Lift me up but not too high

Fast algorithms to solve SDP's with block-diagonal constraints

Nicolas Boumal
Université catholique de Louvain (Belgium)
IDeAS seminar, May 13th, 2014, Princeton

The Riemannian staircase

Because sometimes
you’re just going to the second floor

This talk is about solving this, fast:

$$\min_X \; f(X)$$

$$X = X^T \succcurlyeq 0,$$

$$X_{ii} = I_d \text{ for } i = 1 \ldots m.$$

Let’s see how it comes up in applications.

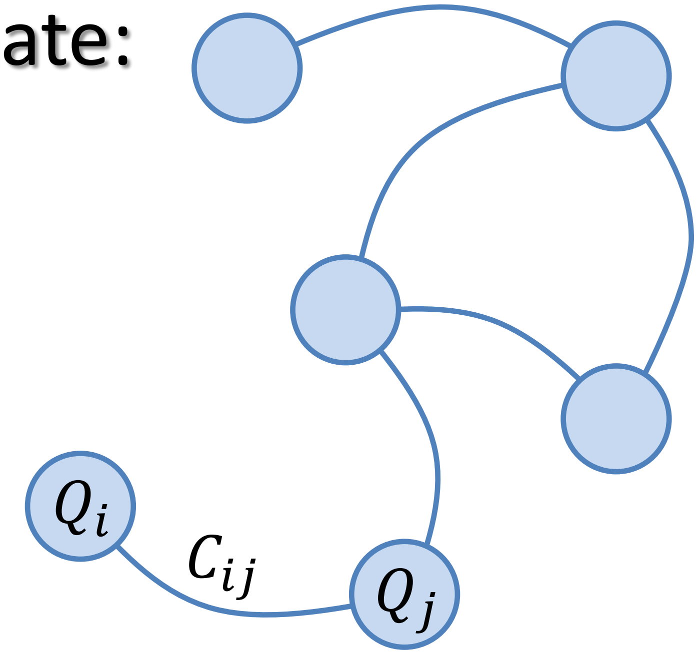
Synchronization of rotations

Orthogonal matrices to estimate:

$$Q_1, Q_2, \dots, Q_m \in O(d).$$

Measurements:

$$C_{ij} = Q_i Q_j^T + \epsilon_{ij}$$



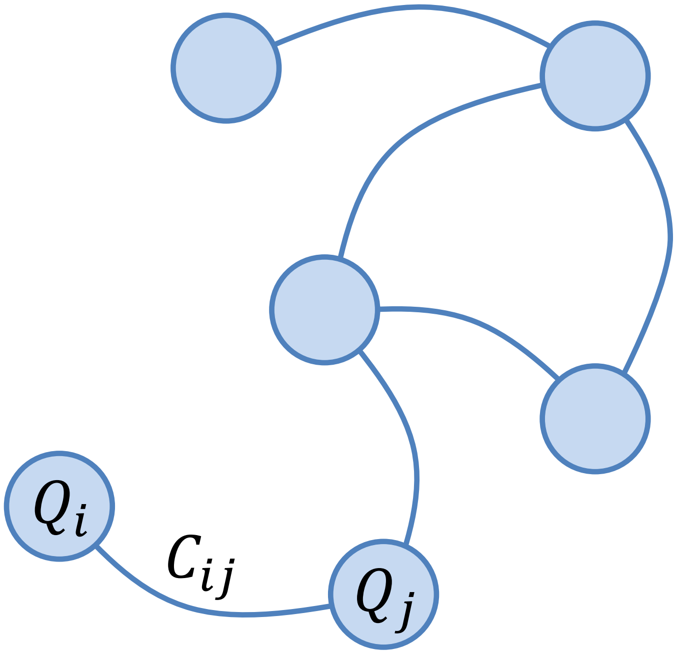
Synchronization of rotations

Measurements (white noise):

$$C_{ij} = Q_i Q_j^T + \epsilon_{ij}$$

Maximum likelihood:

$$\min_{\hat{Q}_i \in O(d)} \sum_{i,j} \left\| C_{ij} - \hat{Q}_i \hat{Q}_j^T \right\|^2$$



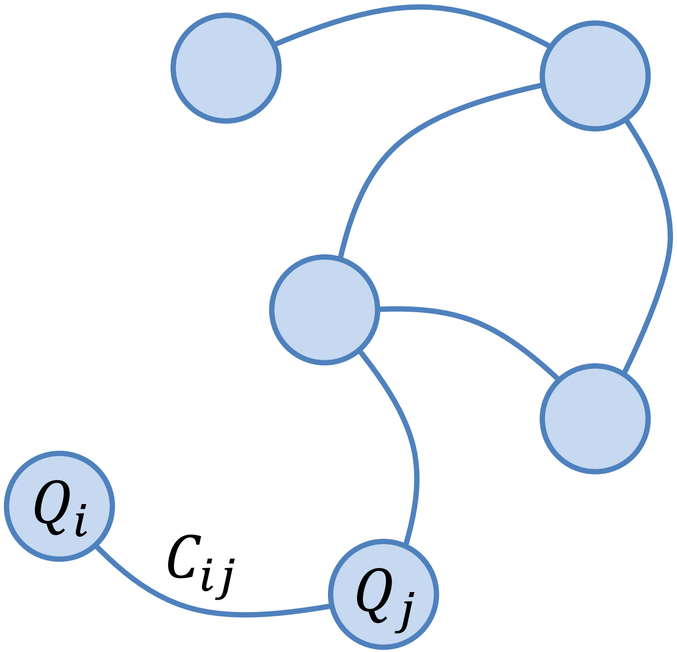
Synchronization of rotations

Measurements (white noise):

$$C_{ij} = Q_i Q_j^T + \epsilon_{ij}$$

Maximum likelihood:

$$\max_{\hat{Q}_i \in O(d)} \sum_{i,j} \mathrm{Trace}(C_{ij}^T \hat{Q}_i \hat{Q}_j^T)$$



Maximizing the likelihood is NP-hard

Indeed: if d = 1, this includes Max-Cut

$$\max \sum_{i,j} \mathrm{Trace}(C_{ij}^T \hat{Q}_i \hat{Q}_j^T)$$

Such that $\hat{Q}_i \hat{Q}_i^T = I_d$ for $i = 1 \ldots m$

The classic trick is to lift:
replace quadratic terms by linear ones.

$$\max \sum_{i,j} \mathrm{Trace}(C_{ij}^T \hat{Q}_i \hat{Q}_j^T)$$

Such that $\hat{Q}_i \hat{Q}_i^T = I_d$ for $i = 1 \ldots m$

Introduce $X_{ij} = \hat{Q}_i \hat{Q}_j^T$

The classic trick is to lift:
replace quadratic terms by linear ones.

$$\max \sum_{i,j} \mathrm{Trace}(C_{ij}^T X_{ij})$$

Such that $X_{ii} = I_d$ for $i = 1 \ldots m$

Introduce $X_{ij} = \hat{Q}_i \hat{Q}_j^T$

From Q to X, a block matrix such that:

$X_{ij} = \hat{Q}_i \hat{Q}_j^T$, thus:

$$X = \begin{pmatrix} \hat{Q}_1 \\ \vdots \\ \hat{Q}_m \end{pmatrix} \begin{pmatrix} \hat{Q}_1^T & \dots & \hat{Q}_m^T \end{pmatrix} \in \mathbb{R}^{md \times md}$$

From Q to X, a block matrix such that:

$$X = \begin{pmatrix} \hat{Q}_1 \\ \vdots \\ \hat{Q}_m \end{pmatrix} \begin{pmatrix} \hat{Q}_1^T & \dots & \hat{Q}_m^T \end{pmatrix} \in \mathbb{R}^{md \times md}$$

In other words:

$$X = X^T \succcurlyeq 0,$$
$$X_{ii} = I_d \text{ for } i = 1 \dots m,$$
$$\text{rank}(X) = d.$$

This new problem formulation is equivalent to the original one.

$$\max_X \ \mathrm{Trace}(CX)$$

$$X = X^T \succcurlyeq 0,$$

$$X_{ii} = I_d \text{ for } i = 1 \ldots m,$$

$$\mathrm{rank}(X) = d$$

Dropping the rank constraint altogether yields an SDP relaxation.

$$\max_X \ \mathrm{Trace}(CX)$$

$$X = X^T \succcurlyeq 0,$$

$$X_{ii} = I_d \text{ for } i = 1 \dots m.$$

This is sometimes called the Orthogonal-Cut SDP.

If $C \succcurlyeq 0$, the value of this SDP approximates the value of the rank-constrained (hard) problem.

More generally, we address this problem (with f convex, smooth):

$$\min_X \quad f(X)$$

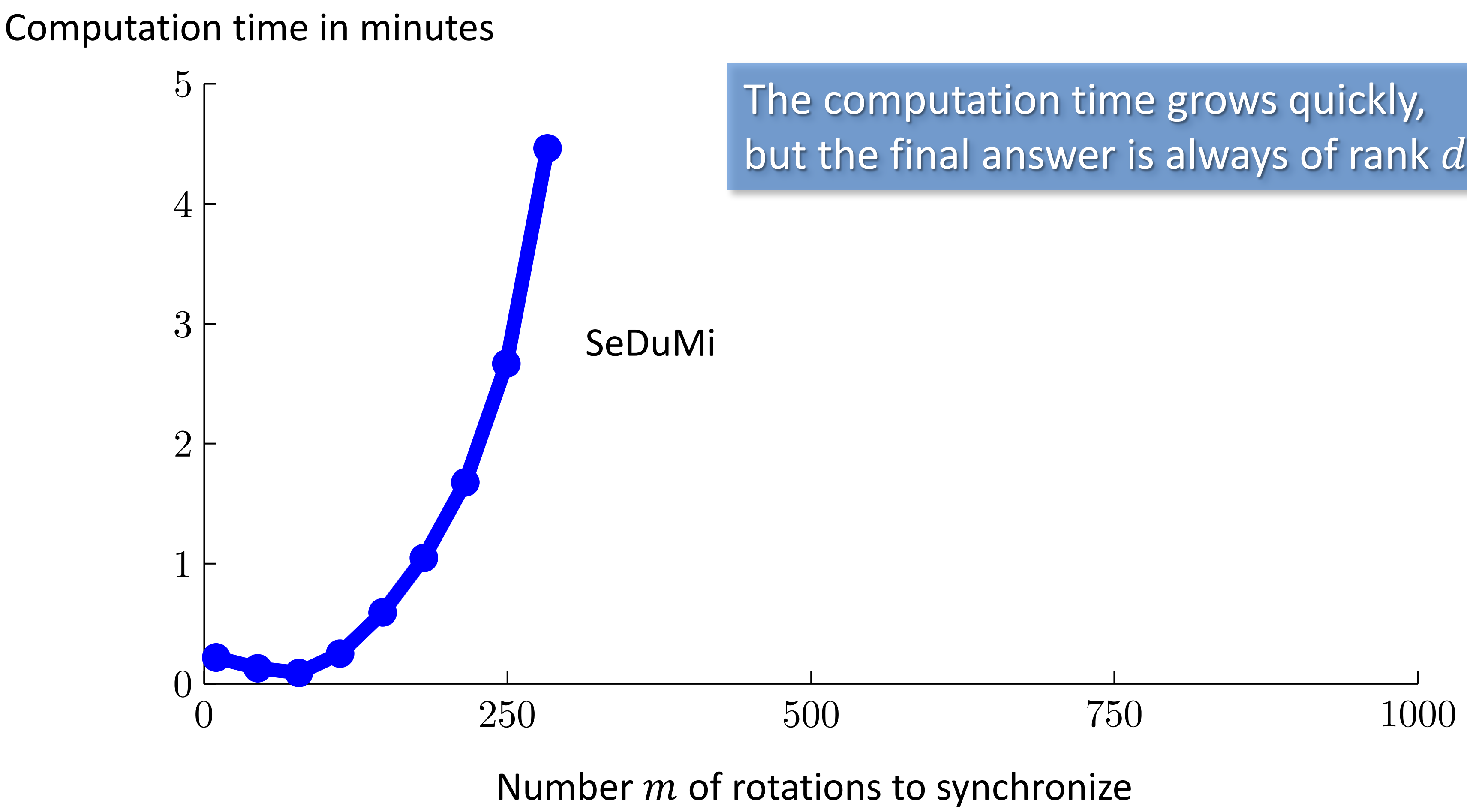
$$X = X^T \succcurlyeq 0,$$

$$X_{ii} = I_d \text{ for } i = 1 \ldots m.$$

Control over the cost means we can aim for robustness.

A few different applications involve the same formulation

- The generalized Procrustes problem
- Global registration (Chaudhury et al. ‘12)
- Synchronization of rotations (Singer ‘11)
- Common lines registration (LUD) (Wang et al. ‘13)
- Orthogonal-Cut (Bandeira et al. ‘13),
  Phase-Cut (Waldspurger et al. ‘12), Max-Cut (Goemans et al. ‘95)

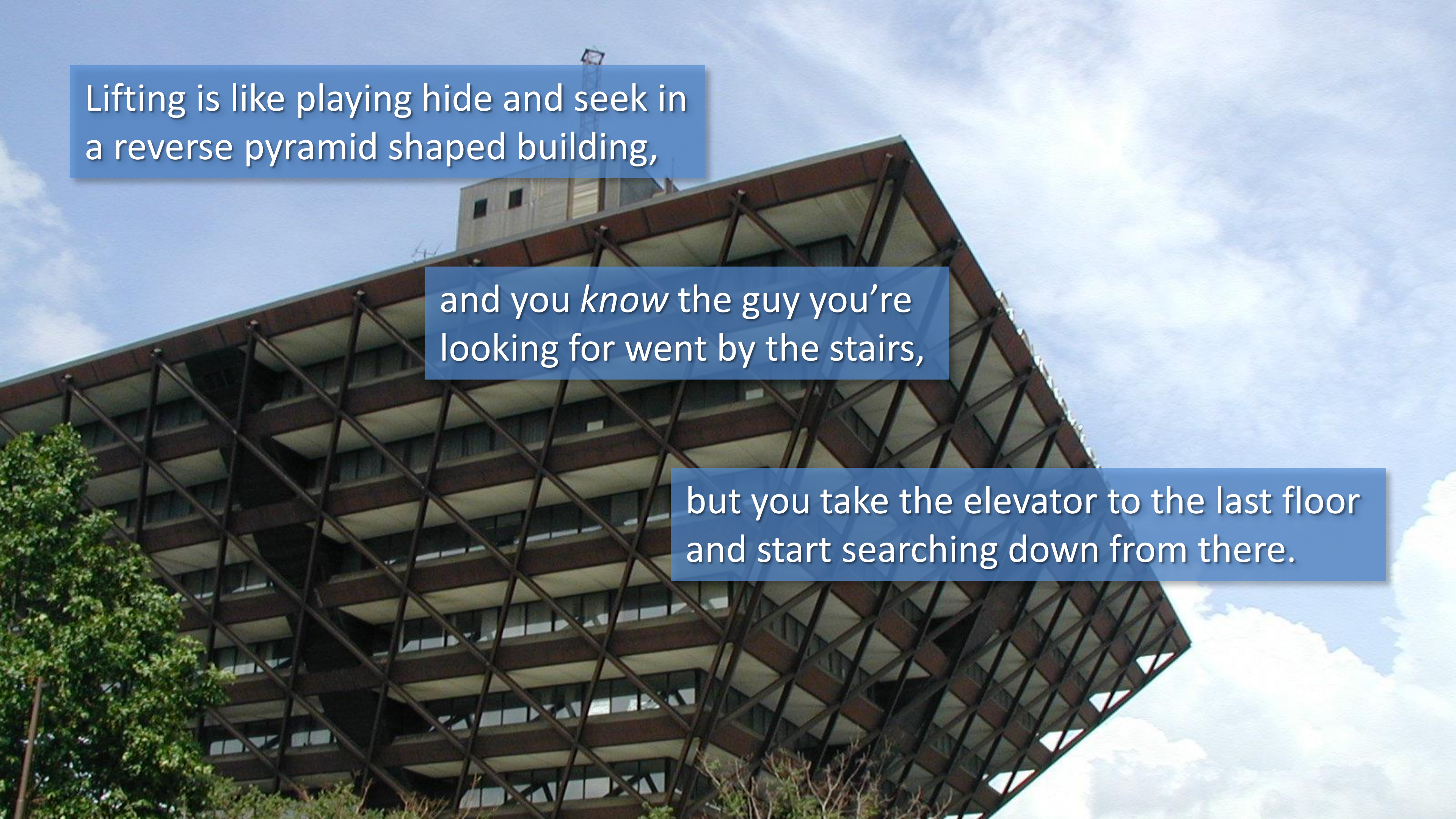


The computation time grows quickly, but the final answer is always of rank d.

We should expect low-rank solutions

There exists a solution of rank $\leq \sqrt{n(d+1)}$
(Pataki ’98, for the linear cost case)

Wishful thinking: the underlying problem “calls” for a low-rank solution… (?)


Lifting is like playing hide and seek in a reverse pyramid shaped building,
and you *know* the guy you're looking for went by the stairs,
but you take the elevator to the last floor and start searching down from there.

The SDPLR idea (Burer et al. ’03, ’04): Factorize with tall and skinny Y.

$$\min_{Y} \ \mathrm{Trace}(CYY^T)$$

$$\text{such that } X = YY^T \text{ is feasible,}$$

$$Y \in \mathbb{R}^{n \times p}.$$

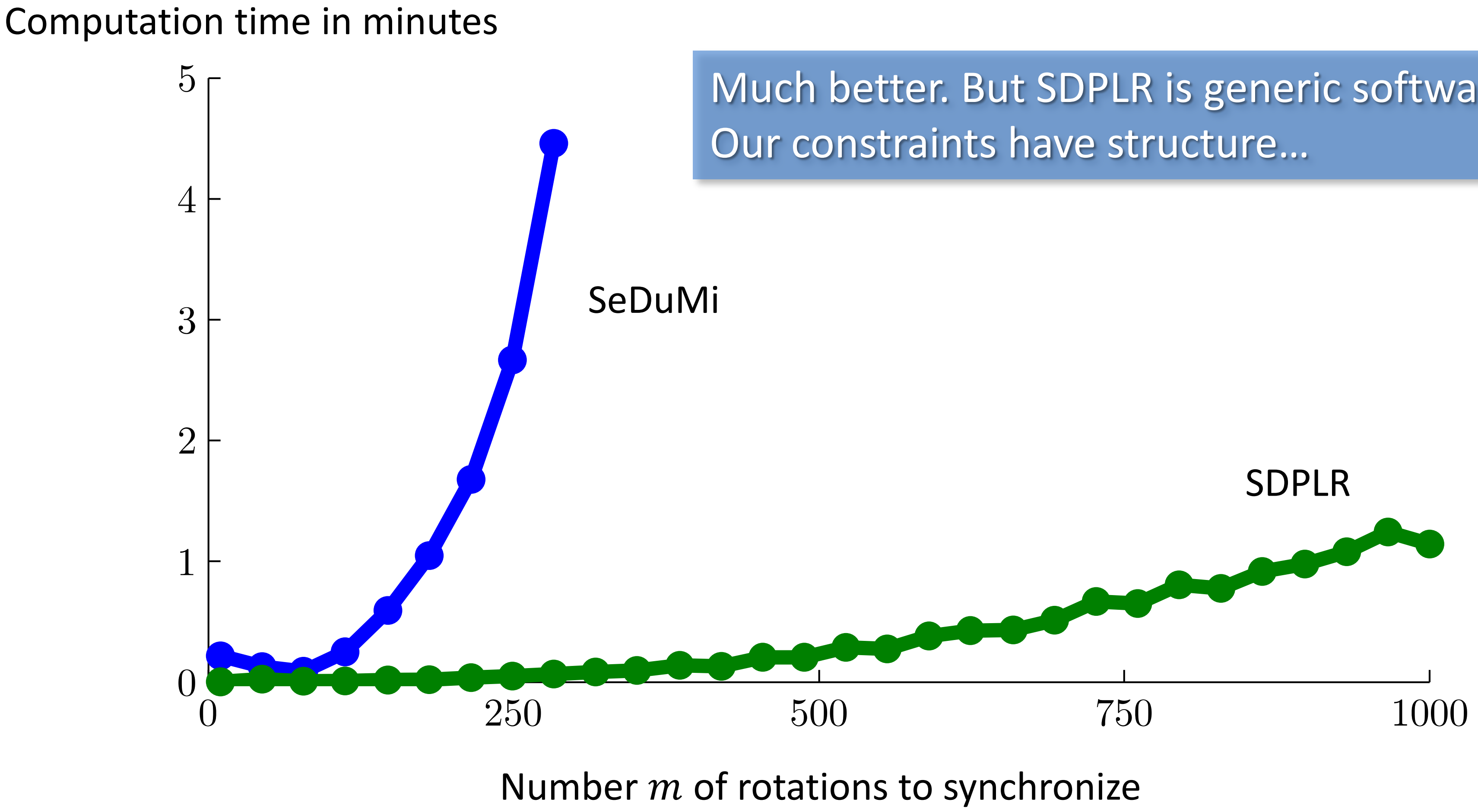
$\mathrm{rank}(X) \leq p$

They handle constraints via an augmented Lagrangian.
If Y is rank deficient, X is optimal.

What if most local optimizers are full-rank?

In practice, we don't see that. Burer and Monteiro ('04) explain why for linear cost functions (Theorem 3.4):

Suppose Y is a local optimizer for p such that $p \geq (d+1)\sqrt{m}$. Then, $X = YY^T$ is contained in the relative interior of a face F of the SDP over which the objective function is constant. Moreover, if F is just an extreme point, then X is a global optimizer of the SDP.



Much better. But SDPLR is generic software.
Our constraints have structure…

Acceptable Y's live on a manifold

$$X = X^T \succcurlyeq 0,$$
$$\text{rank}(X) \leq p,$$
$$X_{ii} = I_d \;\forall i$$

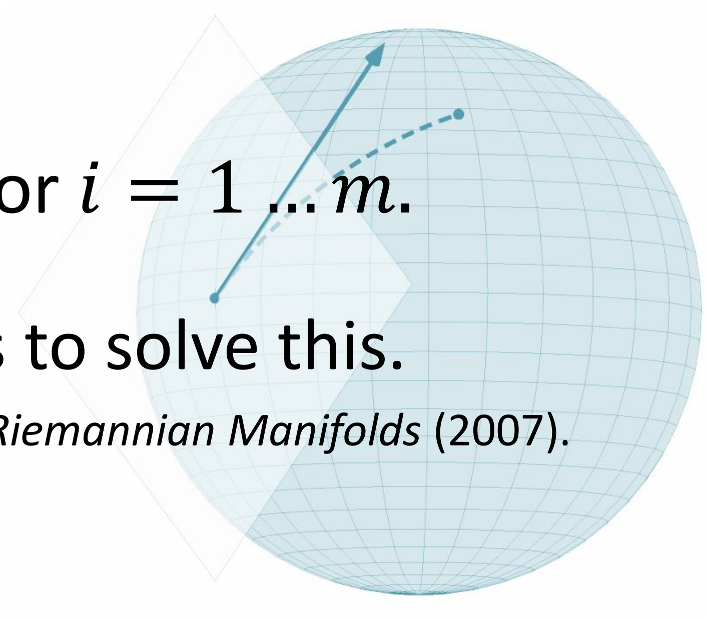
$$\Leftrightarrow$$

$$\exists\, Y \in \mathbb{R}^{n \times p} \text{ such that}$$
$$X = YY^T,$$
$$Y = \begin{pmatrix} Y_1 \\ \vdots \\ Y_m \end{pmatrix}, Y_i \in \mathbb{R}^{d \times p}$$
$$Y_i Y_i^T = I_d \;\forall\, i$$

Thus, the nonlinear program is a Riemannian optimization problem

$$\min_{Y} \; f(YY^T)$$

$$Y_i \text{ is } d \times p \text{ orthonormal for } i = 1 \ldots m.$$



We use Riemannian Trust-Regions to solve this.

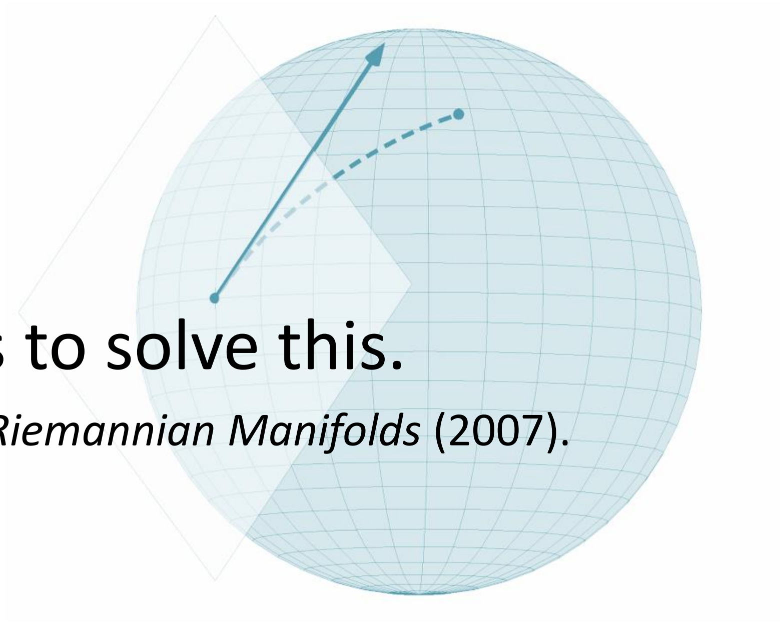
See Absil, Baker, Gallivan: *Trust-Region Methods on Riemannian Manifolds* (2007).

Matlab toolbox: manopt.org

Thus, the nonlinear program is a Riemannian optimization problem

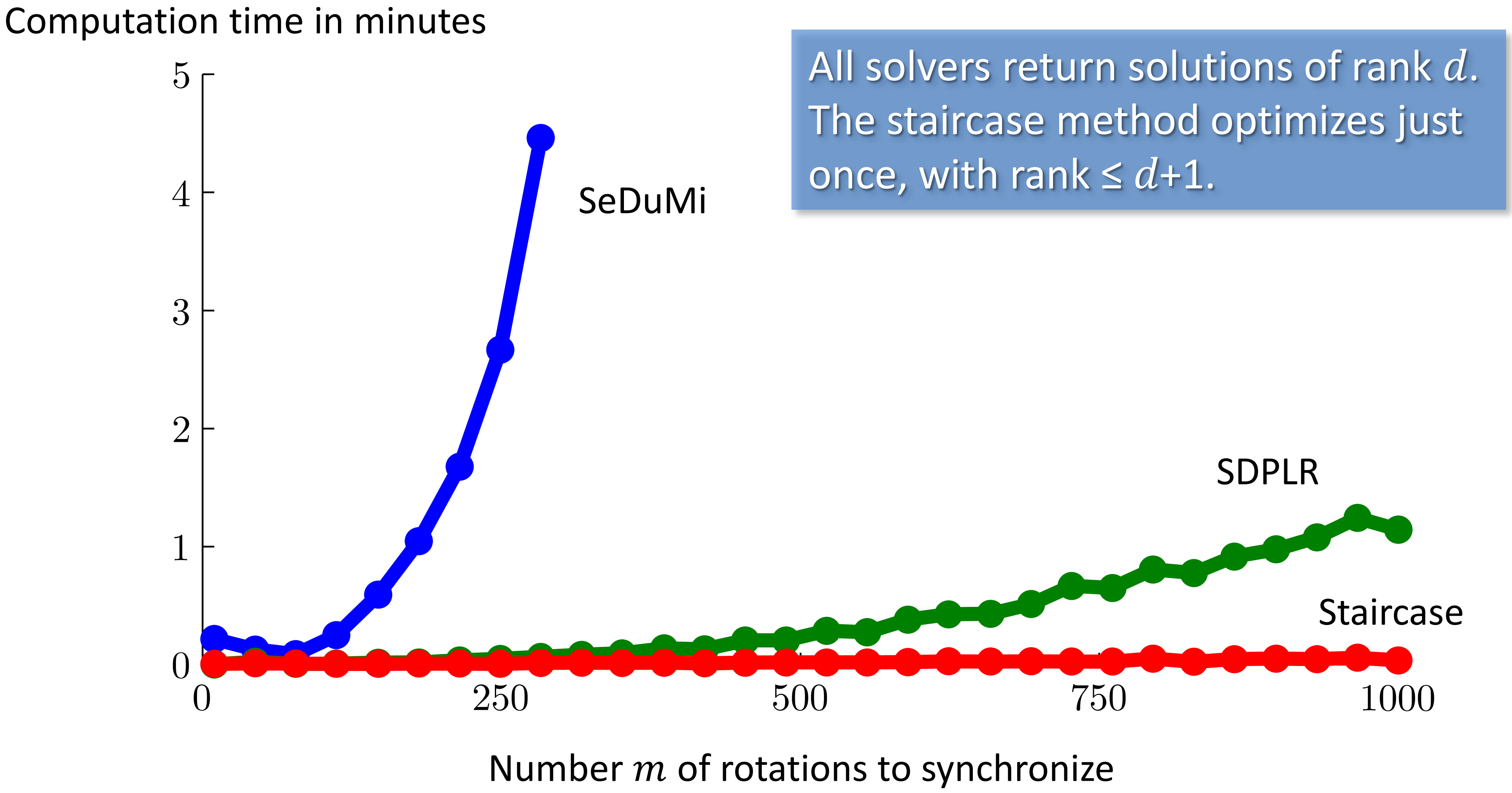
$$\min_{Y} \; f(YY^T)$$

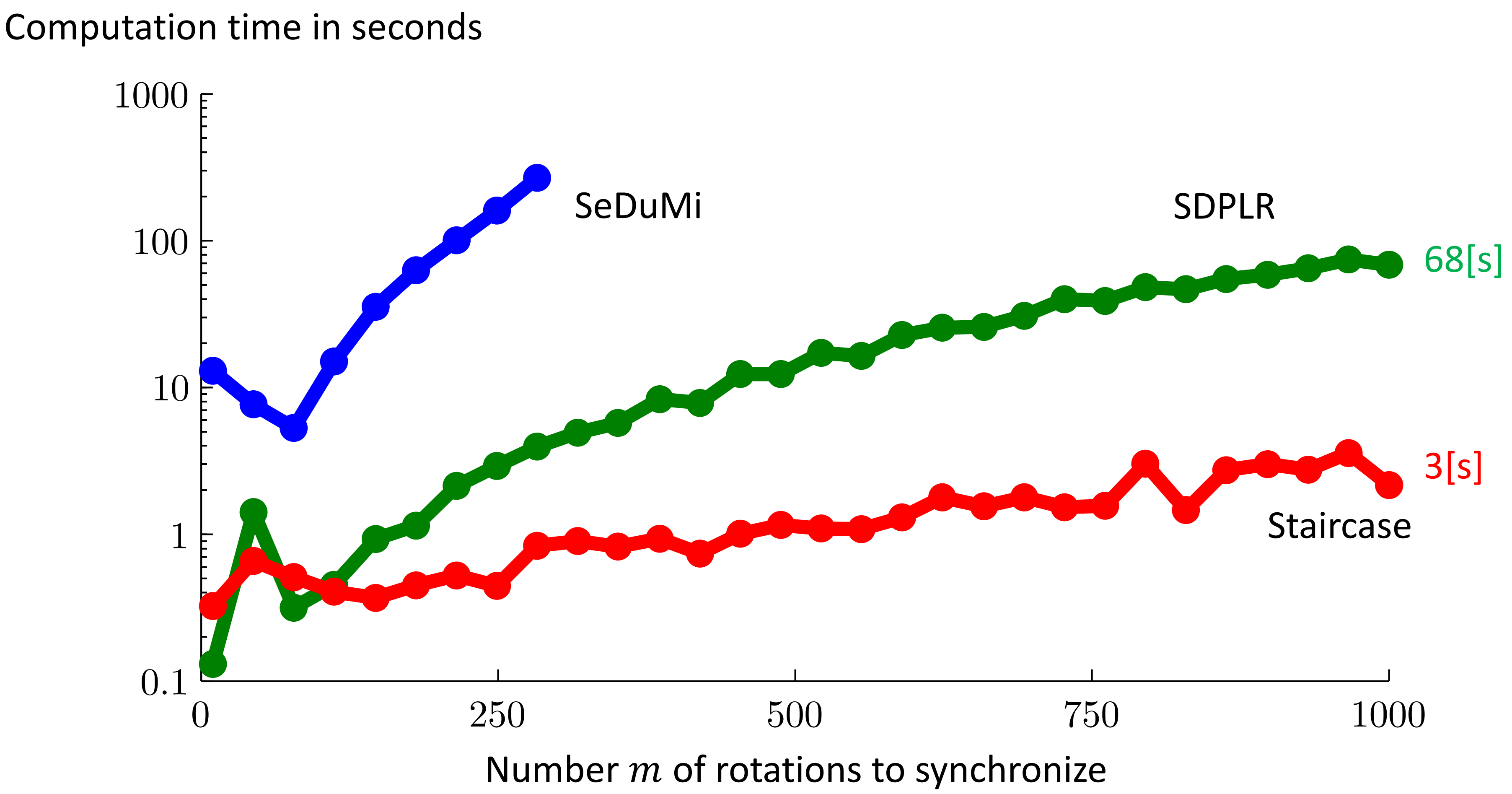
$$Y \in \mathrm{Stiefel}(d, p)^m.$$



We use Riemannian Trust-Regions to solve this.

See Absil, Baker, Gallivan: *Trust-Region Methods on Riemannian Manifolds* (2007).
Matlab toolbox: manopt.org


Computation time in minutes
5
4
3
2
1
0
0
250
500
750
1000
Number m of rotations to synchronize
SeDuMi
SDPLR
Staircase
All solvers return solutions of rank d.
The staircase method optimizes just once, with rank ≤ d+1.



Computation time in seconds
1000
100
10
1
0.1
SeDuMi
SDPLR
68[s]
3[s]
Staircase
0
250
500
750
1000
Number m of rotations to synchronize

Pros and cons

SDPLR	Our method
Deals with any SDP	Is restricted to diagonal block constraints
Handles only linear costs	Handles any smooth cost (guarantees if convex)
Penalizes constraints in the cost	Satisfies the constraints at all iterates
Is mature C code	Is experimental Matlab code

That's all very well in practice, but does it work in theory?

$$\min_X \; f(X) \qquad X \succcurlyeq 0, X_{ii} = I_d.$$

$$\min_Y \; g(Y) = f(YY^T) \qquad Y \in \text{Stiefel}(d,p)^m.$$

Theorem:

Let Y be a local minimizer of the nonlinear program.
If Y is rank deficient and/or if $p = n$ (Y is square),
then $X = YY^T$ is a global minimizer of the convex program.

This suggests an algorithm

1. Set $p = d + 1$.
2. Compute Y_p, a Riemannian local optimizer.
3. If Y_p is rank deficient, stop.
4. Otherwise, increase p and go to 2.

This is guaranteed to return a globally optimal X.
(Worst case scenario: p increases all the way to n.)

From local opt Y to global opt X.

$$\min_X \ f(X)$$
$$X \succcurlyeq 0, X_{ii} = I_d.$$

$$\min_Y \ g(Y) = f(YY^T)$$
$$Y \in \text{Stiefel}(d, p)^m.$$

X is globally optimal iff there exists S such that: (KKT)

$X \succcurlyeq 0, X_{ii} = I_d$

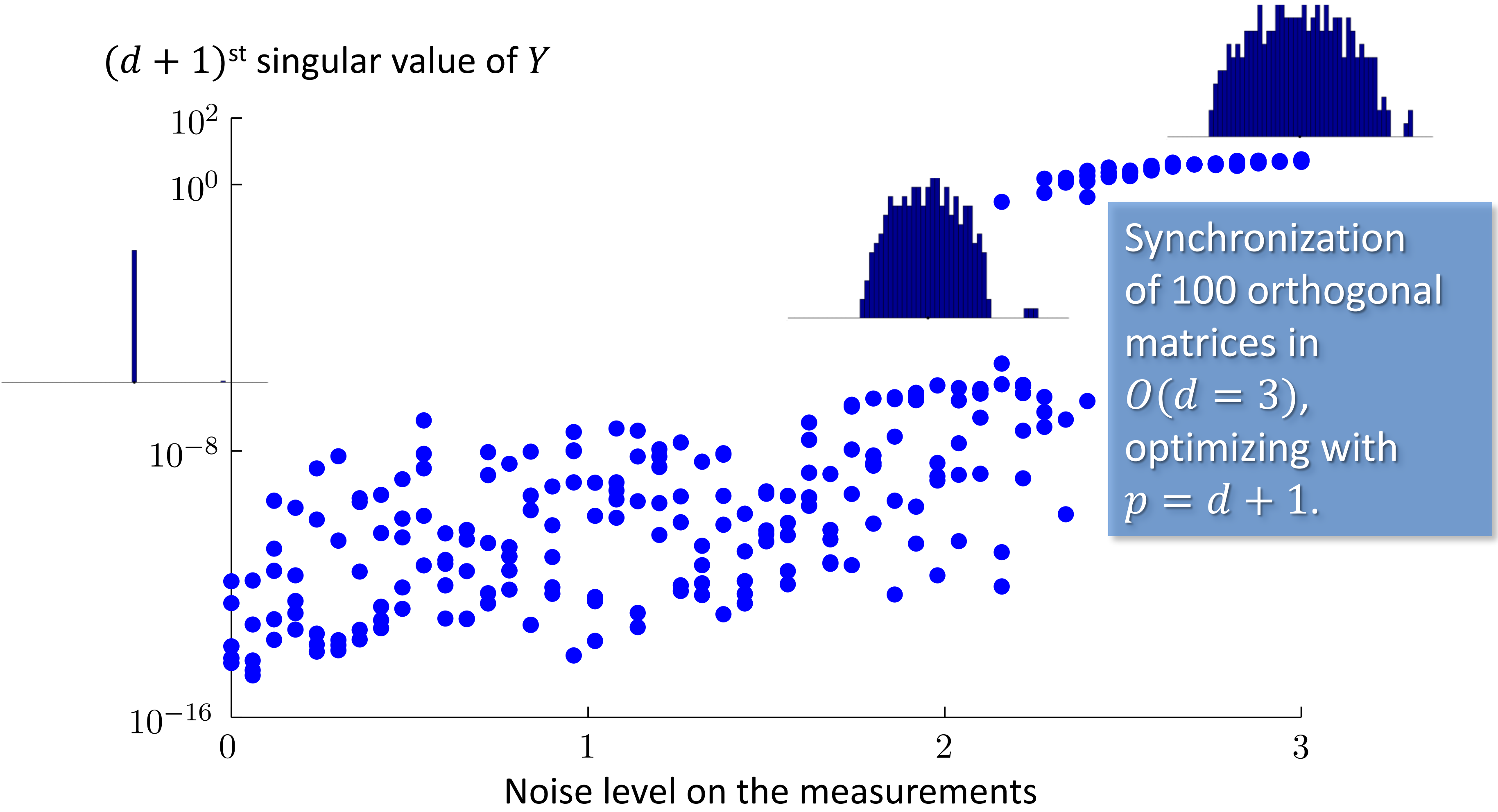
$S \succcurlyeq 0, SX = 0$

$\nabla f(X) - S$ is block diagonal

If Y is locally optimal, then

$$\text{grad } g(Y) = 0$$
$$\text{Hess } g(Y) \succcurlyeq 0$$

These are the Riemannian (projected) gradient and Hessian.



$(d+1)^{\text{st}}$ singular value of Y
10^{2}
10^{0}
10^{-8}
10^{-16}
0
1
2
3
Noise level on the measurements
Synchronization of 100 orthogonal matrices in $O(d = 3)$, optimizing with $p = d + 1$.

Phase transition for rank recovery?

- It appears that even at high levels of noise, the SDP admits a rank d solution.

- This solves the hard problem…

- How can we understand this?

μ-Partial answer: single cycle synch

For synchronization on a cycle, with measurements

$$C_{12}, C_{23}, C_{34}, \dots, C_{m1} \in O(d),$$

if the product of the measurements

Proof: write explicit solution and intuit a dual certificate.

$$P = C_{12} C_{23} \dots C_{m1}$$

has no eigenvalue -1, the SDP has a rank d solution.

Three further ideas to think about

- Robust works too: minimize sum of unsquared errors with Huber regularization, fast.
- Fancy rounding technique: if $\mathrm{rank}(X) > d$, project to rank d and re-optimize.
- Additional constraints could be handled by convex penalties (research in progress).

Will you take the stairs next time?

Code available on my webpage.

Or e-mail me: nicolasboumal@gmail.com

Robust synchronization: the least-unsquared deviation approach (LUD)

$$\min_{R_1,\ldots,R_m} \sum_{i\sim j} \left\| C_{ij} - R_i R_j^T \right\|_F$$

$$\text{such that } R_i R_i^T = I_d \text{ and } \det(R_i) = 1.$$

Wang & Singer 2013:
Exact and stable recovery of rotations for robust synchronization.

Robust synchronization: the least-unsquared deviation approach (LUD)

$$\min_{X \succcurlyeq 0} \sum_{i \sim j} \left\| C_{ij} - X_{ij} \right\|_F$$

such that $X_{ii} = I_d$ ~~and $\det(R_i) = 1$~~.

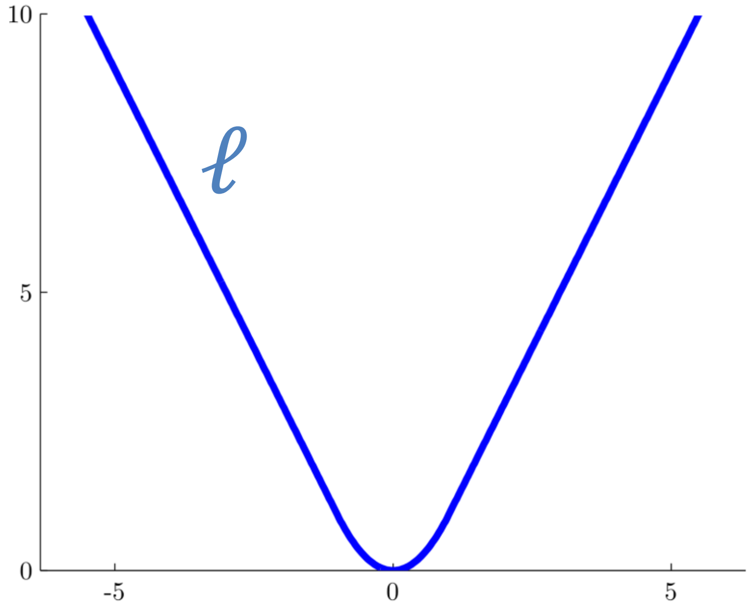
Nonlinear cost: SDPLR does not apply.
The authors use an adapted ADMM.

Wang & Singer 2013:
Exact and stable recovery of rotations for robust synchronization.

Robust synchronization: Smoothing the cost with a Huber loss.

$$\min_{X \succcurlyeq 0} \sum_{i \sim j} \ell\left(\|H_{ij} - X_{ij}\|_F\right)$$

such that $X_{ii} = I_d$.



Dealing with additional constraints?

For example, when searching for permutations,

enforcing $X_{ii} = I_d$ and X_{ij} doubly stochastic

is useful, since if $\mathrm{rank}(X) = d$, then the X_{ij}'s are doubly stochastic *and* orthogonal, hence they are permutations.

There are ways to accommodate more constraints in our algorithm…

For example, enforce $\langle A_i, X \rangle \geq b_i$ by penalizing

$$\min_i \ \langle A_i, X \rangle - b_i \approx -\epsilon \log\left(\sum_i e^{-\frac{\langle A_i, X \rangle - b_i}{\epsilon}}\right)$$

But it adds parameters and it reduces the competitive edge of the Riemannian approach.

(research in progress)