	376	2
task_dopt17	1600	26	376	2
entolib_a_36	37	7497	22491	0
entolib_ento3	28	5172	15516	0
task_dopt12	1600	26	376	2
task_dopt21	1600	26	376	2
entolib_a_25	37	6196	18588	0
entolib_ento45	37	9108	27324	0
entolib_a_26	37	9035	27105	0
entolib_ento25	28	10142	30426	0
entolib_a_16	37	8528	25584	0
entolib_a_56	37	9702	29106	0
exp-ml-scaled-20000	19999	20000	79998	0
entolib_entodif	40	12691	38073	0
exp-ml-20000	19999	20000	79998	0
patil3_conv	418681	413547	1264340	0
c-diaz_test_c47	164404	160000	519810	0
udomsak	97653	97653	294519	0
z19841	160767	160766	483856	0

Exponential/power cone optimization

Result

Name	P. obj.	# sig. fig.	# iter	time(s)
task_dopt3	1.5283637308e+01	9	15	0.6
task_dopt16	1.3214504661e+01	10	14	0.6
entolib_a_bd	-1.1354764143e+01	9	31	0.3
entolib_ento2	-1.1354764143e+01	9	31	0.3
task_dopt10	1.4373687193e+01	9	15	0.6
task_dopt17	1.6884207404e+01	9	16	0.6
entolib_a_36	1.0162572684e+01	4	40	0.9
entolib_ento3	-6.5012761361e+00	7	42	0.5
task_dopt12	2.3128696946e+01	10	19	0.7
task_dopt21	2.5769930311e+01	9	20	0.8
entolib_a_25	-7.9656853444e+00	7	42	0.6
entolib_ento45	-8.7854807787e+00	7	44	0.8
entolib_a_26	-7.6584824859e+00	8	73	2.8
entolib_ento25	-7.2807194689e+00	8	51	1.1
entolib_a_16	-4.7657785314e+00	7	59	1.0
entolib_a_56	-8.2834963438e+00	7	66	2.2
exp-ml-scaled-20000	-3.3123486501e+00	6	82	5.9
entolib_entodif	-6.3526961669e+00	6	46	1.2
exp-ml-20000	-1.9729502378e+04	6	91	5.9
patil3_conv	-1.0539156422e+00	6	84	90.1
c-diaz_test_c47	1.8879886741e-02	8	71	99.8
udomsak	8.0453195193e-02	6	154	567.9
z19841	-2.6100495014e+00	7	86	347.1

Work in progress: Parameterized optimization models

Solve

$$\begin{array}{lll} \min\limits_{x} & c^T x & \\ \text{st} & F(x, p^k) & \leq \quad 0.0, \end{array} \tag{1}$$

for $k = 1, \ldots, K$ where p^k is a fixed parameter vector.
Restriction:

- Structure is fixed.

Goals:

- Make it simple.
- Reuse setup work btween optimizations.
- Make warmstart easy and possible.
- Maybe do a more extensive problem analysis.

Parametric MPC Model in Fusion

$$\begin{aligned} \text{minimize} \quad & \|x\|_2^2 + \|u\|_2^2 \\ \text{subject to} \quad & x(t+1) = Ax(t) + Bu(t),\ t = 0, \ldots, T-1, \\ & x(0) = x_0, x(T) = 0, \|u\|_\infty \leq u_{\max}. \end{aligned}$$

- Fixed structure: A, B, T.
- Parameters: x_0, $u_{\max}$.

```
def makeParamModel(n, m, A, B, T):
  M = Model('MPC')

  x0   = M.parameter(n)
  umax = M.parameter()

  x = M.variable("x", [n, T+1])
  u = M.variable("u", [m, T])
```

Parametric MPC Model in Fusion, cont.

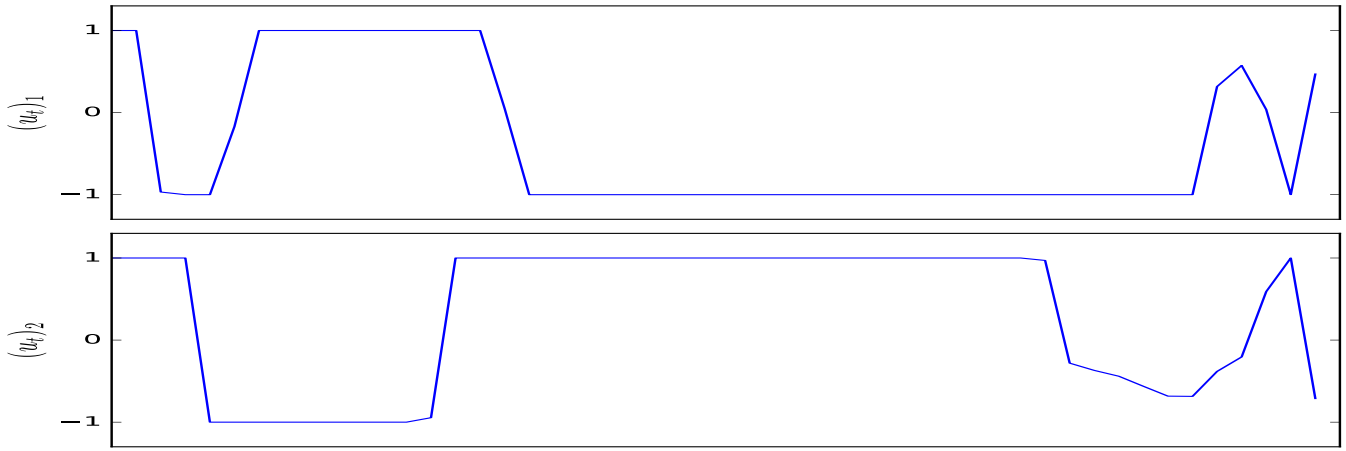
$$\begin{aligned} \text{minimize} \quad & \|x\|_2^2 + \|u\|_2^2 \\ \text{subject to} \quad & x(t+1) = Ax(t) + Bu(t),\ t = 0, \ldots, T-1, \\ & x(0) = x_0, x(T) = 0, \|u\|_\infty \leq u_{\max}. \end{aligned}$$

```
# x(t+1) = A x(t) + B u(t)
M.constraint(Expr.sub(x.slice([0,1], [n,T+1]),
                      Expr.add(Expr.mul(A, x.slice([0,0], [n,T])),
                               Expr.mul(B, u))),
             Domain.equalsTo(0))

# Starting and final point
M.constraint(Expr.sub(x0, x.slice([0,0], [n,1])), Domain.equalsTo(0))
M.constraint(x.slice([0,T], [n,T+1]),             Domain.equalsTo(0))

# Max amplitude
for i,j in product(range(m), range(T)):
    M.constraint(Expr.add(u.index(i,j), umax), Domain.greaterThan(0))
    M.constraint(Expr.sub(umax, u.index(i,j)), Domain.greaterThan(0))

obj = M.variable()
M.constraint(Expr.vstack(obj, Expr.flatten(x), Expr.flatten(u)), Domain.inQCone())
M.objective(ObjectiveSense.Minimize, obj)

return M   # Parametrized Model
```

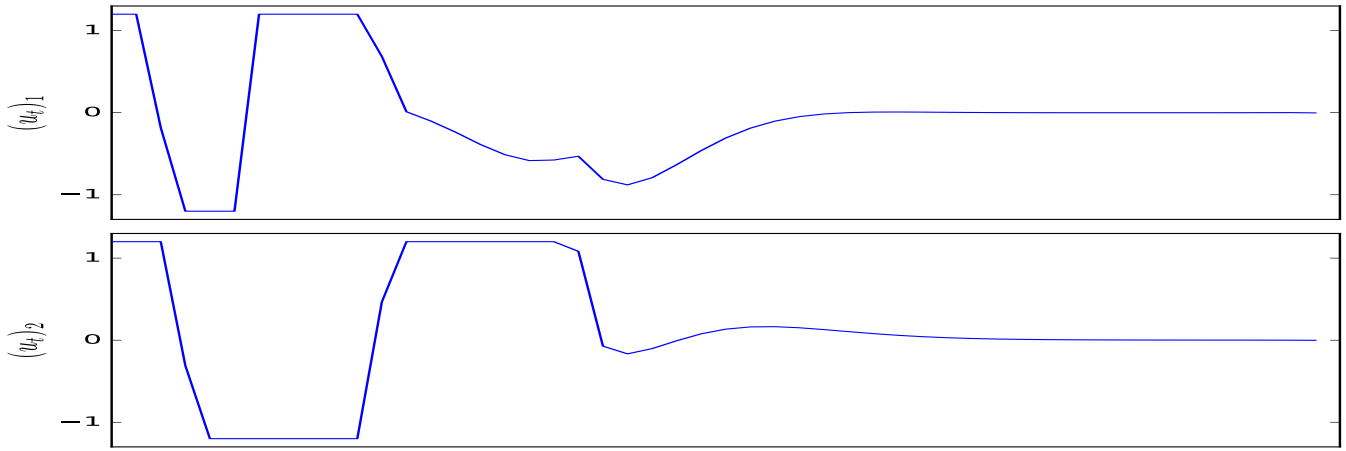
Using the parametrized model

$x = (x_1, \ldots, x_8), u = (u_1, u_2), T = 50$, random A, B, x_0

- `solve(M, x0 = x_0, umax = 1.0)`



- `solve(M, x0 = x_0, umax = 1.2)`



The take home message

- Version 9 supports the power and exponential cones.
  - A breaktrough!
- Future: Easy and efficient handling of parameterized optimization models.

References I

[1] Peter Robert Chares.
Cones and interior-point algorithms for structed convex optimization involving powers and exponentials.
PhD thesis, Ecole polytechnique de Louvain, Universitet catholique de Louvain, 2009.

[2] T. Myklebust and L. Tunçel.
Interior-point algorithms for convex optimization based on primal-dual metrics.
Technical report, 2014.

[3] Santiago Akle Serrano.
Algorithms for unsymmetric cone optimization and an implementation for problems with the exponential cone.
PhD thesis, Stanford University, 2015.

References II

[4] Anders Skajaa and Yinye Ye.
A homogeneous interior-point algorithm for nonsymmetric convex conic optimization.
Math. Programming, 150:391–422, May 2015.

[5] L. Tunçel.
Generalization of primal-dual interior-point methods to convex optimization problems in conic form.
Foundations of Computational Mathematics, 1:229–254, 2001.