Interior point and outer approximation methods for conic optimization

Chris Coey

April 22, 2022

Introduction

Conic optimization

Primal over $x \in \mathbb{R}^n$

$$\begin{aligned} \inf \quad & c'x : \\ & b - Ax = 0 \\ & h - Gx \in \mathcal{K} \end{aligned}$$

Dual over $y \in \mathbb{R}^p, z \in \mathbb{R}^q$

$$\begin{aligned} \sup \quad & -b'y - h'z : \\ & c + A'y + G'z = 0 \\ & z \in \mathcal{K}^* \end{aligned}$$

$\mathcal{K}$ is a *proper* cone: convex, closed, solid, and pointed. In practice, $\mathcal{K} = \mathcal{K}_1 \times \cdots \times \mathcal{K}_K$ is a Cartesian product of recognized cones.

Usually exists a certificate of optimality or of primal or dual infeasibility.

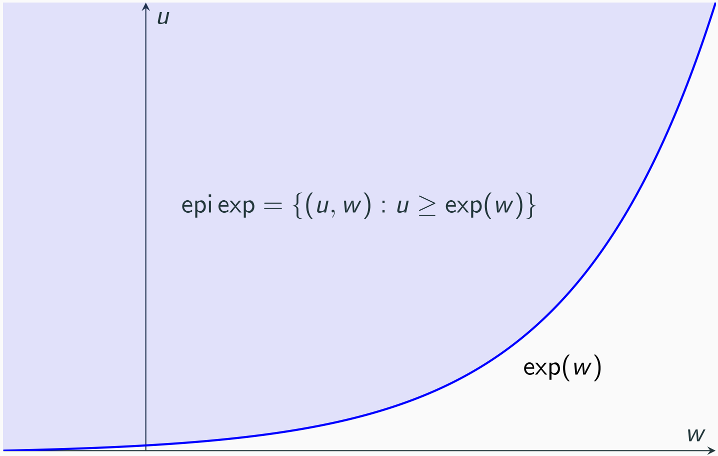
Generalizes LP, SOCP, SDP, which cannot represent e.g. $u \geq \exp(w)$.

For mixed integer (MI) conic problems, the primal has $x_i \in \mathbb{Z}, \forall i \in [\![N]\!]$.

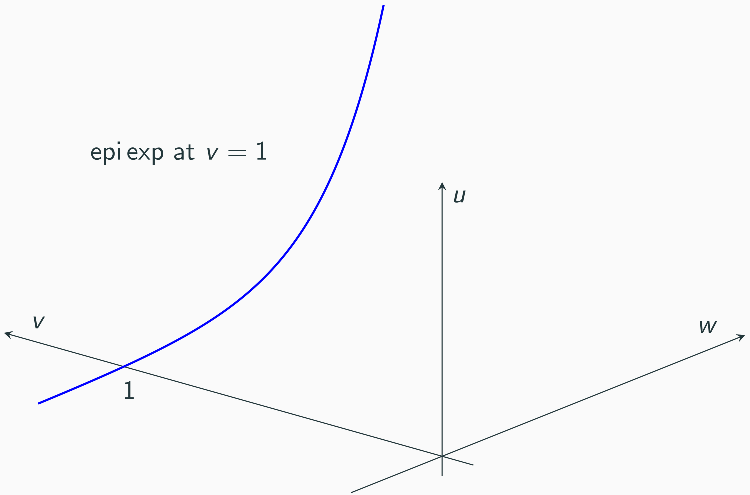
Any (MI) convex problem may be homogenized into (MI) conic form.

Conic duality enables powerful algorithms for (MI) convex problems.

Perspective transformations

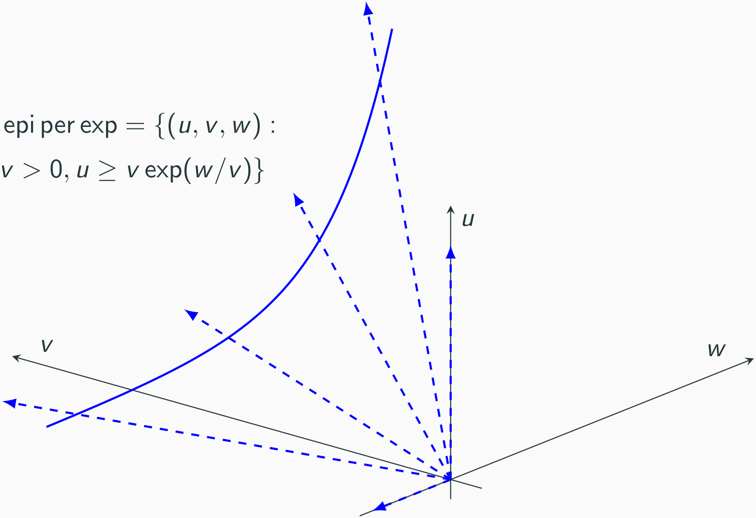


Perspective transformations



Perspective transformations

$$\operatorname{epi}\operatorname{per}\exp = \{(u, v, w) :$$
$$v > 0, u \geq v \exp(w/v)\}$$



Goals and contributions

We improve the generality and practical performance of:

- interior point methods (IPMs) for continuous conic formulations,
- outer approximation methods (OAMs) for MI conic formulations.

We introduce new solvers, Hypatia (IPM) and Pajarito (OAM):

- open source and accessible through JuMP/MathOptInterface,
- highly extensible/customizable algorithmic components,
- generic cone interfaces allow users to add new cones.

We develop efficient, stable oracle procedures for dozens of useful cones.

We model hundreds of (MI) conic formulations from 50 applied examples.

Our benchmarking explores which algorithmic features and what types of conic formulations lead to the best performance of IPMs/OAMs.

We argue for expanding the class of cones recognized by conic solvers.

Chapters 1-3: continuous conic optimization

Chapter 1: IPMs and Hypatia

Conic IPMs

Conic IPMs with polynomial iteration complexity use *logarithmically homogeneous self-concordant barriers* (LHSCBs) for proper cones.

E.g. $-\log$ for $\mathbb{R}_{\geq}$, $-\text{logdet}$ for PSD matrices $\mathbb{S}_{\succeq}$.

Until recently, conic solvers only supported the standard symmetric cones.

The Skajaa and Ye (2015) IPM (*SY*) supports nonsymmetric cones.

SY needs tractable oracles for each cone in the primal: an initial interior point, feasibility check, and gradients and Hessians for the LHSCB.

We generalize SY to support any *exotic* cone $\mathcal{K}$ that has tractable LHSCB oracles for either $\mathcal{K}$ or $\mathcal{K}^*$.

We define two dozen exotic cones through the cone interface.

Most only have analytic/closed-form oracles for $\mathcal{K}$ or $\mathcal{K}^*$, not both.

IPM stepping procedures

The *homogeneous self-dual embedding* (HSDE) is a conic feasibility problem, a solution to which provides a conic certificate (if one exists).

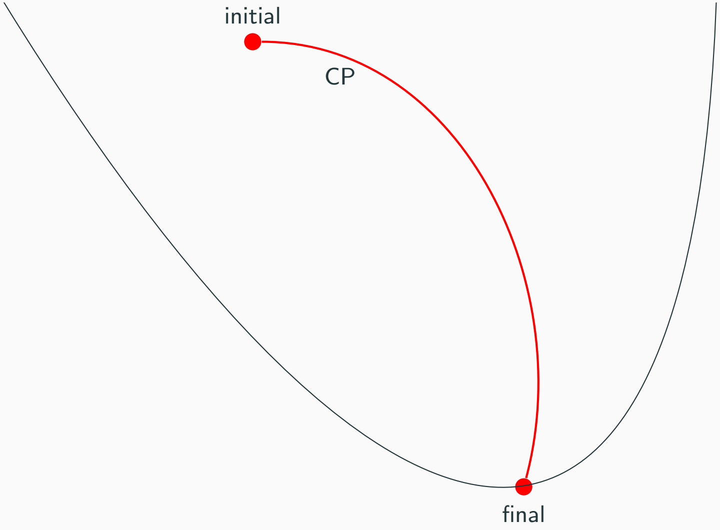
SY approximately traces the *central path* (CP) to an HSDE solution.

- *Prediction* steps take us towards a solution, by treating the CP like a dynamical system. A line search maintains CP proximity.
- *Centering* steps take us towards the CP, from which good prediction directions can be obtained. This is like a Newton step.

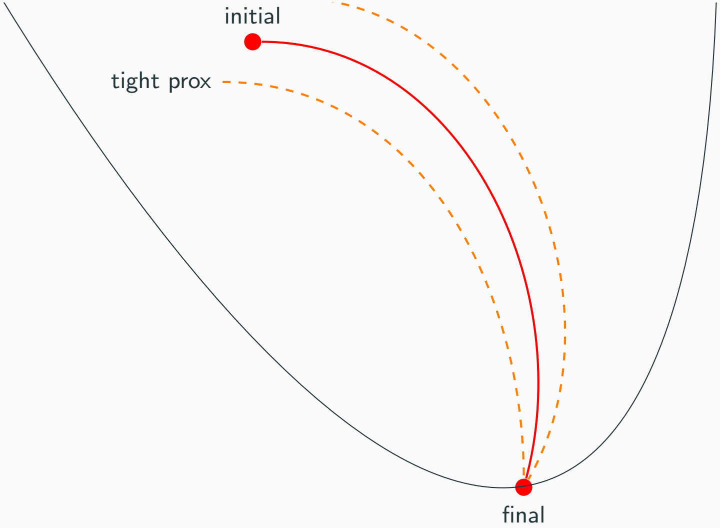
We enhance the practical performance of SY's stepping procedure.

- Use a less restrictive CP proximity condition.
- Adjust the search directions using a new third order directional derivative barrier oracle, and search on a quadratic curve.
- Combine the prediction and centering phases.

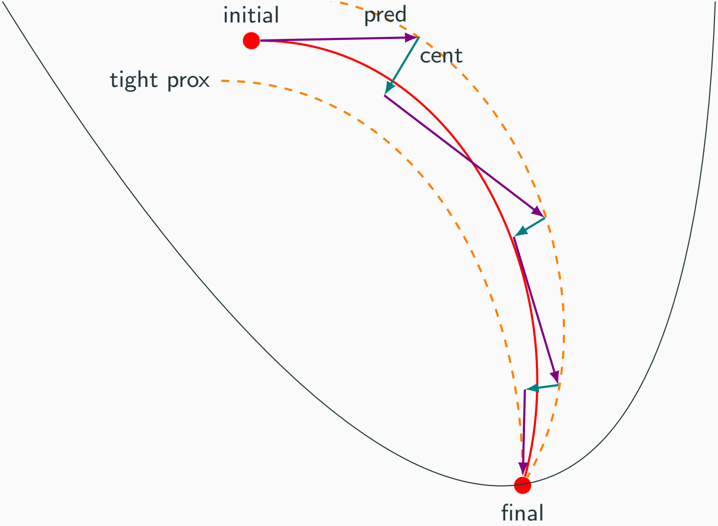
Stepping enhancements



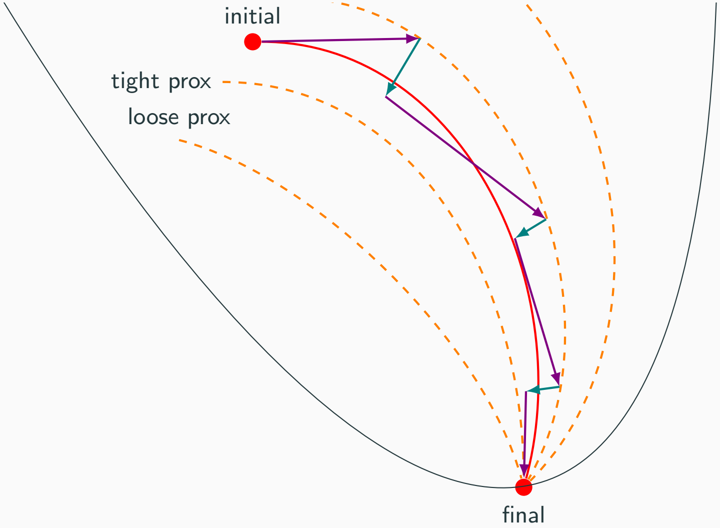
Stepping enhancements



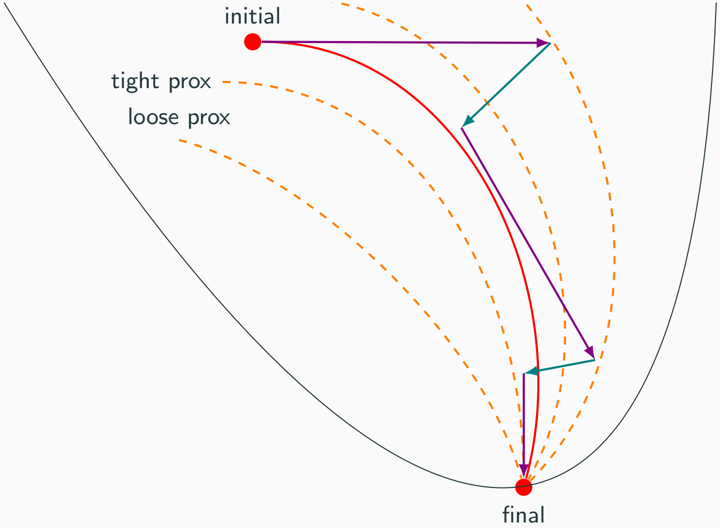
Stepping enhancements



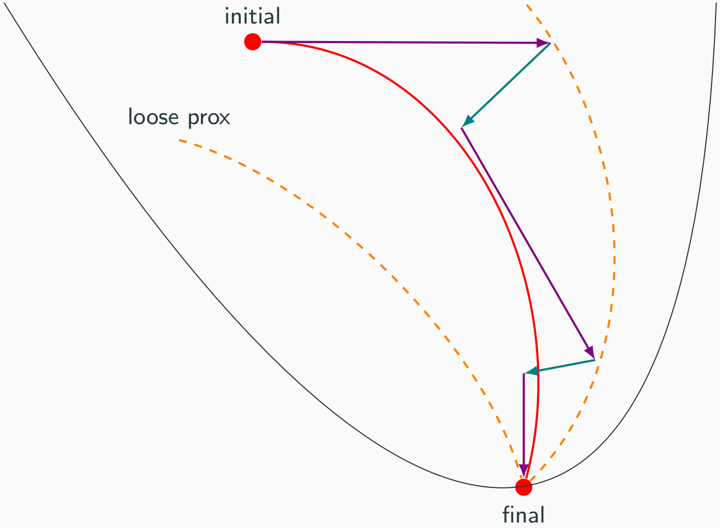
Stepping enhancements



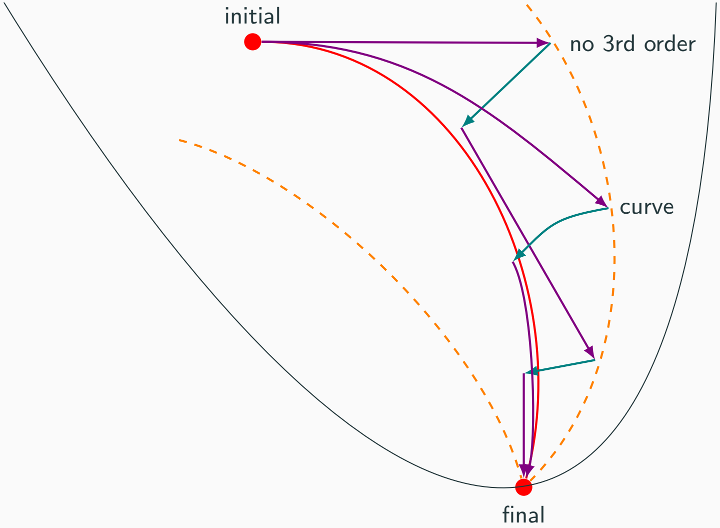
Stepping enhancements



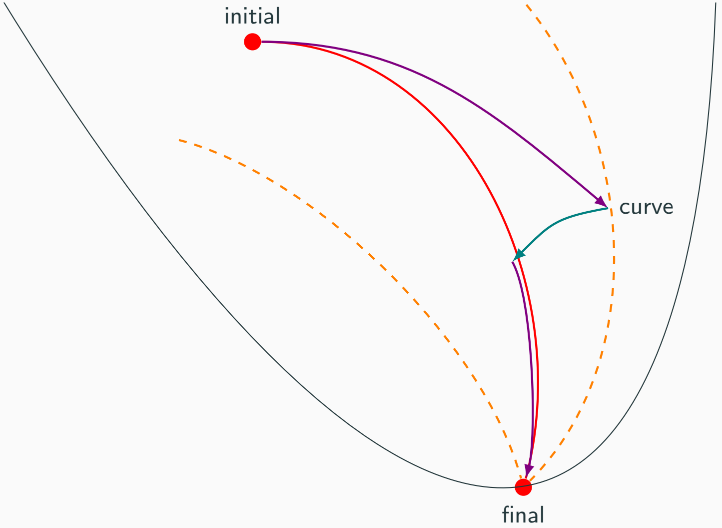
Stepping enhancements



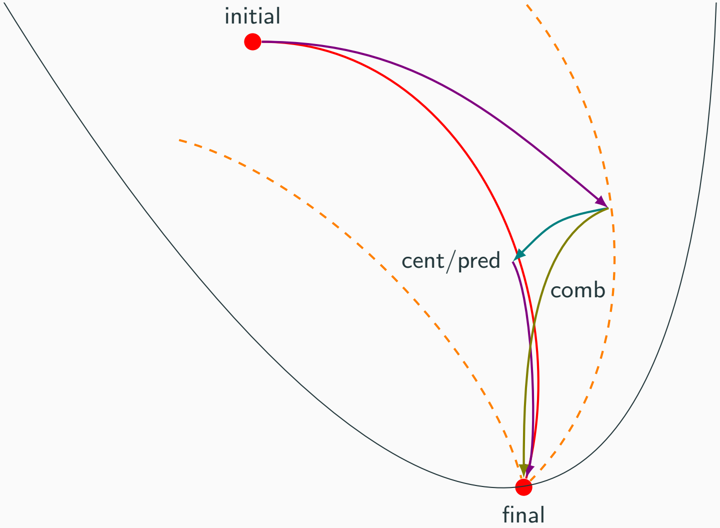
Stepping enhancements



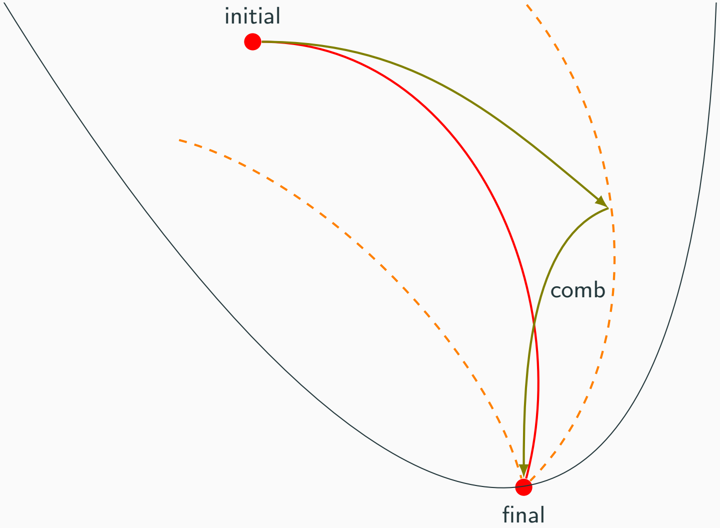
Stepping enhancements



Stepping enhancements



Stepping enhancements



Stepping enhancement results

We generate a benchmark set of 379 instances over Hypatia's cones.

- From 37 applied examples, each with multiple formulation types.
- Most are primal-dual feasible, some are primal or dual infeasible.
- All are small-medium and solve in under 30 minutes.

We set tolerances to around 10^{-7} (for most) and verify certificates.

We report shifted geometric means over the instances solved.

stepper	solved	iterations	impr	time (ms)	impr
SY	371	101.3	-	2131	-
prox	369	64.7	36%	1317	38%
third	372	29.7	54%	742	44%
comb	367	18.3	38%	624	16%

Overall: iterations and solve time improve by at least 80% and 70%.

Solving for directions

The HSDE has $n + p + 2q + 2$ variables $(x, y, z, \tau, s, \kappa)$.

At each iteration, we compute 1-4 search directions from a linear system with fixed LHS and different RHS vectors.

The LHS is a square, sparse structured matrix of 6×6 blocks, containing Hessian evaluations and the fixed affine data c, A, b, G, h.

Eliminating s, κ, τ yields a symmetric indefinite system in $n + p + q$ variables (x, y, z). We can use a sparse LDLT of the LHS:

$$\begin{bmatrix} 0 & A' & G' \\ A & 0 & 0 \\ G & 0 & -H^{-1} \end{bmatrix}$$

Most conic IPMs use this *Sym-Indef* method; good for sparse LP/SOCP.

Solving for directions

But for dense A, G or H, our *QR-Cholesky* method is often better.

- Use a precomputed QR factorization of A' to eliminate the p primal equalities (and y), reducing x to dimension $n - p$.
- Eliminate z to get a positive definite system in x.
- Use a dense Cholesky of side dimension $n - p$.

Hypatia's directions solver interface allows plugging in custom linear system techniques that leverage formulation-specific structure.

Unlike *Sym-Indef*, *QR-Cholesky* only needs (inverse) Hessian products.

The CP proximity checks only use inverse Hessian vector products.

Hence with new oracles that apply the (inverse) Hessian to arrays implicitly, we have no need to form and factorize explicit Hessians.

These oracles are known for symmetric cones, by Jordan algebra theory.

Chapter 2: continuous formulations and IPM oracles

Natural conic formulations

Advanced solvers such as MOSEK 9 only recognize the *standard cones*:

- symmetric: nonnegative, real PSD, second order ($u \geq \|w\|$),
- 3-D nonsymmetric: exponential, power.

Many convex problems can be represented through conic *extended formulations* (EFs) using only the standard cones.

But *natural formulations* (NFs) using exotic cones are usually simpler.

We define three classes of Hypatia's cones with LHSCB oracles:

- *PSD slice cones*: intersections of slices of the PSD cone,
- *spectral norm cones*: epigraphs of infinity norms or spectral norms,
- *spectral function cones*: epigraphs/hypographs of spectral functions.

For these cones and their dual cones, the best EFs have larger dimensions (n, p, q) and often larger barrier parameters (ν) than the NFs.

Examples and IPM oracles

In Chapters 2 and 3, we formulate 11 applied problems over these classes.

E.g. portfolio opt., experiment design, distribution estimation, matrix completion, multi-response regression, convex regression, classical-quantum channel capacity, polynomial minimization.

For each example, over a wide range of sizes, we find that Hypatia solves the NFs more efficiently than both Hypatia and MOSEK 9 solve the EFs.

The NFs are also simpler to write and interpret conic certificates for, and more efficient to build using JuMP.

What makes this possible? Efficient, numerically-stable oracles.

- Chapter 2: oracles for PSD-slice and spectral norm cones.
- Chapter 3: new LHSCBs and oracles for spectral function cones.

Spectral and nuclear norm cones

We vectorize $W \in \mathbb{R}^{d \times s}$ as $w = \text{vec}(W) \in \mathbb{R}^{ds}$ by stacking columns.

Suppose $d \leq s$. The spectral/nuclear norm epigraph cones are:

$$\mathcal{K}_{\ell_{\text{spec}}} = \big\{(u, w) : u \geq \sigma_1(W)\big\},$$
$$\mathcal{K}^*_{\ell_{\text{spec}}} = \big\{(u, w) : u \geq \textstyle\sum_{i \in [\![r]\!]} \sigma_i(W)\big\}.$$

The EF for $(u, w) \in \mathcal{K}^*_{\ell_{\text{spec}}} \subset \mathbb{R}^{1+ds}$ uses a linear and a PSD constraint:

$$\exists \Theta \in \mathbb{S}^d, \Lambda \in \mathbb{S}^s, \quad u \geq (\text{tr}(\Theta) + \text{tr}(\Lambda))/2, \quad \left[\begin{smallmatrix} \Theta & W \\ W' & \Lambda \end{smallmatrix}\right] \succeq 0.$$

For $\mathbb{S}^d$ the vectorized triangle length is $\text{sd}(d) = d(d+1)/2$. The EF has:

- $\text{sd}(d) + \text{sd}(s)$ auxiliary variables,
- conic dimension of $1 + \text{sd}(d + s)$: larger than $1 + ds$,
- barrier parameter ν of $1 + d + s$: larger than $1 + d$ (for $\mathcal{K}_{\ell_{\text{spec}}}$).

Matrix (multi-response) regression example

We estimate a coefficient matrix F given a design matrix X and a response matrix Y:

$$\min \quad \|Y - FX\|_{\text{nuc}} + \lambda\|F\|_2.$$

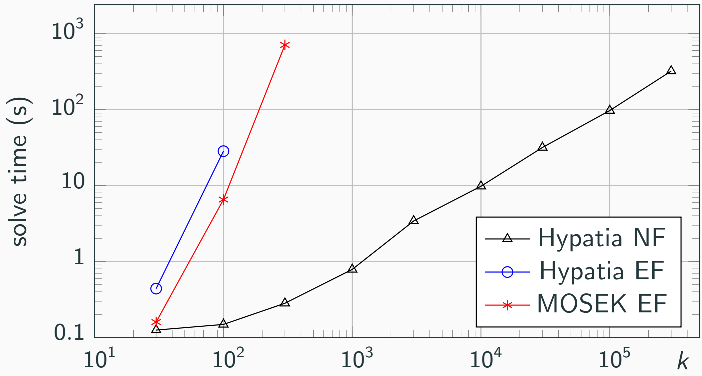
A natural conic formulation over ρ, μ, F is:

$$\begin{aligned} \min \quad \rho + \lambda\mu : \\ (\rho, \text{vec}(Y - FX)) \in \mathcal{K}^*_{\ell_{\text{spec}}}, \\ (\mu, \text{vec}(F)) \in \mathcal{K}_{\ell_2}. \end{aligned}$$

We generate random instances with $\lambda = 0.1$, where:

- F is 15×15,
- $Y - FX$ is $15 \times k$, for varying $k \geq 15$.

Solving matrix regression



Hypatia-NF consistently takes 9 or 10 IPM iterations.

At $k = 3 \times 10^5$, $\mathcal{K}^*_{\ell_{\text{spec}}}$ has dimension $1 + 15 \times k \approx 4.5 \times 10^6$.

At $k = 10^6$ we ran out of RAM in preprocessing, but we can turn this off.

New spectral norm cone oracles

We do not have an LHSCB for $\mathcal{K}^*_{\ell_{\text{spec}}}$ with analytic/closed-form oracles.

Nesterov and Nemirovskii (1994) give a $(1+d)$-LHSCB for $\mathcal{K}_{\ell_{\text{spec}}}$:

$$\Gamma(u, w) = -\log(u) - \operatorname{logdet}(uI(d) - WW'/u).$$

The vector infinity norm cone and symmetric spectral norm cone are slices of $\mathcal{K}_{\ell_{\text{spec}}}$, so the same LHSCB applies. We specialize the oracles.

Our oracles depend on a thin singular value decomposition (SVD) of W. The implementations are optimized for speed, memory, and numerics.

We derive a formula for the inverse Hessian product, i.e. $z \to (\nabla^2 \Gamma)^{-1} z$.

- Uses $\mathcal{O}(ds)$ memory and $\mathcal{O}(d^2 s)$ time.

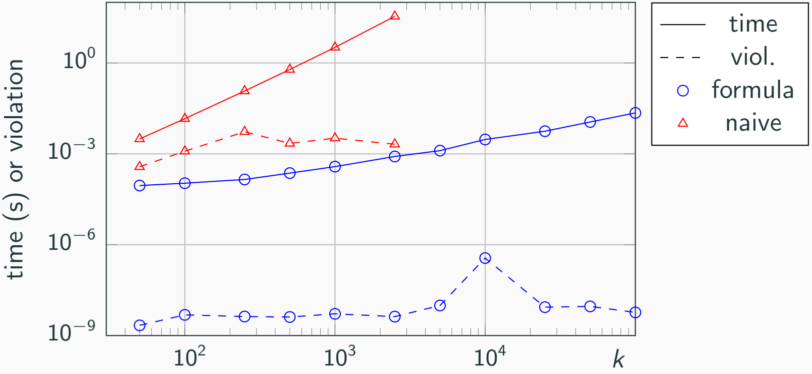
The naive alternative Cholesky-factorizes the explicit Hessian in $\mathbb{S}^{1+ds}$.

- Uses $\mathcal{O}(d^2 s^2)$ memory and $\mathcal{O}(d^3 s^3)$ time.

Comparing the inverse Hessian product procedures

Our formula is much more efficient and numerically stable in practice.

- We solve NF instances for matrix regression using Hypatia.
- At the final PDIPM iterate, we compute $z = (\nabla^2 \Gamma)^{-1}(\nabla \Gamma)$.
- We compute the LH condition violation $|1 - \nu^{-1} z'(\nabla \Gamma)|$.



Chapter 3: barriers and oracles for spectral function cones

Spectral functions

We often encounter symmetric functions of real vectors or eigenvalues of symmetric/Hermitian matrices, i.e. spectral functions on Jordan algebras.

Example: experiment design (Boyd and Vandenberghe, 2004).

- Variable $\mu \in \mathcal{C} \subset \mathbb{R}^m$ is the number of trials for each of m experiments, and $\mathcal{C}$ expresses e.g. nonnegativity, budget.
- $F \in \mathbb{R}^{k \times m}$ is a menu of experiments for estimating a vector in $\mathbb{R}^k$.

Let $W = F \operatorname{Diag}(\mu) F' \in \mathbb{S}^k$. We want a 'small' error covariance W^{-1}.

D-optimal: $\min \operatorname{logdet}(W^{-1}) = -\operatorname{logdet}(W) = -\sum_i \log(\lambda_i(W))$.
A-optimal: $\min \operatorname{tr}(W^{-1}) = \sum_i (\lambda_i(W))^{-1}$.

For D-optimal, let $\mathcal{K}$ be the epigraph of the perspective of $-\operatorname{logdet}$.

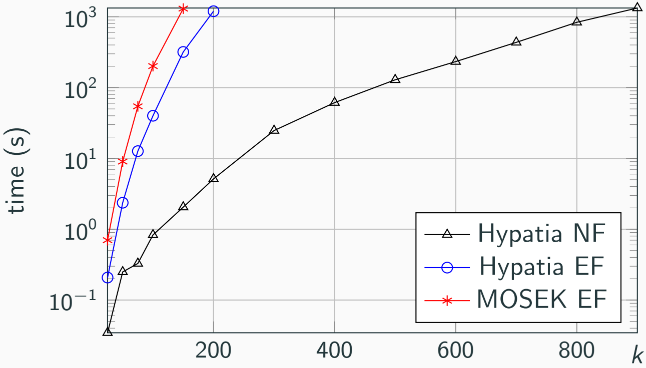
$$\min \quad \psi : \quad \mu \in \mathcal{C}, \quad (\psi, 1, \operatorname{vec}(F \operatorname{Diag}(\mu) F')) \in \mathcal{K} \subset \mathbb{R}^{2+\operatorname{sd}(k)}.$$

Solving D-optimal design

Let $w = \text{vec}(W)$. An EF for $(u, v, w) \in \mathcal{K}$, using exponential/PSD cones:

$$\exists \pi \in \mathbb{R}^k, \Theta \in \mathbb{S}^k, \quad u \geq e'\pi, \quad (\pi_i, v, \Theta_{i,i}) \in \mathcal{K}, \forall i, \quad \begin{bmatrix} W & \Theta \\ \Theta' & \text{Diag}(\Theta) \end{bmatrix} \succeq 0.$$

We let $m = 2k$ and vary k. Recall $\mu \in \mathbb{R}^m$ and $W = F\,\text{Diag}(\mu)F' \in \mathbb{S}^k$.



A class of spectral function cones

Let V be a Jordan algebra of rank d and let $\mathcal{Q}$ be its cone of squares, e.g. for $V = \mathbb{R}^d, \mathcal{Q} = \mathbb{R}^d_{\geq}$ and for $V = \mathbb{S}^d, \mathcal{Q} = \mathbb{S}^d_{\succeq}$.

Let h be a separable spectral function on $\operatorname{int} \mathcal{Q}$. If $\lambda \in \mathbb{R}^d_{>}$ are the eigenvalues of $w \in \operatorname{int} \mathcal{Q}$, then $\operatorname{tr} h(w) = h(\lambda) = \sum_i h(\lambda_i)$.

Suppose h has the *matrix monotone* derivative (MMD) property: $w_1 \succeq w_2 \succeq 0 \Rightarrow h'(w_1) \succeq h'(w_2)$. This implies h is convex.

The MMD cone is the epigraph of the perspective of $\operatorname{tr} h$:

$$\mathcal{K}_{\text{MMD}} = \operatorname{cl}\left\{(u, v, w) \in \mathbb{R} \times \mathbb{R}_{>} \times \operatorname{int} \mathcal{Q} : u \geq v \operatorname{tr} h(w/v)\right\}.$$

We show $\mathcal{K}_{\text{MMD}}$ has an LHSCB with near-optimal $\nu = 2 + d$:

$$\Gamma(u, v, w) = -\log\big(u - v \operatorname{tr} h(w/v)\big) - \log(v) - \operatorname{logdet}(w).$$

Our proof is via domain compatibility (Nesterov and Nemirovskii, 1994). We use the matrix monotone integral representation of Löwner (1934).

Modeling with MMD cones

The convex conjugate of h is h^*, also a spectral function but not MMD.

$$\mathcal{K}^*_{\text{MMD}} = \text{cl}\left\{(u, v, w) \in \mathbb{R}_> \times \mathbb{R} \times \mathcal{R} : v \geq u \operatorname{tr} h^*(w/u)\right\}.$$

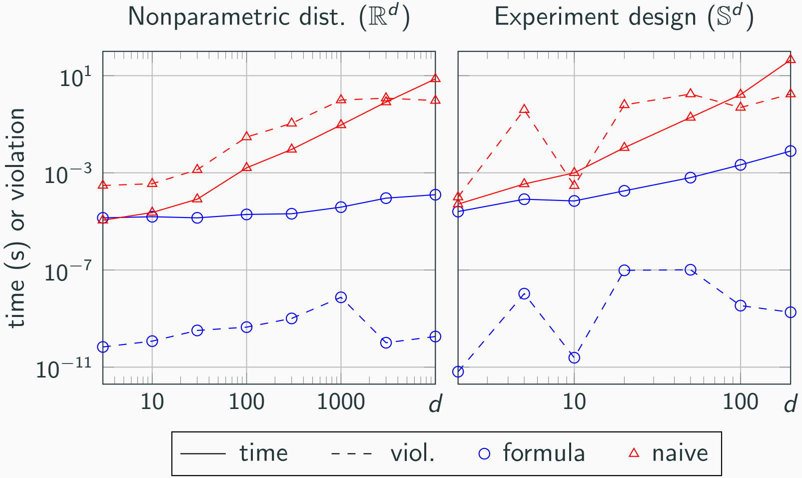
	h	h'	$\mathcal{R}$	h^*
NegLog	$-\log(x)$	$-x^{-1}$	$\mathcal{Q}$	$-1-\log(x)$
NegEntropy	$x\log(x)$	$1+\log(x)$	V	$\exp(-1-x)$
NegSqrt	$-\sqrt{x}$	$-\frac{1}{2}x^{-1/2}$	$\mathcal{Q}$	$\frac{1}{4}x^{-1}$
NegPower, $p \in (0,1)$	$-x^p$	$-px^{p-1}$	$\mathcal{Q}$	$-(p-1)(x/p)^q$
Power, $p \in (1,2]$	x^p	px^{p-1}	V	$(p-1)(x_-/p)^q$

We predefine these MMD functions in Hypatia through simple univariate oracles, from which all our efficient/stable $\mathcal{K}_{\text{MMD}}$ oracles are derived.

$\mathcal{K}_{\text{MMD}}$ and $\mathcal{K}^*_{\text{MMD}}$ allow representing many common disciplined convex programming (DCP) atoms conically, at minimal cost in dimension or ν.

Comparing the inverse Hessian product procedures

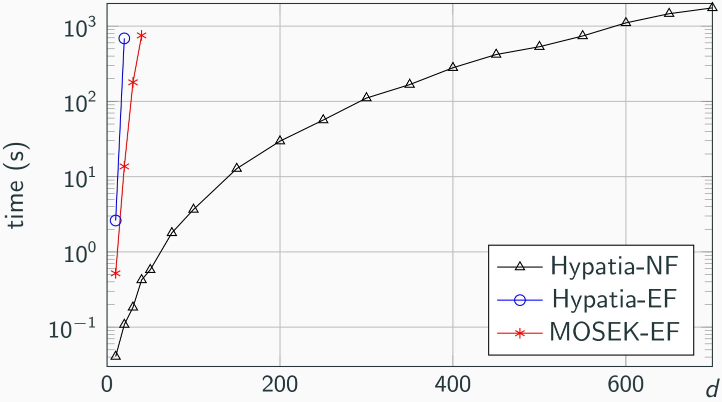
We run the experiment we used for $\mathcal{K}_{\ell_{\text{spec}}}$, on two *NegEntropy* examples.



Classical-quantum channel capacity example

The variable $\rho \in \mathbb{R}^d$ is a probability distribution, and each $P_i \in \mathbb{H}^d_{\succeq}$ is a density matrix (Sutter et al., 2015). φ is the trace of *NegEntropy* on $\mathbb{H}^d_{\succeq}$.

$$\min \quad \varphi\left(\textstyle\sum_{i \in [\![d]\!]} \rho_i P_i\right) - \textstyle\sum_{i \in [\![d]\!]} \rho_i \varphi(P_i) : \quad e'\rho = 1, \quad \rho \geq 0.$$



Chapters 4-5: mixed integer conic optimization

Chapter 4: OAMs and Pajarito

Branch-and-bound with polyhedral relaxations

B&B algorithms recursively partition the possible values of the integer variables, using bounds/solutions from convex subproblems.

MILP solvers use advanced cutting plane techniques and heuristics, and LP Simplex solvers can rapidly reoptimize after cuts are added.

An OAM solves a polyhedral relaxation of the subproblem at each node, so the convex subproblem can be solved less frequently.

Hypothetical: you have an MILP that Gurobi solves well, but now you need to add a convex constraint. Extend Gurobi to an OAM.

Bonami, Kılınç, and Linderoth (2012): Bonmin's OAM beats plain B&B.

But Bonmin uses NLP solvers and gradient cuts, which need smoothness.

Conic solvers handle nonsmoothness, and good MI convex formulations are often conic anyway: e.g. ideal formulation for union of convex sets.

A conic-duality-based OAM

Since $\mathcal{K} = \{s : z's \geq 0, \forall z \in \mathcal{K}^*\}$, we have:

$$h - Gx \in \mathcal{K} \quad \Leftrightarrow \quad z'(h - Gx) \geq 0 \quad \forall z \in \mathcal{K}^*$$

Any finite subset of $\mathcal{K}^*$ cuts gives a polyhedral relaxation.

We present the first conic-duality-based B&B OA algorithm.

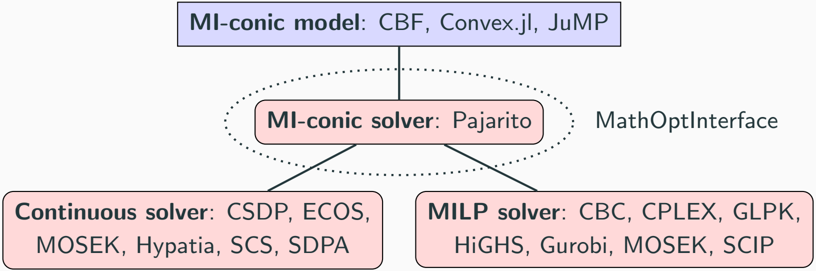
- $\mathcal{K}^*$ cuts from dual solutions/rays for continuous conic subproblems.
- If no strong duality failures, it detects infeasibility, unboundedness, or returns an optimal solution in finite time.

For cone $\mathcal{K}$, we use the following $\mathcal{K}^*$ cuts oracles:

- *initial cuts*: builds a fixed initial polyhedral relaxation of $\mathcal{K}$,
- *subproblem cuts*: given $z \in \mathcal{K}^*$, adds cuts at least as strong as z,
- *separation cuts*: given a point s, checks whether $s \in \mathcal{K}$ (approximately) and if not, adds cuts that separate s.

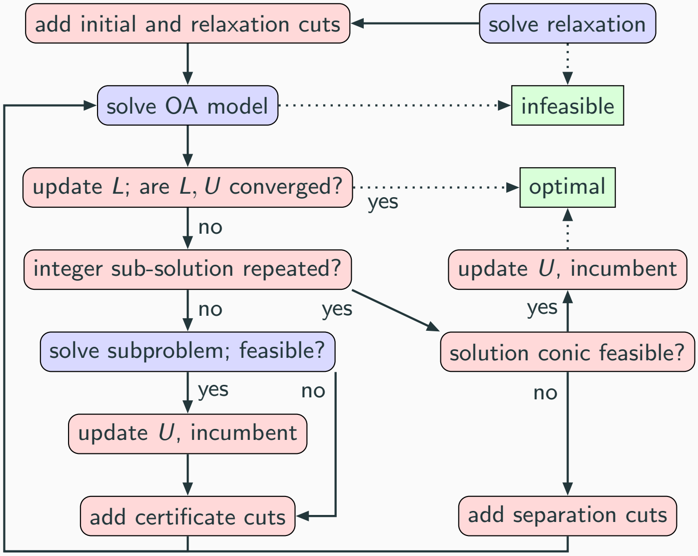
Pajarito's software architecture

To leverage powerful external MILP solvers through a solver independent interface, our practical implementations in Pajarito differ from our idealized B&B algorithm.

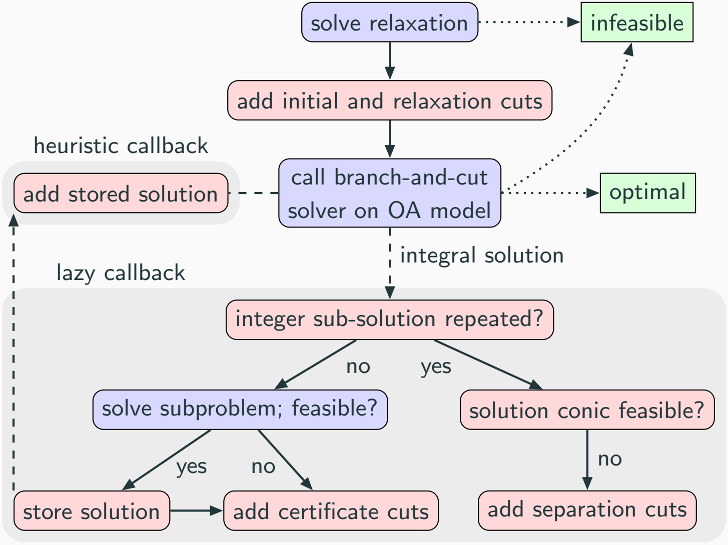


In 2018, benchmarking on 120 MISOCPs: Pajarito greatly outperformed Bonmin, and converged more reliably than CPLEX (in similar time).

Iterative algorithm



MIP-solver-driven (single tree) algorithm



Chapter 5: mixed integer formulations and OAM oracles

MI conic formulations for OAMs

We implement $\mathcal{K}^*$ cut oracles for the PSD slice, spectral norm, and spectral function cones through Pajarito's generic cone interface.

We formulate a dozen MI-conic examples, including sparse inverse covariance estimation, experiment design, matrix completion, polynomial regression, modular device design, and some new OR-flavor problems.

We compare OAM performance under NFs and EFs.

- We use Pajarito's iterative algorithm, with Hypatia and Gurobi.
- OA iterations measure the strength/quality of polyhedral relaxations.

For the vector domain spectral function cones, we have EFs with 3-D cones. These improve iteration counts by tightening the relaxations.

For the other cones, EFs are generally slow and numerically unstable.

Primal PSD slice cones

A PSD slice cone is an intersection of slices of real/complex PSD cones:

$$\mathcal{K} = \{s \in \mathbb{R}^d : \Lambda_l(s) \succeq 0, \forall l \in [\![r]\!]\},$$

where $\Lambda_l : \mathbb{R}^d \to \mathbb{S}^{o_l}, \forall l \in [\![r]\!]$ are linear operators. The EF is obvious.

E.g. PSD, sparse PSD, and dual polynomial sum of squares (SOS) cones.

Consider a PSD slice cone constraint $s \in \mathcal{K}$.

- For initial cuts, we impose $\text{diag}(\Lambda_l(s)) \geq 0$.
- To separate a point $\bar{s} \in \mathbb{R}^d$, we compute eigendecompositions:

  $$\Lambda_l(\bar{s}) = \textstyle\sum_{i \in [\![o_l]\!]} \sigma_{l,i} v_{l,i} v_{l,i}',$$

  and add one separation cut for each negative eigenvalue:

  $$\langle v_{l,i} v_{l,i}', \Lambda_l(s) \rangle \geq 0 \qquad \forall i \in [\![o_l]\!] : \sigma_{l,i} < 0.$$

- For certain $\mathcal{K}$, we can decompose a $z \in \mathcal{K}^*$ into extreme rays of $\mathcal{K}^*$.

Dual cones of PSD slice cones

Oracles for $\mathcal{K}^*$ are more ad-hoc than those for $\mathcal{K}$.

Let $\Lambda_l^* : \mathbb{S}^{o_l} \to \mathbb{R}^d$ be the adjoint of Λ_l, $\forall l \in [\![r]\!]$. Then:

$$\mathcal{K}^* = \big\{ s \in \mathbb{R}^d : \exists S_1, \dots, S_r \succeq 0, s = \textstyle\sum_{l \in [\![r]\!]} \Lambda_l^*(S_l) \big\}.$$

An EF for $\mathcal{K}^*$ often requires many auxiliary variables, for example:

- dual sparse PSD cones, which are PSD-completable matrix cones,
- primal SOS cones, which allow natural MI polynomial models.

If $\mathcal{K}$ is a dual SOS cone parametrized by $P_l \in \mathbb{R}^{d \times o_l}$, $\forall l \in [\![r]\!]$:

$$\Lambda_l(s) = P_l' \operatorname{Diag}(s) P_l, \qquad \Lambda_l^*(S_l) = \operatorname{diag}\big(P_l \Theta_l P_l'\big).$$

In this case, $s \geq 0$ is a simple and strong set of initial fixed cuts.

Optimization-based separation

For a primal SOS cone, we don't know an analytic procedure for checking feasibility of a point and obtaining separation cuts.

Let $\bar{s}$ be the point and $\mathcal{C}$ be the proper cone. We solve a conic problem:

$$\text{(P)} \quad \min \ 0 : \quad \bar{s} \in \mathcal{C} \qquad\qquad \text{(D)} \quad \max_z \ -\bar{s}'z : \quad z \in \mathcal{C}^*$$

P has an optimal solution if and only if $\bar{s}$ is feasible.

If $\bar{s}$ is infeasible, D has an improving ray $z \in \mathcal{C}^*$ with $\bar{s}'z < 0$. So z provides a separation cut.

Hypatia is ideal for solving this separation problem.

- Using QR-Cholesky, no factorizations are needed (besides Hessians).
- We only need one separation model for each unique cone.
- We just modify $\bar{s}$ and re-solve without overhead/allocations.

Polynomial facility location example

We have a capacitated facility location problem with nonnegative polynomial flows over continuous time, from $t = 0$ to $t = 1$:

- facility $i \in [\![n]\!]$ has fixed cost f_i, maximum output rate u_i, and is opened if binary variable $x_i = 1$,
- customer $j \in [\![m]\!]$ has demand rate d_j,
- the cost per unit of flow from i to j is $c_{i,j}$,
- the degree-d polynomial variable $y_{i,j} \in \mathbb{R}_{1,d}[t]$ is the flow rate from i to j over $t \in [0, 1]$,
- we minimize total cost, satisfying capacities and demands.

We formulate the high-level MI polynomial model before converting to an equivalent MI-conic problem over SOS cones.

Formulation for polynomial facility location

$$\begin{aligned} \min \quad & f'x + \textstyle\sum_{i,j} c_{i,j} \int_0^1 y_{i,j}(t)\,\mathrm{d}t : \\ & \textstyle\sum_j y_{i,j}(t) \leq u_i x_i \qquad \forall i \in [\![n]\!], t \in [0,1], \\ & \textstyle\sum_i y_{i,j}(t) \geq d_j \qquad \forall j \in [\![m]\!], t \in [0,1], \\ & y_{i,j}(t) \geq 0 \qquad \forall i \in [\![n]\!], j \in [\![m]\!], t \in [0,1], \\ & x \in \{0,1\}^n. \end{aligned}$$

We reinterpret the polynomial variables in the interpolant basis as $y_{i,j} \in \mathbb{R}^{1+d}$ and linearize the integral term. The MI-conic model is:

$$\begin{aligned} \min \quad & f'x + \textstyle\sum_{i,j} c_{i,j} w' y_{i,j} : \\ & u_i x_i e - \textstyle\sum_j y_{i,j} \in \mathcal{K}_{\mathsf{SOS}(P)} \qquad \forall i \in [\![n]\!], \\ & \textstyle\sum_i y_{i,j} - d_j e \in \mathcal{K}_{\mathsf{SOS}(P)} \qquad \forall j \in [\![m]\!], \\ & y_{i,j} \in \mathcal{K}_{\mathsf{SOS}(P)} \qquad \forall i \in [\![n]\!], j \in [\![m]\!], \\ & x \in \{0,1\}^n. \end{aligned}$$

Solving polynomial facility location

We generate feasible/bounded instances with degree $d = 6$, varying the number of facilities n and letting the number of customers be $m = 2n$.

	NF (SOS)			EF (PSD)		
n	st	it	time	st	it	time
5	co	1	0.2	er	1	2.0
10	co	2	1.2	er	1	19
15	co	2	6.1	tl	4	600
20	co	2	20	tl	3	602
25	co	3	51	tl	2	611
30	co	3	125	tl	0	614
35	co	2	127	tl	0	603
40	co	3	332	tl	0	601

Polynomial two-stage stochastic problem

We have n different crops and m equal-sized plots of land for planting:

- per plot, crop i has fixed cost a_i and uncertain yield $\xi_i \sim U(0,1)$,
- in stage 1, we decide how many plots of crop i to plant,
- in stage 2, we harvest the ξ_i units,
- we fulfill our contractual demand for d_i units,
- we decide how much crop i to buy/sell for prices b_i/c_i,
- we minimize the total expected cost.

Stage 2 decomposes by crop. So for crop i, the buy/sell decisions depend only on ξ_i.

We represent these with polynomial variables $y_i, z_i \in \mathbb{R}_{1,2k}[\xi_i]$, which must nonnegative over $\xi_i \in [0,1]$.

Polynomial two-stage stochastic formulation

We first write a high-level stochastic MI polynomial model.

$$
\begin{aligned}
\min \quad & a'x + \textstyle\sum_i \mathbb{E}_{\xi_i}[b_i y_i(\xi_i) - c_i z_i(\xi_i)] : \\
& e'x \leq m, \\
& x \geq 0, \\
& \xi_i x_i + y_i(\xi_i) - z_i(\xi_i) - d_i = 0 \qquad \forall i \in [\![n]\!], \xi_i \in [0,1], \\
& y_i(\xi_i) \geq 0 \qquad \forall i \in [\![n]\!], \xi_i \in [0,1], \\
& z_i(\xi_i) \geq 0 \qquad \forall i \in [\![n]\!], \xi_i \in [0,1], \\
& x \in \mathbb{Z}^n.
\end{aligned}
$$

We write the expectations as integrals, then use quadrature to linearize.

The polynomial equality constraint is linear, and the $2n$ polynomial inequalities are written with $\mathcal{K}_{\mathrm{SOS}(P)}$ constraints.

Solving the polynomial two-stage stochastic problem

We generate instances with $n = 3$ crops, varying the half-degree k.

The objective values decrease with k, exhibiting diminishing returns.

		NF (SOS)			EF (PSD)		
k	obj	st	it	time	st	it	time
8	16.73477	co	2	0.1	er	2	0.7
16	16.70516	co	2	0.2	er	2	4.1
32	16.69616	co	2	0.5	er	1	67
64	16.69364	co	2	1.6	tl	0	868
128	16.69301	co	2	6.8	*	*	*
256	16.69284	co	1	33	*	*	*
512	16.69280	co	1	234	*	*	*

In future, Pajarito could decompose the conic subproblems by crop.

EFs for vector spectral function cones

We support Hypatia's spectral function cones, e.g. the geomean cone:

$$\mathcal{K}_{\text{geo}} = \left\{ (u, w) \in \mathbb{R} \times \mathbb{R}^d_{\geq} : u \leq \text{geo}(w) \right\}.$$

An EF for $(u, w) \in \mathcal{K}_{\text{geo}}$ in terms of d 3-D exponential cones is:

$$\exists \theta \in \mathbb{R}, \lambda \in \mathbb{R}^d, \quad \theta \geq u, \quad e'\lambda \geq 0, \quad (\lambda_i, \theta, w_i) \in \mathcal{K}_{\text{exp}}, \forall i \in [\![d]\!].$$

Using the EF in the OA model, we can get tighter polyhedral relaxations.

- Think of Fourier-Motzkin: a polynomial number of EF cuts projects to an exponential number of NF cuts.
- E.g. the ℓ_1 ball needs 2^d cuts, but its LP EF only needs $1 + 2d$.

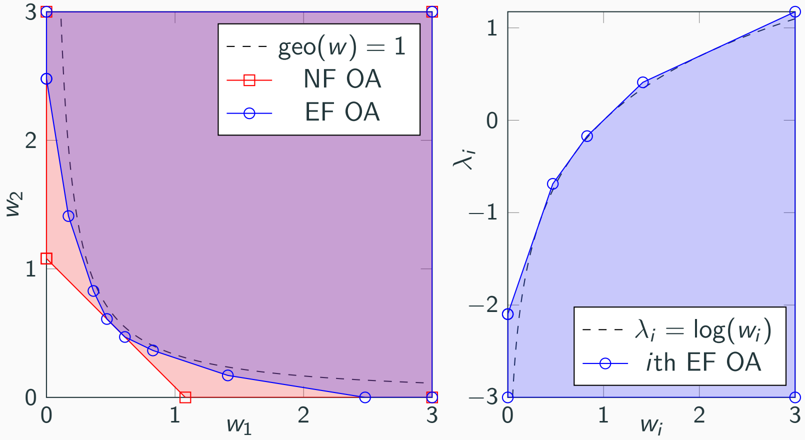
Hypatia can still use the NF: we 'extend' $\mathcal{K}^*$ cuts and primal solutions.

Generalizes Pajarito's SOCP EF approach, based on Vielma et al. (2017).

OA for a geomean ball constraint

For $w \in [0,3]^3$, we relax $\text{geo}(w) \geq 1$ with 7 NF $\mathcal{K}^*$ cuts. Note $\theta = u = 1$.

We project the 3-D NF and 6-D EF OAs onto w_1, w_2. The EF is tighter.



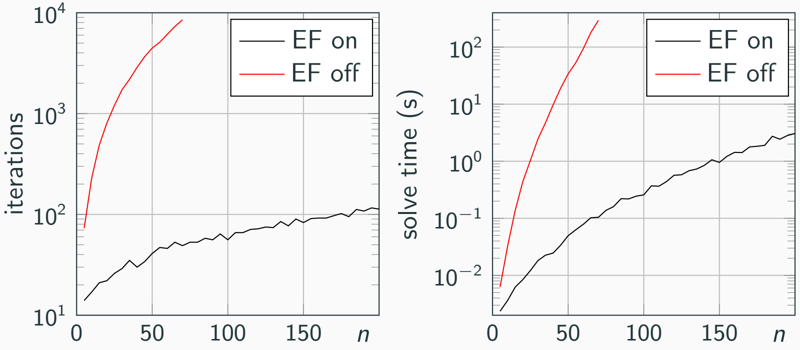
Knapsack problem with convex objective

We select $x_i \geq 1$ units of item $i \in [\![n]\!]$, with weight b_i and value c_i.

Given weight budget $B > 0$, we maximize the geomean of the values.

$$\max \quad \text{geo}\big((c_i x_i)_{i \in [\![n]\!]}\big) : \quad x \geq e, \quad b'x \leq B, \quad x \in \mathbb{Z}^n.$$

Results for the separation-only algorithm on the continuous relaxation.



Knapsack integer formulation results

	EF OA			NF OA		
n	st	it	time	st	it	time
3	co	3	0.0	co	5	0.0
6	co	8	0.1	co	34	0.7
9	co	6	17	co	71	165
12	co	9	22	tl	30	600
15	co	6	4.3	tl	253	600
20	co	7	13	tl	76	600
25	co	8	15	tl	109	600
30	co	9	21	tl	175	600
35	co	7	9.7	tl	543	600
40	co	11	3.1	$*$	$*$	$*$
50	co	8	9.5	$*$	$*$	$*$
60	co	8	14	$*$	$*$	$*$
70	co	7	13	$*$	$*$	$*$
80	co	8	9.5	$*$	$*$	$*$
90	co	12	57	$*$	$*$	$*$
100	co	10	409	$*$	$*$	$*$

Thoughts and observations

Choosing formulations and algorithmic options

Even just the modeling (before solving) can create bottlenecks.

In Hypatia, linear algebra is the most common bottleneck.

- Preprocessing, linear system solving, Hessian inversions.
- Heuristic: NFs are smaller/simpler.

For Pajarito, formulations matter for both OA and continuous solvers.

- Minimize the primal variable dimension n (e.g. MISOS examples).
- MILP solver likes sparsity and good numerics.
- If continuous solver is a bottleneck, try iterative or separation-only.
- Look for ideal (convex hull) formulations, e.g. *copies-of-variables*.
- Avoid subproblem strong duality failures: try to bound variables.

JuMP makes it easy to compare formulations and options.

Comparing very different algorithms is hard

We are concerned with general purpose (MI) conic algorithms. But we have not done extensive comparisons with non-IPMs and non-OAMs.

How do we set convergence tolerances and verify solutions?

- Algorithms use very different convergence measures.
- IPMs usually converge to tighter tolerances than 1st order methods.

No other solvers support as many cones as Hypatia/Pajarito.

How to we choose which equivalent conic formulations to run?

- Algorithms use different linear system routines.
- How do we compare solution quality across different formulations?
- EFs can speed up OAMs, but have no benefit for NL-B&B.

It seems we can only do fair comparisons within specific applications.

Thank you

Bonami, Pierre, Mustafa Kılınç, and Jeff Linderoth (2012). "Algorithms and Software for Convex Mixed Integer Nonlinear Programs". English. In: *Mixed Integer Nonlinear Programming*. Ed. by Jon Lee and Sven Leyffer. Vol. 154. The IMA Volumes in Mathematics and its Applications. Springer New York, pp. 1–39. ISBN: 978-1-4614-1926-6. DOI: 10.1007/978-1-4614-1927-3_1.

Boyd, Stephen and Lieven Vandenberghe (2004). *Convex optimization*. Cambridge University Press.

Löwner, Karl (1934). "Über monotone matrixfunktionen". In: *Mathematische Zeitschrift* 38.1, pp. 177–216. DOI: 10.1007/bf01170633.

Nesterov, Y. and A. Nemirovskii (1994). *Interior-point polynomial algorithms in convex programming*. Studies in Applied Mathematics. Society for Industrial and Applied Mathematics.

References ii

- Skajaa, Anders and Yinyu Ye (2015). "A homogeneous interior-point algorithm for nonsymmetric convex conic optimization". In: *Mathematical Programming* 150.2, pp. 391–422.
- Sutter, David et al. (2015). "Efficient approximation of quantum channel capacities". In: *IEEE Transactions on Information Theory* 62.1, pp. 578–598. DOI: 10.1109/tit.2015.2503755. eprint: 1407.8202.
- Vielma, Juan Pablo et al. (Sept. 2017). "Extended formulations in mixed integer conic quadratic programming". In: *Mathematical Programming Computation* 9.3, pp. 369–418. ISSN: 1867-2957. DOI: 10.1007/s12532-016-0113-y.