



Introduction to Mosek

Modern Optimization in Energy, 28 June 2018

Michał Adamaszek

www.mosek.com



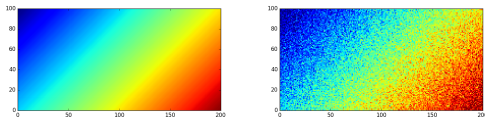


- Started in 1999 by Erling Andersen
- Convex conic optimization package + MIP
- LP, QP, SOCP, SDP, other nonlinear cones
- Low-level optimization API
 - C, Python, Java, .NET, Matlab, R, Julia
- Object-oriented API *Fusion*
 - C++, Python, Java, .NET
- 3rd party
 - GAMS, AMPL, CVXOPT, CVXPY, YALMIP, PICOS, GPkit
- Conda package, .NET Core package
- Upcoming v9

Example: 2D Total Variation

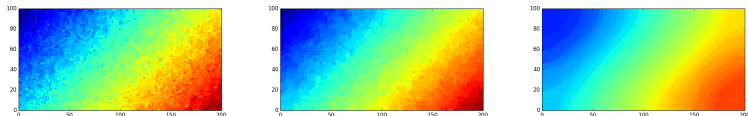


Someone sends you the left signal but you receive noisy f (right):



How to denoise/smoothen out/approximate u ?

$$\begin{aligned} &\text{minimize} && \sum_{ij} (u_{i,j} - u_{i+1,j})^2 + \sum_{ij} (u_{i,j} - u_{i,j+1})^2 \\ &\text{subject to} && \sum_{ij} (u_{i,j} - f_{i,j})^2 \leq \sigma. \end{aligned}$$





A conic problem in canonical form:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax + b \in \mathcal{K} \end{aligned}$$

where \mathcal{K} is a product of cones:

- linear:

$$K = \mathbb{R}_{\geq 0}$$

- quadratic:

$$K = \{x \in \mathbb{R}^n : x_1 \geq \sqrt{x_2^2 + \dots + x_n^2}\}$$

- semidefinite:

$$K = \{X \in \mathbb{R}^{n \times n} : X = FF^T\}$$



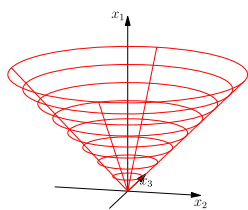


- exponential cone:

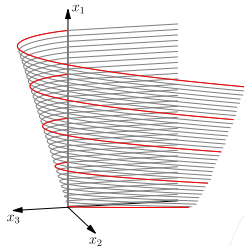
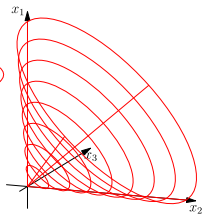
$$K = \{x \in \mathbb{R}^3 : x_1 \geq x_2 \exp(x_3/x_2), x_2 > 0\}$$

- power cone:

$$K = \{x \in \mathbb{R}^3 : x_1^{p-1} x_2 \geq |x_3|^p, x_1, x_2 \geq 0\}, p > 1$$



$$x_1 \geq \sqrt{x_2^2 + x_3^2}, \quad 2x_1x_2 \geq x_3^2$$



$$x_1 \geq x_2 \exp(x_3/x_2)$$



Lots of functions and constraints are representable using these cones.

$$|x|, \|x\|_1, \|x\|_2, \|x\|_\infty, \|Ax + b\|_2 \leq c^T x + d$$

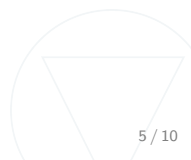
$$xy \geq z^2, x \geq \frac{1}{y}, x \geq y^p, t \geq \left(\sum_i |x_i|^p \right)^{1/p} = \|x\|_p$$

$$t \leq \sqrt{xy}, t \leq (x_1 \cdots x_n)^{1/n}, \text{geometric programming (GP)}$$

$$t \leq \log x, t \geq e^x, t \leq -x \log x, t \geq \log \sum_i e^{x_i}, t \geq \log \left(1 + \frac{1}{x} \right)$$

$$\det(X)^{1/n}, t \leq \lambda_{\min}(X), t \geq \lambda_{\max}(X)$$

$$\text{convex } (1/2)x^T Qx + c^T x + q$$





Find a

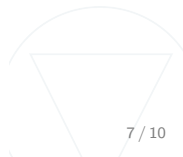
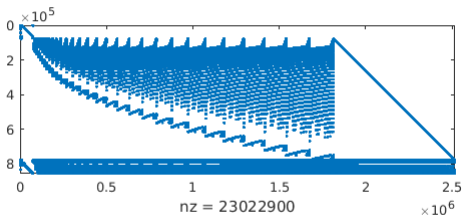
- natural,
- practical,
- important,
- convex

optimization problem, which cannot be expressed in conic form.





- primal simplex and dual simplex for linear problems
- primal-dual interior point method optimizer for all conic problems
- automatic dualization
- QPs transformed to conic quadratic form
- branch and bound and cut integer optimizer
- **exploits sparsity**





$$\begin{aligned} \min \quad & \sum c_i P_{G_i} \\ \text{s.t.} \quad & P_{G_i}^{min} \leq P_{G_i} \leq P_{G_i}^{max} \\ & \mathbf{B} \cdot \boldsymbol{\theta} = \mathbf{P}_G - \mathbf{P}_D \\ & |\theta_i - \theta_j|/x_{ij} \leq P_{ij,max} \end{aligned}$$

```
def dcopf(ng, nb, PGmin, PGmax, Pmax, B, x, PD, c):
    M = Model()
    PG = M.variable(ng, Domain.inRange(PGmin, PGmax))
    theta = M.variable(nb)

    M.constraint(Expr.sub(Expr.mul(B, theta), Expr.sub(PG, PD)),
                 Domain.equalsTo(0.0))
    M.constraint(Expr.sub(Var.hrepeat(theta, nb),
                          Var.vrepeat(theta.transpose(), nb)),
                 Domain.lessThan(Pmax*x))

    M.objective(ObjectiveSense.Minimize, Expr.dot(c, PG))
```



A basic version of the Unit Commitment Problem with:

- quadratic generation costs,
- ramp constraints,
- minimal uptime and downtime,
- startup costs,

as an example of MISOCP. See:

- <https://mosek.com/documentation>
 - The MOSEK notebook collection
 - Unit commitment





Licensing:

- Trial 30-day license
- Personal academic license
- Group academic license
- Commercial licenses

More:

- www.mosek.com
- <https://github.com/MOSEK/Tutorials>
- <https://themosekblog.blogspot.com/>
- support@mosek.com
- Modeling Cookbook www.mosek.com/documentation

Thank you!