Extending MOSEK's algorithm to the general power cone

Utkarsh Detha
MOSEK ApS

International Conference on Continuous Optimization
July 23, 2025

Outline

Introduction

Conic programming

A cone program (CP) in *standard* form:

$$\begin{aligned} p_\star = \min_x \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \in \mathbf{K} \end{aligned} \tag{Primal CP}$$

Dual conic program:

$$\begin{aligned} d_\star = \max_{y,s} \quad & b^T y \\ \text{s.t.} \quad & A^T y + s = c \\ & s \in \mathbf{K}^* \end{aligned} \tag{Dual CP}$$

Assume: $\mathbf{K}$ and $\mathbf{K}^*$ are *proper* cones and A is full row-rank.

The general power cone

Given $\sum_{i=1}^{m} \alpha_i = 1$, the general power cone [1] is defined as:

$$\mathbf{K}_{\alpha}^{(m,n)} = \left\{ (x_{\mathcal{I}}, x_{\mathcal{J}}) \in \mathbf{R}_{+}^{m} \times \mathbf{R}^{n} : \prod_{i \in \mathcal{I}} x_i^{\alpha_i} \geq ||x_{\mathcal{J}}||_2 \right\}$$

An *LHSC* barrier, with $\vartheta = m + 1$:

$$F(x) = -\log\left(\prod_{i \in \mathcal{I}} x_i^{2\alpha_i} - ||x_{\mathcal{J}}||_2^2 \right) - \sum_{i \in \mathcal{I}} (1 - \alpha_i) \log(x_i)$$

$\rightarrow$ can be split into $m - 1$ cones of dimension 3 each.

Homogenous self-dual embedding

$$
\begin{aligned}
\min_{x,y,s,\tau,\kappa} \quad & 0 \\
\text{s.t.} \quad & Ax - b\tau && = 0 \\
& A^T y - c\tau + s && = 0 \\
& b^T y - c^T x - \kappa && = 0 \\
& x \in \mathbf{K}, \quad s \in \mathbf{K}^*, \tau \geq 0, \kappa \geq 0
\end{aligned}
$$

$$\text{where, } z = (x, \tau, s, \kappa, y)$$

- $\tau^\star > 0 \implies (\frac{x^\star}{\tau}, \frac{s^\star}{\tau}, \frac{y^\star}{\tau})$ is a primal-dual feasible solution.
- $\kappa^\star > 0 \implies b^T y^\star > 0$(primal infeasible) and/or $c^T x^\star < 0$ (dual infeasible)

The primal-dual central path

Given *strictly* feasible point z_0, points along the central path are defined for $\mu \in (0, 1]$,

$$\begin{aligned} G(z_\mu) &= \mu G(z_0) \\ s = -\mu \nabla F(x) \quad &; \quad x = -\mu \nabla F_*(s) \end{aligned} \tag{1}$$

where,

$$F_*(s) = \sup_{x \in \mathrm{int}(K)} \{-\langle x, s\rangle - F(x)\} \tag{2}$$

is the conjugate to the primal barrier.

Challenge: no closed-form expression for the conjugate is known. *Solution*:[2]

Scaling for primal-dual methods

Given $W \in \mathbf{R}^{m \times m}$ s.t. $v = Wx = W^{-T}s$ and $\widetilde{v} = W\widetilde{x} = W^{-T}\widetilde{s}$, the central path becomes,

$$\begin{aligned} G(z_\mu) &= \mu G(z_0) \\ v &= \mu \widetilde{v} \end{aligned} \tag{3}$$

- Symmetric cones: $W^T W = \nabla^2 F(w)$ for some $w \in K$ [3].
- Non-symmetric cones: Hessian of the power cone barrier satisfies only one secant equation. How should one define the *scaling* matrices?

Scaling matrices for non-symmetric cones

Quasi-Newton method for multiple secant equations

A BFGS-type update with *multiple* secant equations, i.e. the solution to [4, 5]:

$$\begin{aligned} \inf \quad & \|(W^T W)^{-1} - (\mu_K \nabla^2 F_K(x))^{-1}\|_{W^T W} \\ \text{s.t.} \quad & W^T W x = s \\ & W^T W \widetilde{x} = \widetilde{s} \\ & W^T W \in \mathbf{K}_{\text{PSD}} \end{aligned}$$

where,

$$\mu_K = \frac{\langle x, s \rangle}{\vartheta_K}$$

for $x \in K$ and $s \in K^*$.

BFGS scaling matrices

$$W^T W = \mu_K \nabla^2 F(x) + \frac{\delta_s q_s^T + q_s \delta_s^T}{2\mu_K \vartheta_K} - \frac{\mu_K}{\langle r, r \rangle_{\nabla^2 F}} r r^T$$

BFGS scaling for general power cones

$$W^T W = \underbrace{\mu_K \nabla^2 F(x)}_{\text{diag + rank-2}} + \overbrace{\frac{\delta_s q_s^T + q_s \delta_s^T}{2\mu_K \vartheta_K} - \frac{\mu_K}{\langle r, r \rangle_{\nabla^2 F}} r r^T}^{rank-3}$$

BFGS scaling for general power cones

$$W^T W = \mu_K \mathbf{D} + U \mathbf{R} U^T - V \mathbf{S} V^T$$

$$\text{where, } \mathbf{R} = \begin{bmatrix} \mu_K & 0 \\ 0 & \frac{1}{4\mu_K \vartheta} \end{bmatrix} \quad , \quad \mathbf{S} = \begin{bmatrix} \mu_K & 0 & 0 \\ 0 & \frac{1}{4\mu_K \vartheta_K} & 0 \\ 0 & 0 & \frac{\mu_K}{\langle r_N, r_N \rangle_{\nabla^2 F}} \end{bmatrix}$$

$$\text{and, } \mathbf{D} = \text{Diagonal}\left(\frac{\widetilde{s}}{x}\right)$$

Computing the search directions

Computing search directions

$$\begin{bmatrix} 0 & A & -b \\ -A^T & 0 & c \\ b^T & -c^T & 0 \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \\ \Delta \tau \end{bmatrix} - \begin{bmatrix} 0 \\ \Delta s \\ \Delta \kappa \end{bmatrix} = \begin{bmatrix} r_p \\ r_d \\ r_g \end{bmatrix}$$

$$W\Delta x + W^{-T}\Delta s = r_{xs}$$

$$\tau\Delta\kappa + \kappa\Delta\tau = r_{\tau\kappa}$$

Computing search directions

$$\begin{bmatrix} W^TW & -A^T & c \\ A & 0 & -b \\ -c^T & b^T & \tau^{-1}\kappa \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \tau \end{bmatrix} = \begin{bmatrix} r_d + W^T r_{xs} \\ r_p \\ r_g + \tau^{-1} r_{\tau\kappa} \end{bmatrix}$$

Exploiting sparsity

$$\begin{bmatrix} -W^T W & A^T \\ A & 0 \end{bmatrix} = \begin{bmatrix} -\mathbf{D} - U\mathbf{R}U^T + V\mathbf{S}V^T & A^T \\ A & 0 \end{bmatrix}$$

$\rightarrow$ use modified Schur complement method originally introduced for handling dense columns in LPs[6].

Exploiting sparsity

$$
\begin{bmatrix}
-\mathbf{D} & A^T & U & V & -c \\
A & 0 & 0 & 0 & -b \\
U^T & 0 & \mathbf{R}^{-1} & 0 & 0 \\
V^T & 0 & 0 & -\mathbf{S}^{-1} & 0 \\
-c^T & b^T & 0 & 0 & \tau^{-1}\kappa
\end{bmatrix}
\begin{bmatrix}
\Delta x \\ \Delta y \\ \Delta t_u \\ \Delta t_v \\ \Delta \tau
\end{bmatrix}
=
\begin{bmatrix}
-r_d - W^T r_{xs} \\ r_p \\ 0 \\ 0 \\ r_g + \tau^{-1} r_{\tau\kappa}
\end{bmatrix}
\tag{4}
$$

Exploiting sparsity

$$\begin{bmatrix} A\mathbf{D}^{-1}A^T & A\mathbf{D}^{-1}U & A\mathbf{D}^{-1}V \\ U^T\mathbf{D}^{-1}A^T & \mathbf{R}^{-1} + U^T\mathbf{D}^{-1}U & U^T\mathbf{D}^{-1}V \\ V^T\mathbf{D}^{-1}A^T & V^T\mathbf{D}^{-1}U & -\mathbf{S}^{-1} + V^T\mathbf{D}^{-1}V \end{bmatrix}$$

Exploiting sparsity

$$\begin{bmatrix} \mathbf{R}^{-1} + U^T\mathbf{D}^{-1}U - \overline{U}^T A\mathbf{D}^{-1}U & U^T\mathbf{D}^{-1}V - \overline{U}^T A\mathbf{D}^{-1}V \\ V^T\mathbf{D}^{-1}U - \overline{V}^T A\mathbf{D}^{-1}U & -\mathbf{S}^{-1} + V^T\mathbf{D}^{-1}V - \overline{V}^T A\mathbf{D}^{-1}V \end{bmatrix}$$

where,

$$\overline{U} = (A\mathbf{D}^{-1}A^T)^{-1}A\mathbf{D}^{-1}U$$

$$\overline{V} = (A\mathbf{D}^{-1}A^T)^{-1}A\mathbf{D}^{-1}V$$

Motivation for the search directions

Search directions

Centrality condition:

$$s + \Delta s = -\mu \nabla F(x + \Delta x)$$

use linear approximation for gradient and pre-multiply W^{-T},

$$W^{-T} \Delta s + \underbrace{\mu W^{-T} \nabla^2 F(x)}_{\approx W \text{ near central}} \Delta x = -v + \mu \tilde{v}$$

Affine and centering directions

For a given centering parameter γ,

$$\underbrace{\overbrace{W^{-T}\Delta s + W\Delta x = -v}^{\text{Affine}} + \gamma\mu\tilde{v}}_{\text{Affine + Centering}} \tag{5}$$

Search directions

Centrality condition:

$$s + \Delta s = -\mu \nabla F(x + \Delta x)$$

use quadratic approximation and pre-multiply with W^{-T},

$$W^{-T} \Delta s + W \Delta x = -v + \mu \tilde{v} - \frac{1}{2} \mu W^{-T} \nabla^3 F(x)[\Delta x, \Delta x]$$

Higher-order correction

A full affine step $(\Delta x_a, \Delta s_a)$ would make the higher-order term:

$$
\begin{aligned}
& -\frac{1}{2}\mu\nabla^3 F(x)[\Delta x_a, \Delta x_a] \\
= & -\frac{1}{2}\mu\nabla^3 F(x)[\Delta x_a, -x - W^{-1}W^{-T}\Delta s_a] \\
= & \frac{\mu}{2}\nabla^3 F(x)[\Delta x_a, x] + \frac{\mu}{2}\nabla^3 F(x)[\Delta x_a, \frac{\nabla^2 F(x)^{-1}}{\mu}\Delta s_a] \\
= & -\mu\nabla^2 F(x)\Delta x_a + \underbrace{\frac{1}{2}\nabla^3 F(x)[\Delta x_a, \nabla^2 F(x)^{-1}\Delta s_a]}_{-\eta}
\end{aligned}
$$

Combined direction

For a given centering parameter γ,

$$W^{-T}\Delta s + W\Delta x = -v + \gamma\mu\tilde{v} - W^{-T}\eta \tag{6}$$

Outline of the algorithm

Key steps in the primal-dual algorithm for non-symmetric cones [7]

1. Starting point: $x_0 = s_0 = -\nabla F(x_0)$
2. Scaling info: compute $\mathbf{D}^{-1}$, $\mathbf{R}^{-1}$, $\mathbf{S}^{-1}$, U and V.
3. Pure affine step + Corrector evaluation
4. Affine step-length to boundary (α_a) and centering parameter γ
5. Combined step and largest step within $N\beta$ neighbourhood.
6. Check termination and loop back to step 2.

Solving a simple problem with general power cone

MLE of a convex density function

Given outcomes $y_1 < \cdots < y_n$, estimate underlying convex density function [8]

$$
\begin{aligned}
\max_{u,x} \quad & u \\
\text{s.t.} \quad & \frac{x_{i+1} - x_i}{y_{i+1} - y_i} - \frac{x_{i+2} - x_{i+1}}{y_{i+2} - y_{i+1}} \leq 0 \ \forall \ i = 1, \cdots, n-2 \\
& \sum_{i=1}^{n-1} (y_{i+1} - y_i) \frac{x_{i+1} + x_i}{2} = 1 \\
& (x, u) \in \mathcal{K}_{\alpha}^{(n,1)}
\end{aligned}
\tag{Conic MLE}
$$

$\rightarrow$ **Weighted** MLE (for clustered y_i's) is modelled using different α_i's.

Impact of higher-order correction

dim(K)	η	μ	**Status**	**Iter**
500	off	8.8e-10	Optimal	113
	on	2.5e-09	NearOpt	127
750	off	1.4e-09	NearOpt	112
	on	3.2e-09	NearOpt	142
1000	off	9.7e-10	Optimal	90
	on	0.0022621	Unknown	301
1500	off	7.2e-10	Optimal	98
	on	0.000246772	Unknown	301
2000	off	7.4e-10	Optimal	101
	on	0.000151739	Unknown	301

N-D vs 3D Power cone

dim(K)	Solver	μ	Status	Iter	Time (s)
500	ND (:off)	8.8e-10	Optimal	113	1.8
	3D (:on)	4.7e-11	NearOpt	81	16.0
750	ND (:off)	1.4e-09	NearOpt	112	2.6
	3D (:on)	4.5e-09	NearOpt	52	21.0
1000	ND (:off)	9.7e-10	Optimal	90	2.7
	3D (:on)	2.3e-08	NearOpt	50	40.0
1500	ND (:off)	7.2e-10	Optimal	98	4.4
	3D (:on)	3.1e-08	NearOpt	67	150.0
2000	ND (:off)	7.4e-10	Optimal	101	5.8
	3D (:on)	2.8e-08	NearOpt	86	430.0

Summary

- Outlined MOSEK's algorithm for non-symmetric conic problems
- Extension to general power cone benefits from modified Schur complement approach
- The higher-order correction can be motivated analogously to the linear case
- Effectiveness of the higher-order correction does not scale to larger cones directly

References

[1] Robert Chares. “Cones and interior point algorithms for structured convex optimization involving powers and exponentials”. PhD. Universite catholique de Louvain Ecole Polytechnique de Louvain, 2009.

[2] Lea Kapelevich et al. “Computing Conjugate Barrier Information for Nonsymmetric Cones”. en. In: *Journal of Optimization Theory and Applications* (Aug. 2022).

[3] Yu. E. Nesterov and M. J. Todd. “Primal-Dual Interior-Point Methods for Self-Scaled Cones”. In: *SIAM Journal on Optimization* 8.2 (May 1998). Publisher: Society for Industrial and Applied Mathematics, pp. 324–364.

[4] Levent Tunçel. “Generalization of Primal—Dual Interior-Point Methods to Convex Optimization Problems in Conic Form”. en. In: *Foundations of Computational Mathematics* 1.3 (July 2001), pp. 229–254.

[5] Robert B. Schnabel. *Quasi-Newton Methods Using Multiple Secant Equations ; CU-CS-247-83*. http://id.loc.gov/vocabulary/iso639-2/eng.

[6] Knud D. Andersen. “A modified Schur-complement method for handling dense columns in interior-point methods for linear programming”. en. In: *ACM Transactions on Mathematical Software* 22.3 (Sept. 1996). Publisher: Association for Computing Machinery (ACM), pp. 348–356.

[7] Joachim Dahl and Erling D. Andersen. “A primal-dual interior-point algorithm for nonsymmetric exponential-cone optimization”. en. In: *Mathematical Programming* 194.1 (July 2022), pp. 341–370.

[8] T. Terlaky and J Vial. "Computing Maximum Likelihood Estimators of Convex Density Functions". en. In: *SIAM Journal on Scientific and Statistical Computing* 19.2 (1998), pp. 675–694.