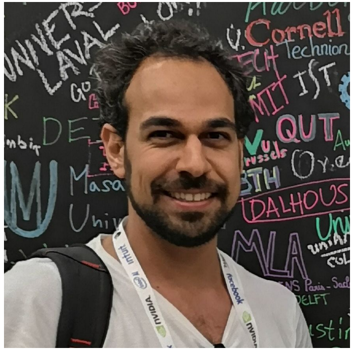
Scalable Semidefinite Programming

Volkan Cevher | OWOS



Alp Yurtsever

Olivier Fercoq

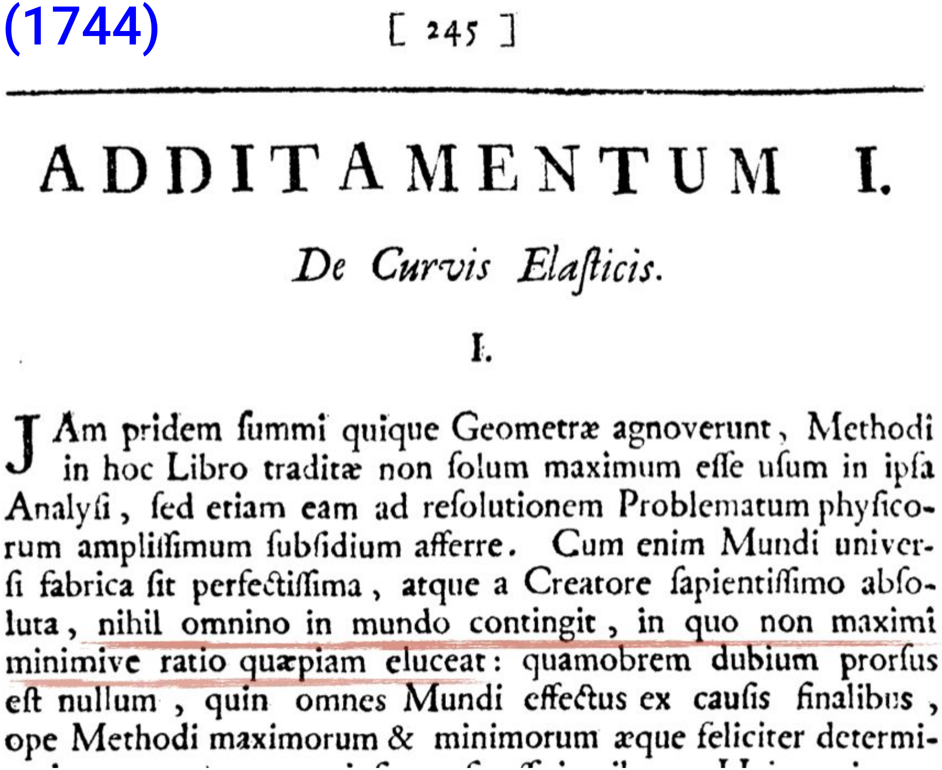
Joel Tropp

Madeleine Udell

Armin Eftekhari

Fatih Sahin

Leonhard Euler (1744)



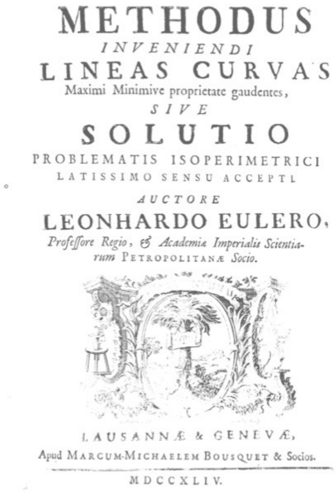
[245]

ADDITAMENTUM I.

De Curvis Elaſticis.

I.

JAm pridem ſummi quique Geometræ agnoverunt, Methodi in hoc Libro traditæ non ſolum maximum eſſe uſum in ipſa Analyſi, ſed etiam eam ad reſolutionem Problematum phyſicorum ampliſſimum ſubſidium afferre. Cum enim Mundi univerſi fabrica ſit perfectiſſima, atque a Creatore ſapientiſſimo abſoluta, nihil omnino in mundo contingit, in quo non maximi minimive ratio quæpiam eluceat: quamobrem dubium prorſus eſt nullum, quin omnes Mundi effectus ex cauſis finalibus, ope Methodi maximorum & minimorum æque feliciter determi-



METHODUS
INVENIENDI
LINEAS CURVAS
Maximi Minimive proprietate gaudentes,
SIVE
SOLUTIO
PROBLEMATIS ISOPERIMETRICI
LATISSIMO SENSU ACCEPTI
AUCTORE
LEONHARDO EULERO,
Profeſſore Regio, & Academiæ Imperialis Scientiarum PETROPOLITANÆ *Socio.*

LAUSANNÆ & GENEVÆ,
Apud MARCUM-MICHAELEM BOUSQUET & Socios.
MDCCXLIV.

'Nothing takes place in the world whose meaning is not that of some maximum or minimum.'

Technically speaking, what we will talk about today...

$$\min_{x \in \mathcal{D}} f(x) \quad \text{subject to} \quad Ax = b$$

cost | loss | objective function

domain

decision variable

$$f(x^\star) \leq f(x) \quad \text{for all } x \in \mathcal{D} \text{ such that } Ax = b$$

solution

“In general, optimization problems are unsolvable” Y. Nesterov

$$\min_{x \in \mathcal{D}} \; f(x) \quad \text{subject to} \quad Ax = b$$

ϵ-Approximate Solution

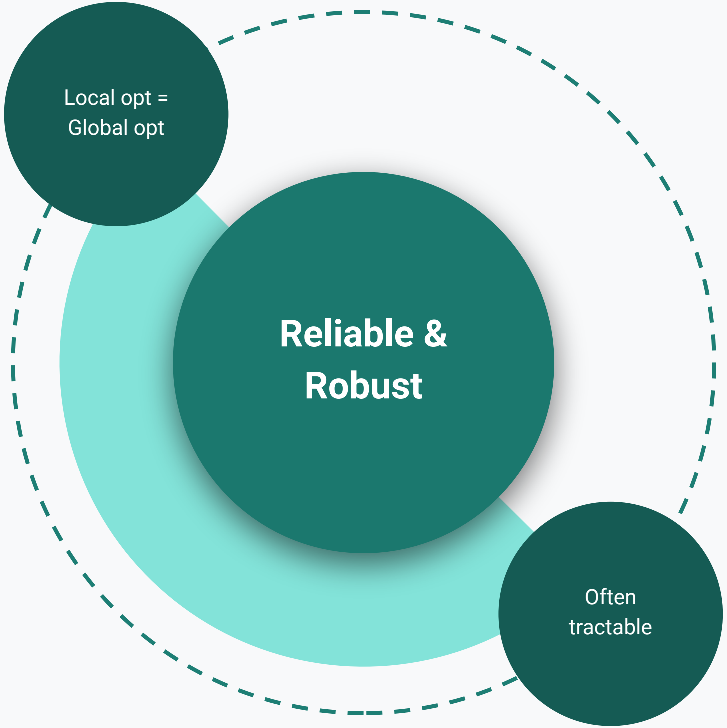
$$x_\epsilon \in \mathcal{D}, \quad \mathrm{f}(\mathrm{x}_\epsilon) - f(x^\star) \leq \epsilon, \quad \text{and} \quad \|Ax_\epsilon - b\| \leq \epsilon$$

Game of Trade-offs

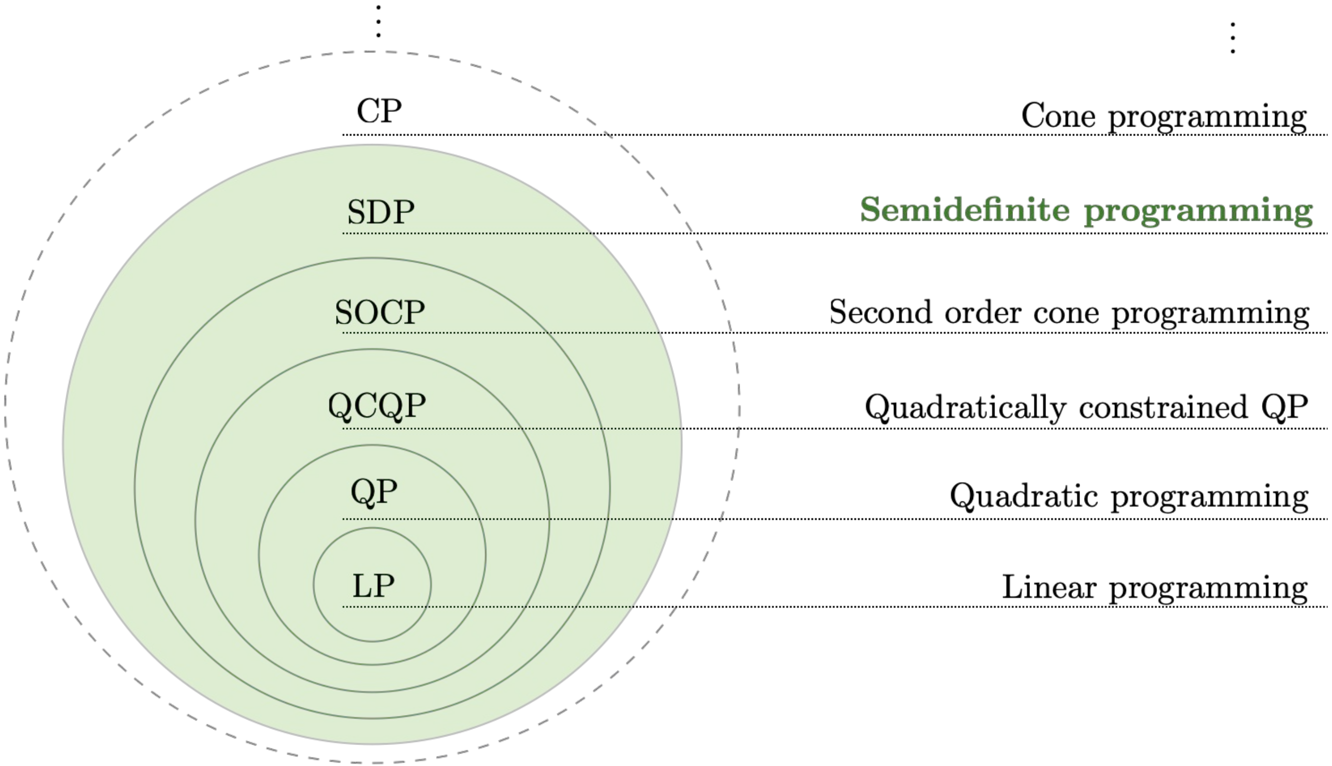
$$\text{Arithmetic Cost} = \sum_{k=1}^{\#\text{ iterations (to } \epsilon \text{ error)}} \text{Arithmetic Cost at iteration } k$$

$$\text{Storage Cost} = \max_{\text{over all iterations (to } \epsilon \text{ error)}} \text{Storage Cost at iteration } k$$

House Convex



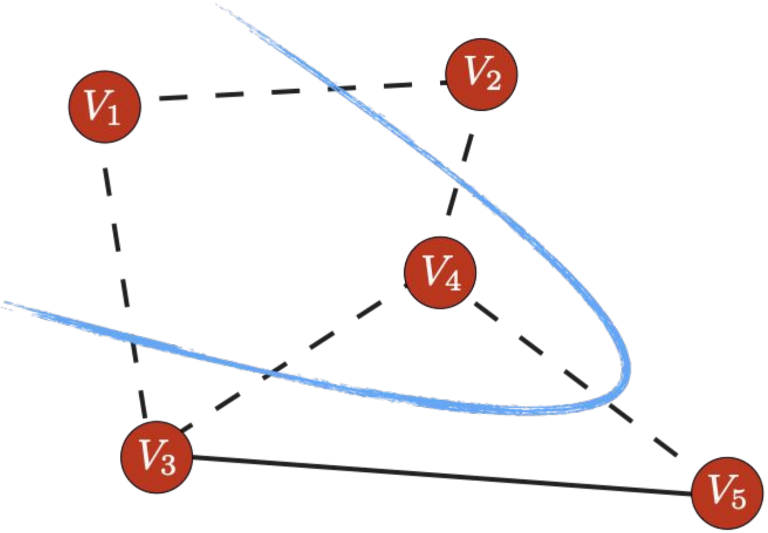
Hierarchies in House Convex



Semidefinite programming

$$\min_{X \in \mathbb{S}_+^n} \quad \langle C, X \rangle \quad \text{subj.to} \quad \underbrace{AX = b}_{A : \mathbb{S}_+^n \to \mathbb{R}^d}$$

Example: Max-cut



$$x = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$$

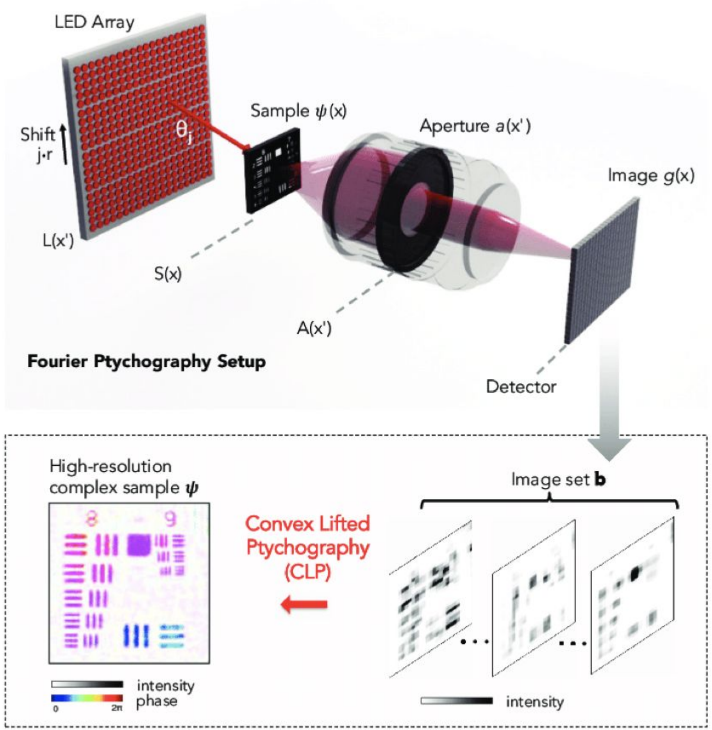
Roadmaps
n ~ 1 million <> 4TB

Social networks
n~ 1 billion <> 4PB

$$\min_{x} \; x^\top C x \;\; \text{s.t.} \;\; x_i \in \{-1, +1\} \qquad \triangleright \; \text{Tr}(x^\top C x) = \text{Tr}(x x^\top C) = \langle x x^\top, C \rangle$$

$$\min_{X \in \mathbb{S}^n_+} \; \langle X, C \rangle \;\; \text{s.t.} \;\; X_{ii} = 1, \; \cancel{\text{rank}(X) = 1}$$

Example: Fourier Ptychography



1 MPix

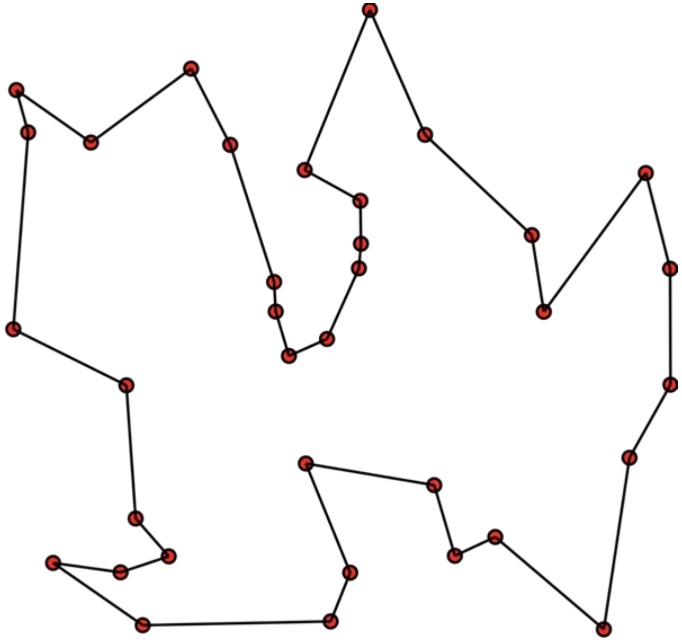
n ~ 1 million <> 4TB

$$
\begin{aligned}
b_i = |a_i^\top x|^2 &= x^\top a_i a_i^\top x \\
&= \mathrm{Tr}(x^\top a_i a_i^\top x) \\
&= \langle x x^\top, a_i a_i^\top \rangle \\
&= \langle X, a_i a_i^\top \rangle
\end{aligned}
$$

Many examples: Quadratic assignment, Lipschitz estimation of NNs...

$$n \xrightarrow{\text{SDP Relaxation}} n^2 \times n^2$$

$$\text{Difficult for } n \geq 100$$



Special case: Traveling salesman problem

From the archives…

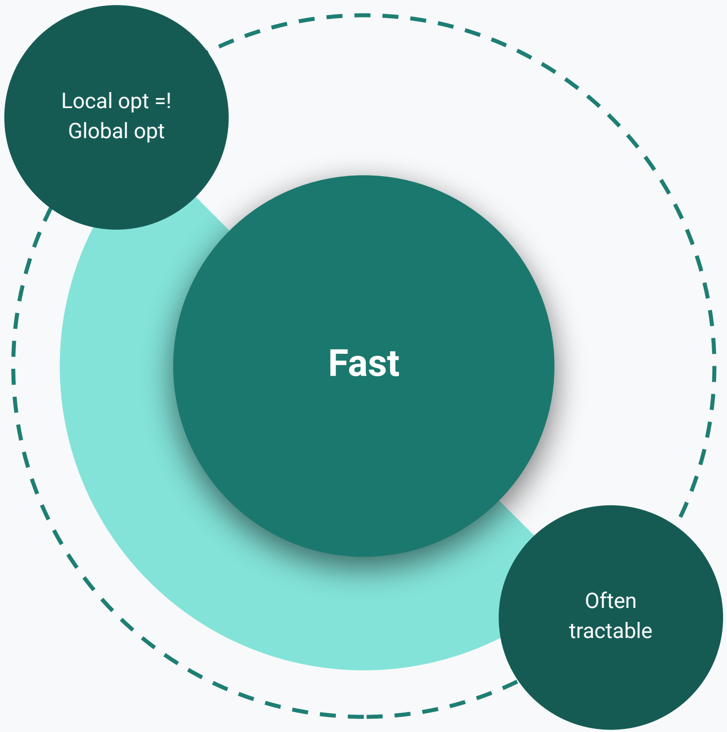
While in principle SDP based relaxations offer tractable solutions, they become computationally prohibitive as the dimension of the signal increases. Indeed, **for a large number of unknowns in the tens of thousands, say, the memory requirements are far out of reach of desktop computers** so that these SDP relaxations are de facto impractical.

"Phase Retrieval via Wirtinger Flow: Theory and Algorithms"
E. Candes, X. Li, M. Soltanolkotabi, 2015

Interior point methods solve (SDP) in polynomial time. In practice however, **for n beyond a few thousands, such algorithms run out of memory (and time)**, prompting research for alternative solvers.

"The non-convex Burer–Monteiro approach works on smooth semidefinite programs" N. Boumal, V. Voroninski, A.S. Bandeira, 2016

House Nonconvex



A key structure in weakly-constrained SDPs

$$\min_{X \in \mathbb{S}_+^n} \quad \langle C, X \rangle \quad \text{subj.to} \quad \underbrace{\mathcal{A}X = b}_{\mathcal{A} : \mathbb{S}_+^n \to \mathbb{R}^d}$$

Pataki(1998)-**Barvinok**(1995): $\text{rank}(X^\star) \leq \sqrt{2(d+1)}$

Optimization solution needs storage square-root of d times n vs n-squared!

State of the art

$$\text{Replace } X \text{ with } UU^\top: \quad \longrightarrow U \in \mathbb{R}^{n\times R}$$

$$\min_{U\in\mathbb{R}^{n\times R}} \langle C, UU^\top \rangle \quad \text{subj.to} \quad \mathcal{A}(UU^\top) = b$$

Barvinok (1995) “Problems of distance geometry and convex properties of quadratic maps”. **Pataki** (1998) “On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues”. **Burer, Monteiro** (2003) “A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization” | **Burer, Monteiro** (2005) “Local minima and convergence in low-rank semidefinite programming” | **Kulis, Surendran, Platt.** (2007) “Fast low-rank semidefinite programming for embedding and clustering” | (2012) **Cartis., Gould, Toint**, “Complexity bounds for second-order optimality in unconstrained optimization”| **Boumal, Voroninski, Bandeira.** (2016) “The non-convex Burer-Monteiro approach works on smooth semidefinite programs” | **Bhojanapalli et al.** (2018) “Smoothed analysis for low-rank solutions to semidefinite programs in quadratic penalty form” | **Pumir, Jelassi, Boumal.** (2018) “Smoothed analysis of the low-rank approach for smooth semidefinite programs” | **Sahin et al.** (2019) “An inexact augmented Lagrangian framework for non convex optimization with nonlinear constraints” | and **many more….**

The Lagrangian approach

$$\mathcal{L}_\beta(U,y) = \mathrm{tr}(CUU^\top) + \langle y, AUU^\top - b\rangle + \frac{1}{2\beta}\|AUU^\top - b\|^2.$$

- Burer-Monteiro's heuristic: $\begin{cases} u^+ = \arg\min_U \mathcal{L}_\beta(U,y) \\ \text{Update } y^+ \text{ or } \beta^+ \text{ according to feasibility progress} \end{cases}$
  - ▷ No inexact analysis for solving subproblems
  - ▷ Subproblem complexities e.g., $\begin{cases} \text{APGM (Ghadimi \& Lan, 2016): } \mathcal{O}(\frac{1}{\epsilon}) \\ \text{Trust region (Cartis et al., 2012): } \mathcal{O}(\frac{1}{\epsilon^3}) \end{cases}$
- Manifold optimization (ManOpt):
  - ▷ Smooth manifold assumption: Requires projectable sets
  - ▷ $\mathcal{O}\left(p^{10}/\epsilon^3\right)$ total complexity—$\mathcal{O}\left(p^6\right)$ flops per iteration

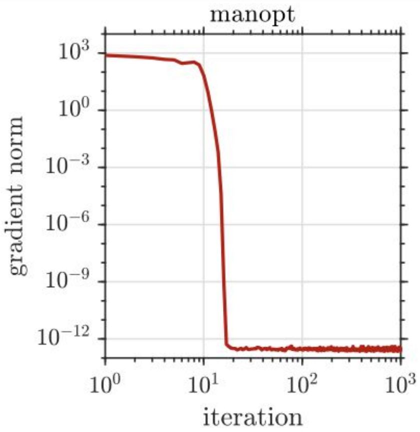
An inexact augmented Lagrangian framework [Sahin et al, NeurIPS'19]

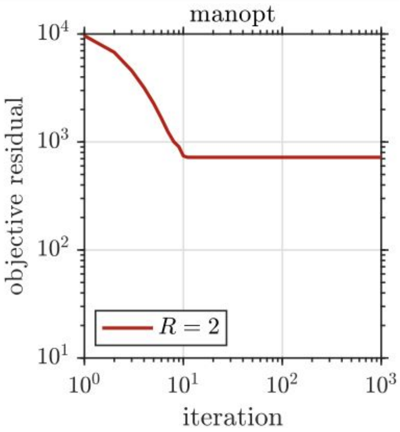
FOS with $\mathcal{O}\left(\frac{1}{\epsilon^3}\right)$ & SOS $\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^5}\right)$ total complexity

$$\text{iALM:} \begin{cases} \text{Obtain} & U^+ \text{ such that} \qquad \triangleright\ L(U) = AUU^\top \ \&\ g(U) = \mathrm{tr}(CUU^\top) \\ & \mathrm{dist}(-\nabla_U \mathcal{L}_\beta(U^+, y), \partial g(U^+)) \le \epsilon_f \text{ , or} \qquad \text{[1st order stationarity]} \\ & \lambda_{\min}(\nabla_{UU} \mathcal{L}_\beta(U^+, y)) \ge -\epsilon_s \qquad \text{[2nd order stationarity]} \\ y^+ = & y + \sigma\left(L(U^+) - b\right) \\ \text{Pick} & \beta^+ < \beta \text{ and } \epsilon^+ = \beta^+ \\ \text{Update} & \sigma^+ = \sigma_0 \min\left(\frac{1}{\|L(u)-b\| k \log^2(k+1)}, 1\right) \implies \text{Bounded dual} \end{cases}$$

Storage issues persists

n = 80



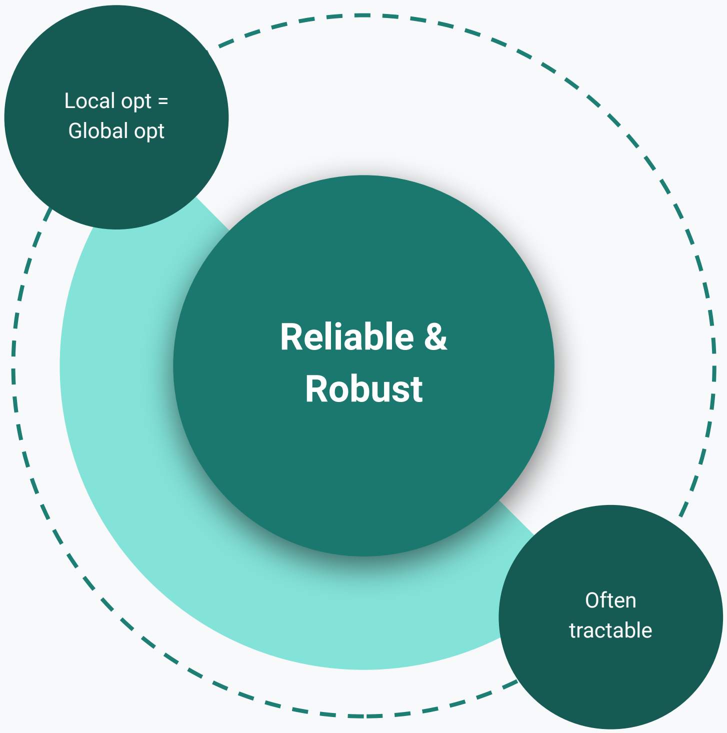


Dataset / R	$R=2$	3	4	5	6	7	8	9	10	11	12	13
C_1	82	69	63	53	35	32	24	12	11	1	4	0
C_2	77	56	56	36	19	17	12	2	0	0	0	0
C_3	89	65	54	47	44	46	23	11	5	0	3	0
C_4	84	69	50	40	27	23	18	17	1	0	9	0
C_5	85	68	52	51	43	30	31	20	14	3	4	0
C_6	81	68	53	41	23	22	10	10	2	0	1	0
C_7	83	76	60	39	19	19	19	3	0	0	1	0
C_8	81	73	44	34	41	25	8	12	5	4	10	0
C_9	84	64	46	35	25	17	1	10	0	2	4	0
C_{10}	83	71	54	50	31	25	24	16	13	0	8	0

Waldspurger, Waters (2019) “Rank optimality for the Burer-Monteiro factorization”

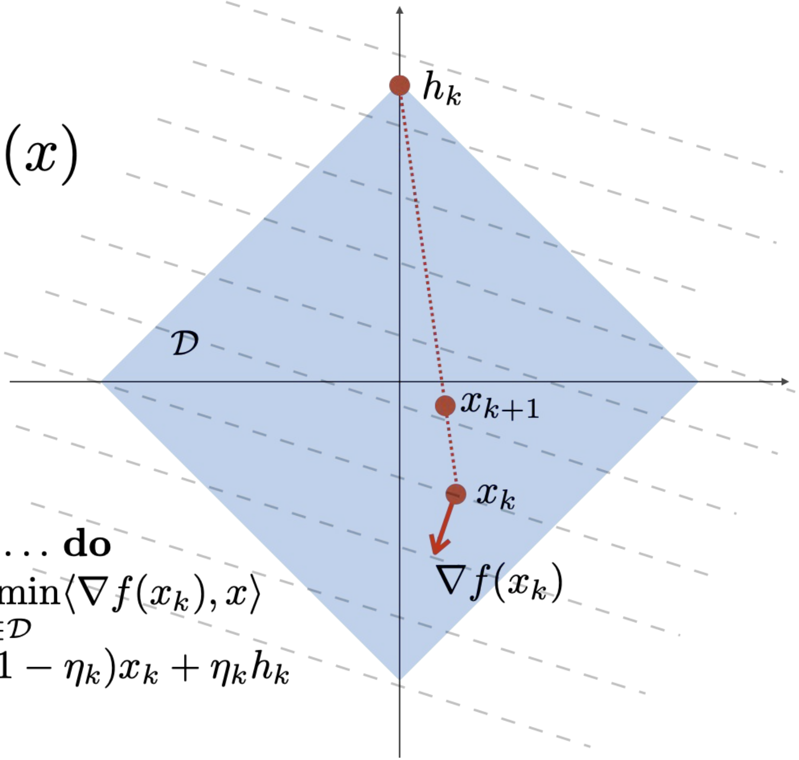
Challenge: Solve SDPs within storage to specify the problem and its solution

House Convex



Conditional gradient method

$$\min_{x \in \mathcal{D}} \quad f(x)$$



for $k = 1, 2, \ldots$ **do**

$$h_k = \underset{x \in \mathcal{D}}{\operatorname{argmin}} \langle \nabla f(x_k), x \rangle$$

$$x_{k+1} = (1 - \eta_k) x_k + \eta_k h_k$$

end for

Key feature: Rank-1 updates

$$\min_{X \in \Delta} \underbrace{\frac{1}{2}\|\mathcal{A}X - b\|^2}_{f(X)}$$

$$\underbrace{\mathrm{X} \in \mathbb{S}_+^n \text{ and } \mathrm{Tr}(X) = 1}_{\Delta}$$

for $k = 1, 2, \ldots$ **do**

$$H_k = \underset{X \in \Delta}{\mathrm{argmin}} \langle \nabla f(X_k), X \rangle$$

$$X_{k+1} = (1 - \eta_k) X_k + \eta_k H_k$$

end for

$$\Longrightarrow \begin{cases} u_k = \mathrm{minEigVec}(\mathcal{A}^\top(\mathcal{A}X_k - b)) \\ H_k = u_k u_k^\top \end{cases}$$

Dual conditional gradient method (CGM)

for $k = 1, 2, \ldots$ **do**

$$u_k = \text{minEigVec}(\mathcal{A}^\top(\mathcal{A}X_k - b))$$
$$X_{k+1} = (1 - \eta_k)X_k + \eta_k u_k u_k^\top$$

end for

Dual conditional gradient method (CGM)

$$
\begin{aligned}
&\textbf{for } k = 1, 2, \ldots \textbf{ do}\\
&\quad u_k = \mathrm{minEigVec}(\mathcal{A}^\top(z_k - b))\\
&\quad X_{k+1} = (1-\eta_k)X_k + \eta_k u_k u_k^\top\\
&\quad z_{k+1} = (1-\eta_k)z_k + \eta_k \mathcal{A} u_k u_k^\top\\
&\textbf{end for}
\end{aligned}
$$

Actions:

1- Introduce $z_k = \mathcal{A}X_k$

Storage cost $\mathcal{O}(n^2 + d)$

Dual conditional gradient method (CGM)

for $k = 1, 2, \ldots$ **do**

$u_k = \text{minEigVec}(\mathcal{A}^\top(z_k - b))$

~~$X_{k+1} = (1 - \eta_k)X_k + \eta_k u_k u_k^\top$~~

$z_{k+1} = (1 - \eta_k)z_k + \eta_k \mathcal{A} u_k u_k^\top$

end for

Actions:

1- Introduce $z_k = \mathcal{A}X_k$

2- Discard X_k

Storage cost $\mathcal{O}(n + d)$

Streaming linear updates

$$X_{k+1} = (1 - \eta_k) X_k + \eta_k u_k u_k^\top$$

$$X_{k+1} = \alpha_1 H_1 + \alpha_2 H_2 + \alpha_3 H_3 + \ldots + \alpha_k H_k$$

Key idea:

Use single pass algorithms for low-rank matrix approximation

Brief detour on sketching

[TYUC17a] Practical sketching algorithms for low-rank matrix approximation

[TYUC17b] Fixed-rank approximation of a positive-semidefinite matrix from streaming data

[TYUC19] Streaming Low-Rank Matrix Approximation with an Application to Scientific Simulation

Nystrom sketch

Draw and fix a standard normal test matrix $\Omega \in \mathbb{R}^{n \times R}$

Sketch "S" of "X" takes the form

$$S = X\Omega \in \mathbb{R}^{n \times R}$$

Sketch supports linear updates

$$X_{k+1} = (1 - \eta_k) X_k + \eta_k u_k u_k^\top$$

$$S_{k+1} = (1 - \eta_k) S_k + \eta_k u_k (u_k^\top \Omega)$$

Result: $\mathcal{O}(R^2 n)$ arithmetic and $\mathcal{O}(Rn)$ storage cost

$$S = X\Omega \in \mathbb{R}^{n \times R}$$

We reconstruct X via Nystrom approximation

$$\hat{X} = S(\Omega^\top S)^\dagger S^\top = (X\Omega)(\Omega^\top X\Omega)^\dagger (X\Omega)^\top$$

Rank-r Truncation

$$\mathbb{E}_\Omega \|X - \hat{X}\|_* \leq \left(1 + \frac{r}{R-r}\right) \|X - [X]_r\|_* \qquad (\forall r < R)$$

Sketchy CGM

for $k = 1, 2, \ldots$ **do**

$$u_k = \text{minEigVec}(\mathcal{A}^\top(z_k - b))$$
$$z_{k+1} = (1 - \eta_k) z_k + \eta_k \mathcal{A} u_k u_k^\top$$
$$S_{k+1} = (1 - \eta_k) S_k + \eta_k u_k u_k^\top \Omega$$

end for

$$U_k = \text{SketchReconstruct}(\Omega, S_k)$$

$$X_{\text{output}} = U_k U_k^\top$$

Actions:

1- Introduce $z_k = \mathcal{A} X_k$

2- Discard X_k

3- Introduce sketch $S_k = X_k \Omega$

Storage cost $\mathcal{O}(rn + d)$

Homotophy CGM

$$\min_{X\in\Delta} \quad \langle C, X\rangle \quad \text{subj.to} \quad \mathcal{A}X = b$$

Constrained formulation

$$\min_{X\in\Delta} \quad \underbrace{\langle C, X\rangle + \frac{\lambda}{2}\|\mathcal{A}X - b\|^2}_{f_\lambda(X)}$$

Quadratic penalty formulation

$\lambda \to \infty$

for $k = 1, 2, \ldots$ **do**

$\quad \lambda_k = \lambda_0\sqrt{k+1}$

$\quad u_k = \text{minEigVec}(C + \lambda_k \mathcal{A}^\top(\mathcal{A}X_k - b))$

$\quad X_{k+1} = (1-\eta_k)X_k + \eta_k u_k u_k^\top$

end for

$$f(x_k) - f^\star \le \mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$$

$$\|Ax_k - b\|_2 \le \mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$$

Optimal

Conditional Gradient Augmented Lagrangian (CGAL)

$$\min_{x} \left\{ \underbrace{\mathrm{tr}(cx) + \frac{1}{2\beta}\|Ax - b\|^2 + y^*(Ax - b)}_{=:\mathcal{L}_\beta(x,y)} : x \succeq 0, x^* = x, \mathrm{tr}(x) = \rho \right\}$$

For $k = 0$ **to** $k_{\max}$:

$\eta_k = \frac{2}{k+1}$ and $\beta_k = \frac{\beta_0}{\sqrt{k+1}}$

$\nabla_x \mathcal{L}_{\beta_k} = c + \frac{1}{\beta_k} A^*(Ax_k - b) + A^* y_k$

$u_k = \mathtt{MaxEigVec}(\nabla_x \mathcal{L}_{\beta_k})$

$\dot{x}_k = \rho u_k u_k^*$

$x_{k+1} = (1 - \eta_k)x_k + \eta_k \dot{x}_k$

$y_{k+1} = y_k + \sigma_k(Ax_{k+1} - b)$

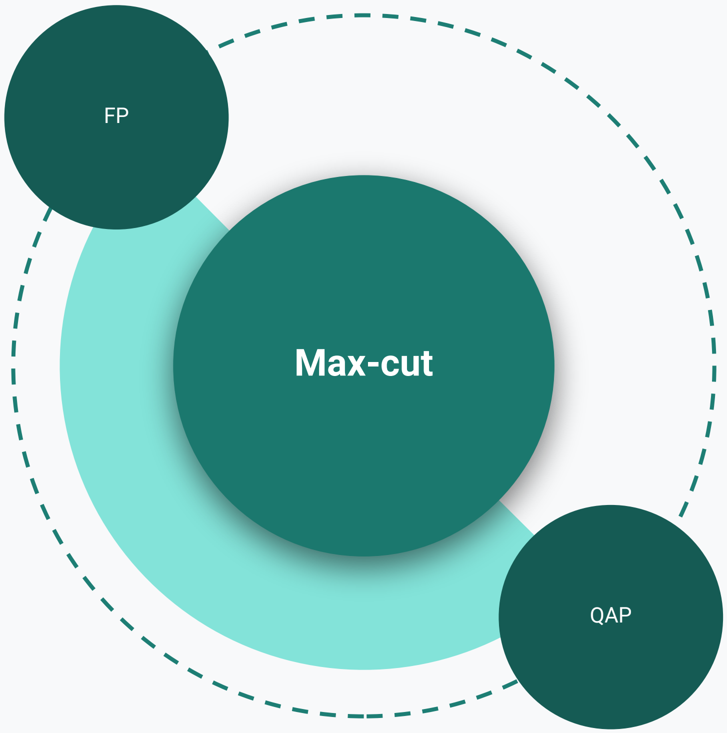
End for

$$|f(x_k) - f^\star| = \mathcal{O}\left(\frac{1}{\sqrt{k}}\right) \quad \|Ax_k - b\| = \mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$$

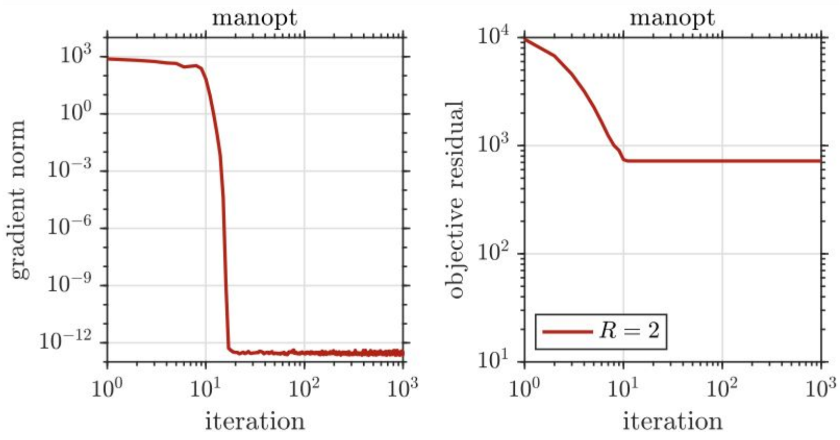
$$\text{SDP complexity} = \mathcal{O}\left(\frac{n}{\epsilon^3}\right)$$

Yurtsever, Fercoq, Cevher (2019) “A conditional gradient based augmented Lagrangian framework”
Yurtsever, Tropp, Fercoq, Udell, Cevher (2020) “Scalable semidefinite programming”

Numerical Evidence



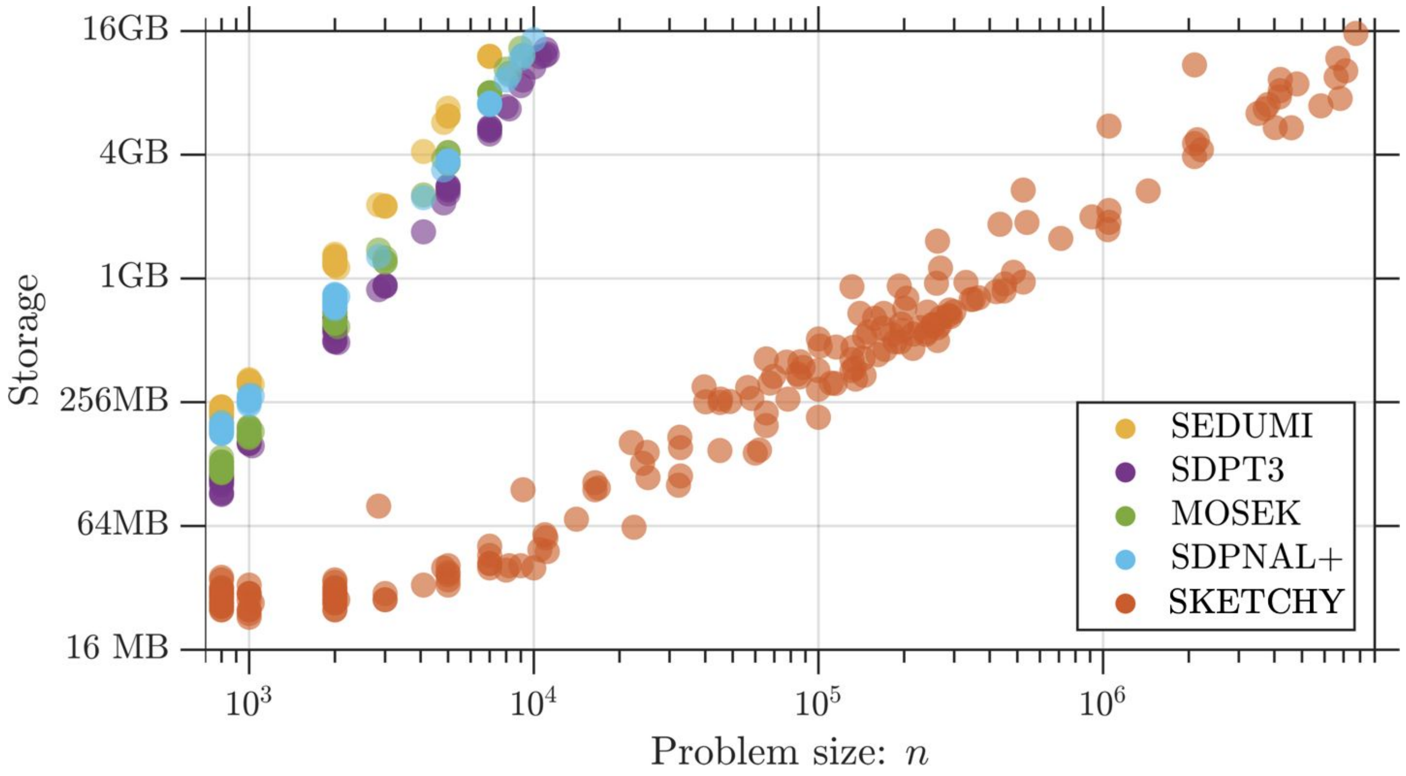
Max-cut



Dataset / R	$R=2$	3	4	5	6	7	8	9	10	11	12	13
C_1	82	69	63	53	35	32	24	12	11	1	4	0
C_2	77	56	56	36	19	17	12	2	0	0	0	0
C_3	89	65	54	47	44	46	23	11	5	0	3	0
C_4	84	69	50	40	27	23	18	17	1	0	9	0
C_5	85	68	52	51	43	30	31	20	14	3	4	0
C_6	81	68	53	41	23	22	10	10	2	0	1	0
C_7	83	76	60	39	19	19	19	3	0	0	1	0
C_8	81	73	44	34	41	25	8	12	5	4	10	0
C_9	84	64	46	35	25	17	1	10	0	2	4	0
C_{10}	83	71	54	50	31	25	24	16	13	0	8	0

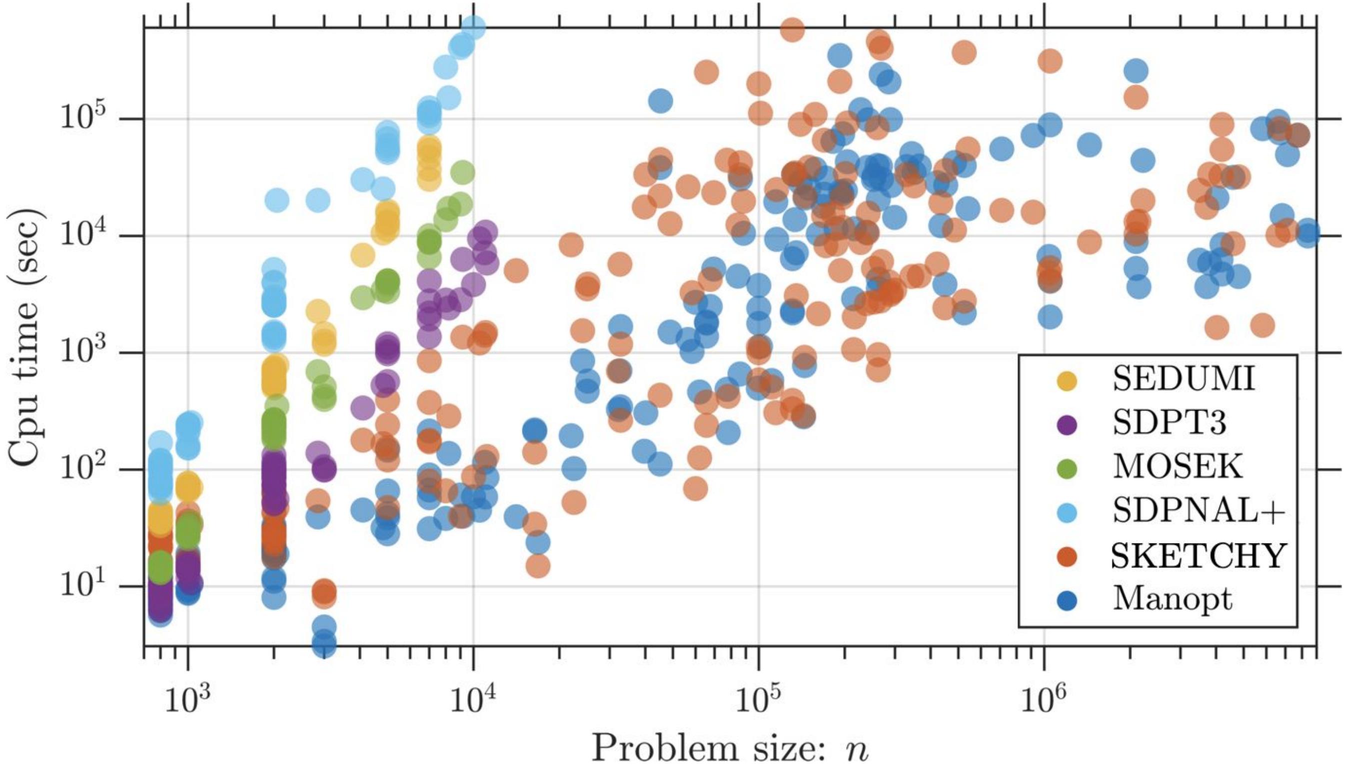
Waldspurger, Waters (2019) “Rank optimality for the Burer-Monteiro factorization”

Max-cut



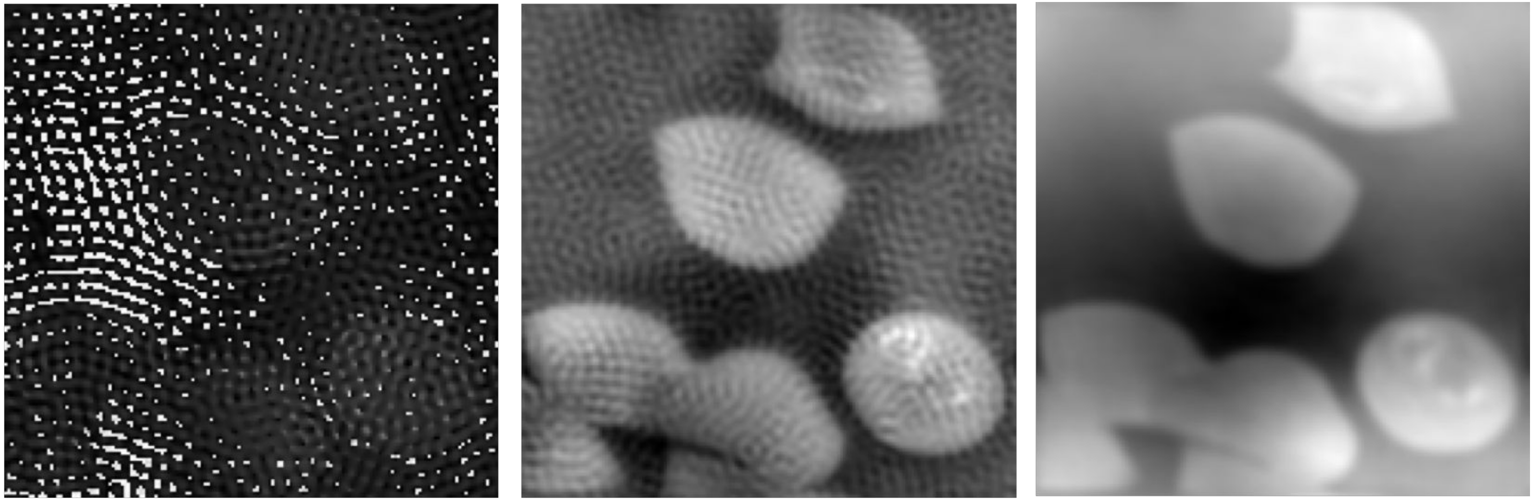
GSet
DIMACS10
data groups

Max-cut



GSet
DIMACS10
data groups

Fourier Ptychography



Wirtinger Flow
(Rank-1 BM)

Burer-Monteiro

Sketchy

Real data

image size 160 × 160 pixels
matrix size 25′600 × 25′600 ⟹ ~5.25 GB
d = 185′600 measurements

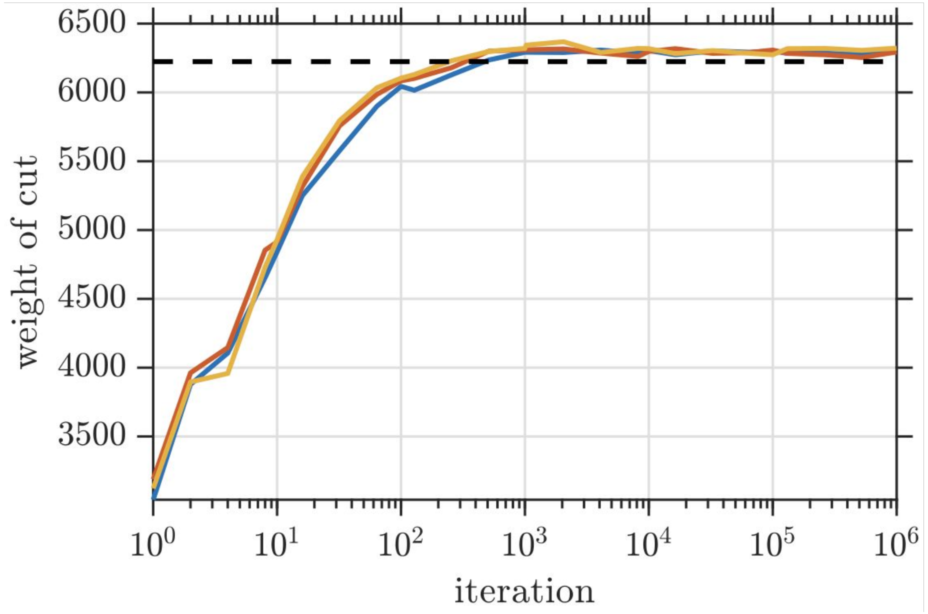
On the accuracy of solutions

$$\mathrm{err} \approx \mathrm{err}_{\mathrm{model}} + \mathrm{err}_{\mathrm{opt}}$$

On the accuracy of solutions

G67 Dataset

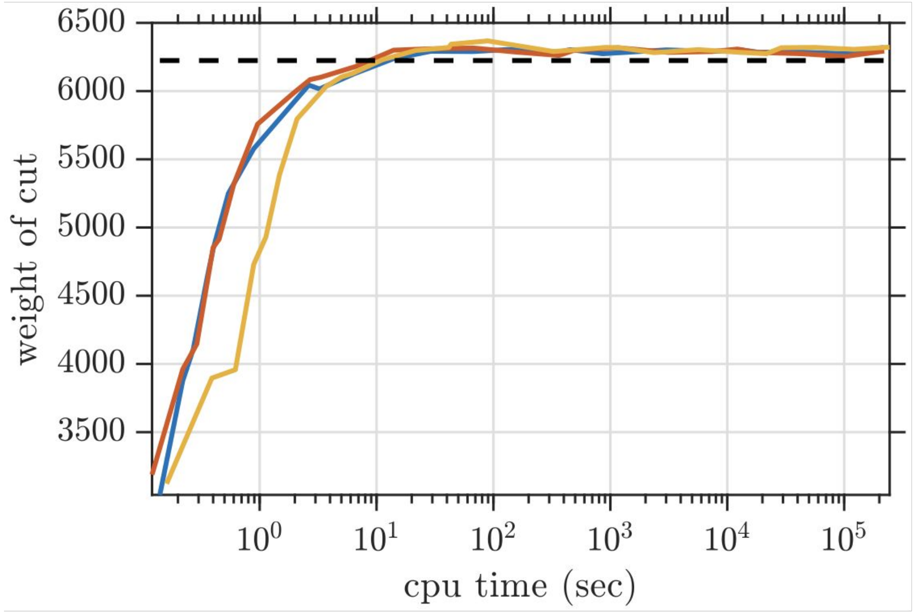
$n = 10^4$



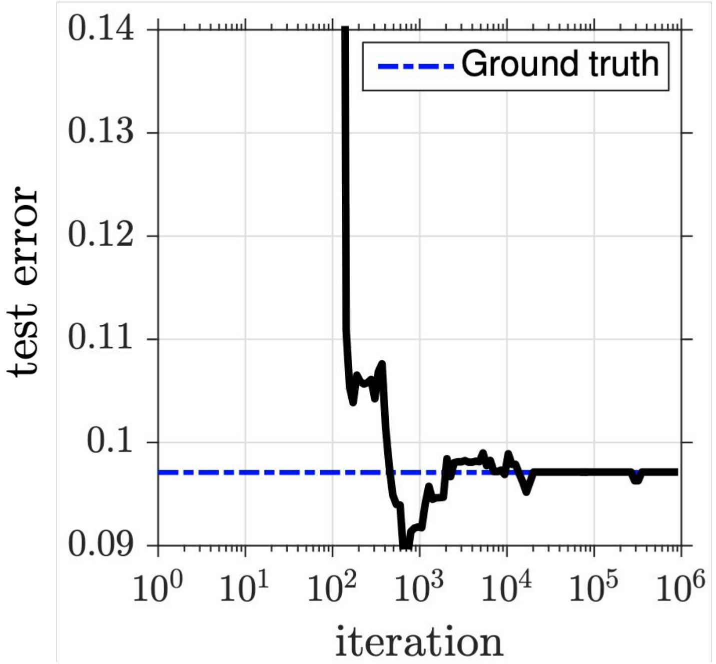
On the accuracy of solutions

G67 Dataset

$n = 10^4$



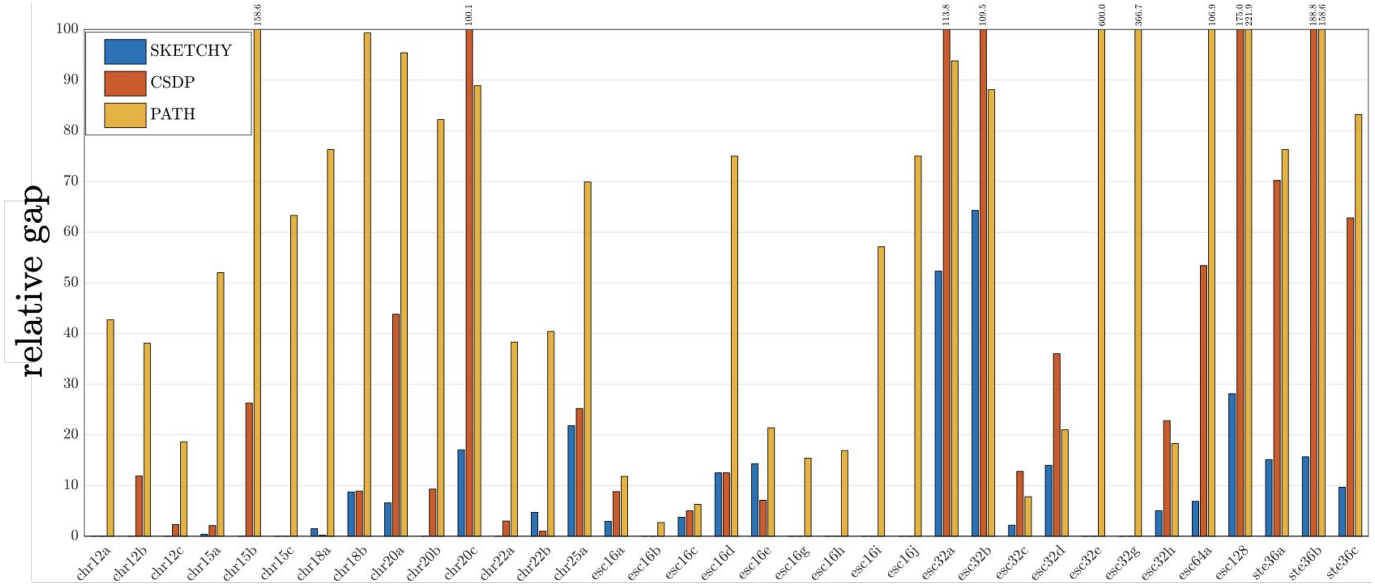
On the accuracy of solutions



Preprocessed (& sampled) MNIST data

$n = 10^3$

Quadratic assignment



Comparison against:

Ferreira, Khoo, Singer, (2017)

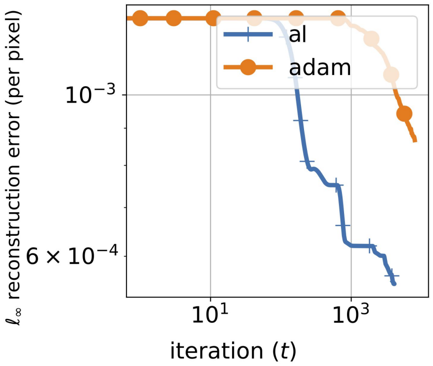
"Semidefinite Programming Approach for the Quadratic Assignment Problem with a Sparse Graph"

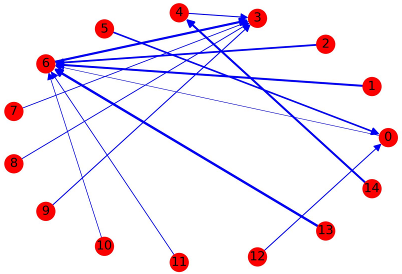
Conclusions

- Extensions to stochastic SDPs
  - Locatello et al. NeurIPS 2019
  - Vladeran et al. ICML 2020

- Non-convex nonlinear programs
  - Latorre et al. NeurIPS 2019
  - Sahin et al. under review

- Algorithm design with computational primitives
  - randomized linear algebra





volkan.cevher@epfl.ch