Certifiably Correct SLAM

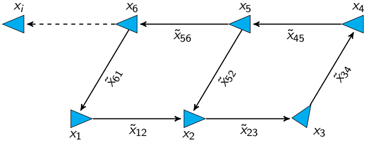
David M. Rosen

NEU EECE 5550 Mobile Robotics
Nov 19, 2021

Recap: Pose-graph SLAM

Given:

- Unknown poses $x_1, \ldots, x_n \in \mathrm{SE}(d)$
- Noisy relative measurements $\tilde{x}_{ij} = (\tilde{t}_{ij}, \tilde{R}_{ij}) \approx x_i^{-1} x_j$



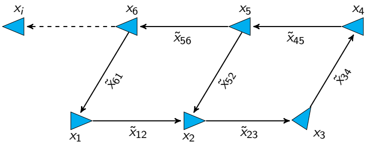
Find: An estimate $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_n)$ for the hidden states:

$$\hat{x} \in \operatorname*{argmin}_{\substack{t_i \in \mathbb{R}^d \\ R_i \in \mathrm{SO}(d)}} \sum_{(i,j) \in \vec{\mathcal{E}}} \kappa_{ij} \| R_j - R_i \tilde{R}_{ij} \|_F^2 + \tau_{ij} \| t_j - t_i - R_i \tilde{t}_{ij} \|_2^2$$

Recap: Pose-graph SLAM

Given:

- Unknown poses $x_1, \ldots, x_n \in \mathrm{SE}(d)$
- Noisy relative measurements $\tilde{x}_{ij} = (\tilde{t}_{ij}, \tilde{R}_{ij}) \approx x_i^{-1} x_j$



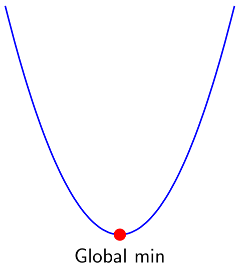
Find: An estimate $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_n)$ for the hidden states:

$$\hat{x} \in \underset{\substack{t_i \in \mathbb{R}^d \\ R_i \in \mathrm{SO}(d)}}{\operatorname{argmin}} \sum_{(i,j) \in \vec{\mathcal{E}}} \kappa_{ij} \| R_j - R_i \tilde{R}_{ij} \|_F^2 + \tau_{ij} \| t_j - t_i - R_i \tilde{t}_{ij} \|_2^2$$

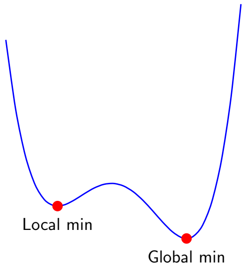
The Big Problem: This is a *high-dimensional*, *nonconvex* problem

Recap: Pose-graph SLAM

Convex



Nonconvex



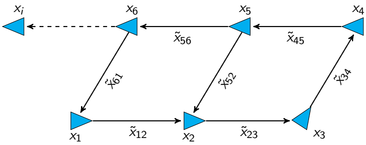
- Local min $\Rightarrow$ *global* min
- *Easy* to optimize

- Can have *many* local minima
- *Hard* to optimize (generally)

Recap: Pose-graph SLAM

Given:

- Unknown poses $x_1, \ldots, x_n \in \mathrm{SE}(d)$
- Noisy relative measurements $\tilde{x}_{ij} = (\tilde{t}_{ij}, \tilde{R}_{ij}) \approx x_i^{-1} x_j$



Find: An estimate $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_n)$ for the hidden states:

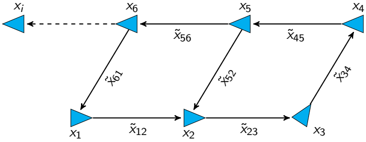
$$\hat{x} \in \operatorname*{argmin}_{\substack{t_i \in \mathbb{R}^d \\ R_i \in \mathrm{SO}(d)}} \sum_{(i,j) \in \vec{\mathcal{E}}} \kappa_{ij} \| R_j - R_i \tilde{R}_{ij} \|_F^2 + \tau_{ij} \| t_j - t_i - R_i \tilde{t}_{ij} \|_2^2$$

The Big Problem: This is a *high-dimensional*, *nonconvex* problem

Recap: Pose-graph SLAM

Given:

- Unknown poses $x_1, \ldots, x_n \in \mathrm{SE}(d)$
- Noisy relative measurements $\tilde{x}_{ij} = (\tilde{t}_{ij}, \tilde{R}_{ij}) \approx x_i^{-1} x_j$



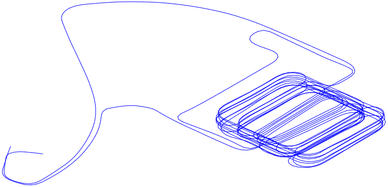
Find: An estimate $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_n)$ for the hidden states:

$$\hat{x} \in \underset{\substack{t_i \in \mathbb{R}^d \\ R_i \in \mathrm{SO}(d)}}{\operatorname{argmin}} \sum_{(i,j) \in \vec{\mathcal{E}}} \kappa_{ij} \| R_j - R_i \tilde{R}_{ij} \|_F^2 + \tau_{ij} \| t_j - t_i - R_i \tilde{t}_{ij} \|_2^2$$

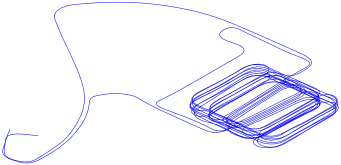
The Big Problem: This is a *high-dimensional*, *nonconvex* problem
$\Rightarrow$ **LOTS** of *local minima*
$\Rightarrow$ **No guarantees** on estimates recovered via *local search*!

The problem of nonconvexity

The problem of nonconvexity

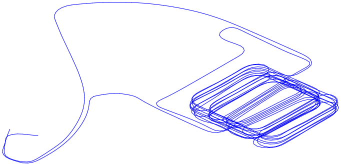


The problem of nonconvexity

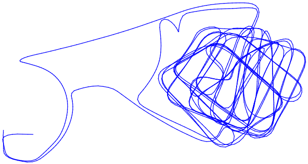


Optimal estimate

The problem of nonconvexity

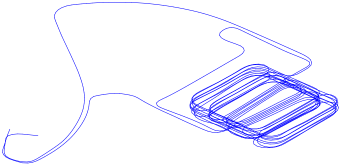


Optimal estimate

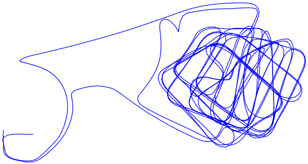


Suboptimal critical point

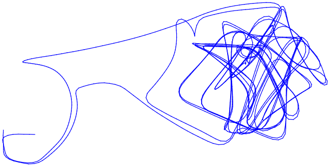
The problem of nonconvexity



Optimal estimate

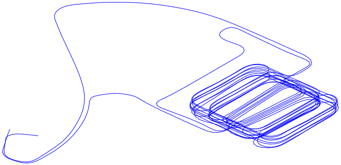


Suboptimal critical point

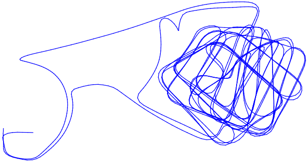


Suboptimal critical point

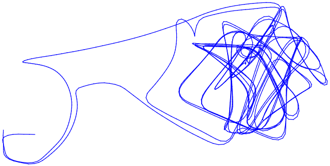
The problem of nonconvexity



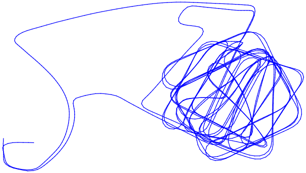
Optimal estimate



Suboptimal critical point



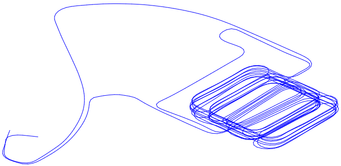
Suboptimal critical point



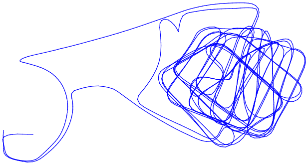
Suboptimal critical point

The problem of nonconvexity

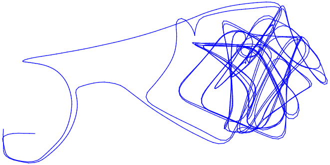
Main Question: Is it possible to *ensure* good SLAM performance?



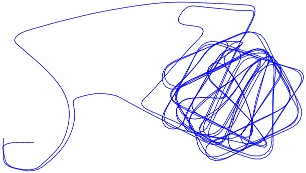
Optimal estimate



Suboptimal critical point



Suboptimal critical point



Suboptimal critical point

The Main Idea: Convex relaxation for the win!

We can't efficiently solve (NP-hard) pose-graph SLAM in general

[1] Bandeira 2016; Chandrasekaran et al. 2012; Chandrasekaran et al. 2013.

The Main Idea: Convex relaxation for the win!

We can't efficiently solve (NP-hard) pose-graph SLAM *in general*

But: Maybe we can *well-approximate* "*reasonable*" instances?

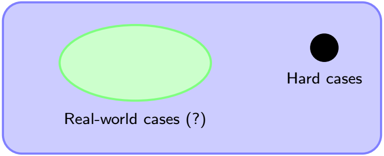
[1] Bandeira 2016; Chandrasekaran et al. 2012; Chandrasekaran et al. 2013.

The Main Idea: Convex relaxation for the win!

We can't efficiently solve (NP-hard) pose-graph SLAM *in general*

But: Maybe we can *well-approximate* "*reasonable*" instances?

Intuition: Nature is not "out to get you"



All pose-graph SLAM instances

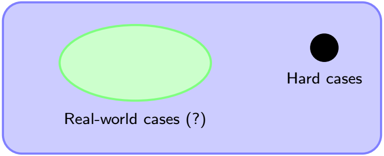
[1] Bandeira 2016; Chandrasekaran et al. 2012; Chandrasekaran et al. 2013.

The Main Idea: Convex relaxation for the win!

We can't efficiently solve (NP-hard) pose-graph SLAM *in general*

But: Maybe we can *well-approximate* "*reasonable*" instances?

Intuition: Nature is not "out to get you"



All pose-graph SLAM instances

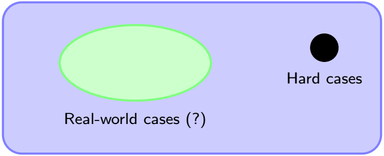
Question: How to *generate* these approximations?

[1] Bandeira 2016; Chandrasekaran et al. 2012; Chandrasekaran et al. 2013.

The Main Idea: Convex relaxation for the win!

We can't efficiently solve (NP-hard) pose-graph SLAM *in general*

But: Maybe we can *well-approximate* "*reasonable*" instances?
Intuition: Nature is not "out to get you"



All pose-graph SLAM instances

Question: How to *generate* these approximations?

One general approach: *Convex relaxation*
$\Rightarrow$ Solve a *convex approximation* of original problem

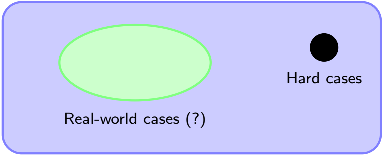
[1] Bandeira 2016; Chandrasekaran et al. 2012; Chandrasekaran et al. 2013.

The Main Idea: Convex relaxation for the win!

We can't efficiently solve (NP-hard) pose-graph SLAM *in general*

But: Maybe we can *well-approximate* "*reasonable*" instances?

Intuition: Nature is not "out to get you"



All pose-graph SLAM instances

Question: How to *generate* these approximations?

One general approach: *Convex relaxation*

$\Rightarrow$ Solve a *convex approximation* of original problem

- Standard machinery for constructing these
- Admit (some) formal guarantees on solution quality.[1]

[1] Bandeira 2016; Chandrasekaran et al. 2012; Chandrasekaran et al. 2013.

In this talk

1. **Lagrangian duality**: lower bounds for optimization problems via *convex relaxation*
2. **Solution verification**: *certificates of optimality* for pose-graph SLAM
3. **SE-Sync**: Fast *global* pose-graph optimization

A primer on Lagrangian duality

Main idea: Compute *lower bounds* for the *optimal value* of an optimization problem:

$$p^* = \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \left\{ \begin{array}{ll} g_i(x) = 0, & i = 1, \ldots, l \\ h_j(x) \leq 0, & j = 1, \ldots, m \end{array} \right\}$$

A primer on Lagrangian duality

Main idea: Compute *lower bounds* for the *optimal value* of an optimization problem:

$$p^* = \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \left\{ \begin{array}{ll} g_i(x) = 0, & i = 1, \ldots, l \\ h_j(x) \leq 0, & j = 1, \ldots, m \end{array} \right\}$$

Define the *Lagrangian*:

$$\mathcal{L} \colon \mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^m_+ \to \mathbb{R}$$

$$\mathcal{L}(x, \lambda, \nu) \triangleq f(x) + \sum_{i=1}^{l} \lambda_i g_i(x) + \sum_{j=1}^{m} \nu_j h_j(x).$$

A primer on Lagrangian duality

Main idea: Compute *lower bounds* for the *optimal value* of an optimization problem:

$$p^* = \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \begin{Bmatrix} g_i(x) = 0, & i = 1, \dots, l \\ h_j(x) \leq 0, & j = 1, \dots, m \end{Bmatrix}$$

Define the *Lagrangian*:

$$\mathcal{L} \colon \mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^m_+ \to \mathbb{R}$$

$$\mathcal{L}(x, \lambda, \nu) \triangleq f(x) + \sum_{i=1}^{l} \lambda_i g_i(x) + \sum_{j=1}^{m} \nu_j h_j(x).$$

Key observation: For any $\nu \geq 0$, λ, and feasible x:

A primer on Lagrangian duality

Main idea: Compute *lower bounds* for the *optimal value* of an optimization problem:

$$p^* = \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \left\{ \begin{array}{ll} g_i(x) = 0, & i = 1, \ldots, l \\ h_j(x) \leq 0, & j = 1, \ldots, m \end{array} \right\}$$

Define the *Lagrangian*:

$$\mathcal{L} \colon \mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^m_+ \to \mathbb{R}$$

$$\mathcal{L}(x, \lambda, \nu) \triangleq f(x) + \sum_{i=1}^{l} \lambda_i g_i(x) + \sum_{j=1}^{m} \nu_j h_j(x).$$

Key observation: For any $\nu \geq 0$, λ, and feasible x:

A primer on Lagrangian duality

Main idea: Compute *lower bounds* for the *optimal value* of an optimization problem:

$$p^* = \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \left\{ \begin{array}{ll} g_i(x) = 0, & i = 1, \ldots, l \\ h_j(x) \leq 0, & j = 1, \ldots, m \end{array} \right\}$$

Define the *Lagrangian*:

$$\mathcal{L} \colon \mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^m_+ \to \mathbb{R}$$

$$\mathcal{L}(x, \lambda, \nu) \triangleq f(x) + \sum_{i=1}^{l} \lambda_i g_i(x) + \sum_{j=1}^{m} \nu_j h_j(x).$$

Key observation: For any $\nu \geq 0$, λ, and feasible x:

$$\mathcal{L}(x, \lambda, \nu) \leq f(x).$$

The dual function

Recall: For any $\nu \geq 0$, λ, and feasible x:

$$\mathcal{L}(x, \lambda, \nu) \leq f(x).$$

The dual function

Recall: For any $\nu \geq 0$, λ, and feasible x:

$$\mathcal{L}(x, \lambda, \nu) \leq f(x).$$

Define the *dual function*:

$$d \colon \mathbb{R}^l \times \mathbb{R}^m_+ \to \mathbb{R}$$
$$d(\lambda, \nu) \triangleq \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \nu).$$

The dual function

Recall: For any $\nu \geq 0$, λ, and feasible x:

$$\mathcal{L}(x, \lambda, \nu) \leq f(x).$$

Define the *dual function*:

$$d \colon \mathbb{R}^l \times \mathbb{R}^m_+ \to \mathbb{R}$$
$$d(\lambda, \nu) \triangleq \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \nu).$$

Lagrangian duality: For all $\nu \geq 0$, λ, and feasible x:

$$d(\lambda, \nu) \triangleq \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \nu)$$

The dual function

Recall: For any $\nu \geq 0$, λ, and feasible x:

$$\mathcal{L}(x, \lambda, \nu) \leq f(x).$$

Define the *dual function*:

$$d \colon \mathbb{R}^l \times \mathbb{R}^m_+ \to \mathbb{R}$$
$$d(\lambda, \nu) \triangleq \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \nu).$$

Lagrangian duality: For all $\nu \geq 0$, λ, and feasible x:

$$d(\lambda, \nu) \triangleq \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \nu) \leq \inf_{x \text{ feasible}} \mathcal{L}(x, \lambda, \nu)$$

The dual function

Recall: For any $\nu \geq 0$, λ, and feasible x:

$$\mathcal{L}(x, \lambda, \nu) \leq f(x).$$

Define the *dual function*:

$$d \colon \mathbb{R}^l \times \mathbb{R}^m_+ \to \mathbb{R}$$
$$d(\lambda, \nu) \triangleq \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \nu).$$

Lagrangian duality: For all $\nu \geq 0$, λ, and feasible x:

$$d(\lambda, \nu) \triangleq \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \nu) \leq \inf_{x \text{ feasible}} \mathcal{L}(x, \lambda, \nu) \leq f(x).$$

The dual function

Recall: For any $\nu \geq 0$, λ, and feasible x:

$$\mathcal{L}(x, \lambda, \nu) \leq f(x).$$

Define the *dual function*:

$$d \colon \mathbb{R}^l \times \mathbb{R}^m_+ \to \mathbb{R}$$
$$d(\lambda, \nu) \triangleq \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \nu).$$

Lagrangian duality: For all $\nu \geq 0$, λ, and feasible x:

$$d(\lambda, \nu) \triangleq \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \nu) \leq \inf_{x \text{ feasible}} \mathcal{L}(x, \lambda, \nu) \leq f(x).$$

The dual function

Recall: For any $\nu \geq 0$, λ, and feasible x:

$$\mathcal{L}(x, \lambda, \nu) \leq f(x).$$

Define the *dual function*:

$$d \colon \mathbb{R}^l \times \mathbb{R}^m_+ \to \mathbb{R}$$
$$d(\lambda, \nu) \triangleq \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \nu).$$

Lagrangian duality: For all $\nu \geq 0$, λ, and feasible x:

$$d(\lambda, \nu) \triangleq \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \nu) \leq \inf_{x \text{ feasible}} \mathcal{L}(x, \lambda, \nu) \leq f(x).$$

$\Rightarrow$ Dual function $d(\lambda, \nu)$ *lower-bounds* $f(x)$ for all feasible x.

The dual function

Recall: For any $\nu \geq 0$, λ, and feasible x:

$$\mathcal{L}(x, \lambda, \nu) \leq f(x).$$

Define the *dual function*:

$$d\colon \mathbb{R}^l \times \mathbb{R}^m_+ \to \mathbb{R}$$
$$d(\lambda, \nu) \triangleq \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \nu).$$

Lagrangian duality: For all $\nu \geq 0$, λ, and feasible x:

$$d(\lambda, \nu) \triangleq \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \nu) \leq \inf_{x \text{ feasible}} \mathcal{L}(x, \lambda, \nu) \leq f(x).$$

$\Rightarrow$ Dual function $d(\lambda, \nu)$ *lower-bounds* $f(x)$ for all feasible x.

Key fact: The dual function is *always concave*.

The Lagrangian dual problem

For all $\nu \geq 0$, λ, and feasible x, we have the *lower bound*:

$$d(\lambda, \nu) \leq p^* \leq f(x).$$

How to get the *tightest* possible bound?

The Lagrangian dual problem

For all $\nu \geq 0$, λ, and feasible x, we have the *lower bound*:

$$d(\lambda, \nu) \leq p^* \leq f(x).$$

How to get the *tightest* possible bound? Maximize dual function!

Lagrangian dual problem:

$$d^* = \max_{\nu \geq 0,\, \lambda} d(\lambda, \nu)$$

The Lagrangian dual problem

For all $\nu \geq 0$, λ, and feasible x, we have the *lower bound*:

$$d(\lambda, \nu) \leq p^* \leq f(x).$$

How to get the *tightest* possible bound? Maximize dual function!

Lagrangian dual problem:

$$d^* = \max_{\nu \geq 0,\, \lambda} d(\lambda, \nu)$$

Recall: The dual function is *always concave*

The Lagrangian dual problem

For all $\nu \geq 0$, λ, and feasible x, we have the *lower bound*:

$$d(\lambda, \nu) \leq p^* \leq f(x).$$

How to get the *tightest* possible bound? Maximize dual function!

Lagrangian dual problem:

$$d^* = \max_{\nu \geq 0,\, \lambda} d(\lambda, \nu)$$

Recall: The dual function is *always concave*
$\Rightarrow$ Lagrangian dual problem is *always convex*

The Lagrangian dual problem

For all $\nu \geq 0$, λ, and feasible x, we have the *lower bound*:

$$d(\lambda, \nu) \leq p^* \leq f(x).$$

How to get the *tightest* possible bound? Maximize dual function!

Lagrangian dual problem:

$$d^* = \max_{\nu \geq 0,\, \lambda} d(\lambda, \nu)$$

Recall: The dual function is *always concave*
$\Rightarrow$ Lagrangian dual problem is *always convex*
$\Rightarrow$ We can *always* compute the *optimal lower-bound d^**

Strong duality

Primal problem

$$p^* = \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \left\{ \begin{array}{l} g(x) = 0 \\ h(x) \leq 0 \end{array} \right\}$$

Dual problem

$$d^* = \max_{\nu \geq 0,\, \lambda} d(\lambda, \nu)$$

Lagrangian duality: $d^* \leq p^*$

Strong duality

Primal problem

$$p^* = \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \left\{ \begin{matrix} g(x) = 0 \\ h(x) \leq 0 \end{matrix} \right\}$$

Dual problem

$$d^* = \max_{\nu \geq 0,\, \lambda} d(\lambda, \nu)$$

Lagrangian duality: $d^* \leq p^*$

Strong duality: $d^* = p^*$

Strong duality

Primal problem

$$p^* = \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \left\{ \begin{array}{l} g(x) = 0 \\ h(x) \leq 0 \end{array} \right\}$$

Dual problem

$$d^* = \max_{\nu \geq 0,\, \lambda} d(\lambda, \nu)$$

Lagrangian duality: $d^* \leq p^*$

Strong duality: $d^* = p^*$

Observe: If strong duality holds and x^* and (λ^*, ν^*) are primal and dual solutions, then:

$$p^* = f(x^*) = \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda^*, \nu^*) = d(\lambda^*, \nu^*) = d^*.$$

Strong duality

Primal problem

$$p^* = \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \left\{ \begin{array}{l} g(x) = 0 \\ h(x) \leq 0 \end{array} \right\}$$

Dual problem

$$d^* = \max_{\nu \geq 0,\, \lambda} d(\lambda, \nu)$$

Lagrangian duality: $d^* \leq p^*$

Strong duality: $d^* = p^*$

Observe: If strong duality holds and x^* and (λ^*, ν^*) are primal and dual solutions, then:

$$p^* = f(x^*) = \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda^*, \nu^*) = d(\lambda^*, \nu^*) = d^*.$$

Strong duality

Primal problem

$$p^* = \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \begin{Bmatrix} g(x) = 0 \\ h(x) \leq 0 \end{Bmatrix}$$

Dual problem

$$d^* = \max_{\nu \geq 0,\, \lambda} d(\lambda, \nu)$$

Lagrangian duality: $d^* \leq p^*$

Strong duality: $d^* = p^*$

Observe: If strong duality holds and x^* and (λ^*, ν^*) are primal and dual solutions, then:

$$p^* = f(x^*) = \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda^*, \nu^*) = d(\lambda^*, \nu^*) = d^*.$$

$\Rightarrow x^*$ is an *unconstrained minimizer* of $\mathcal{L}(x, \lambda^*, \nu^*)$.

Strong duality

Primal problem

$$p^* = \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \left\{ \begin{array}{l} g(x) = 0 \\ h(x) \leq 0 \end{array} \right\}$$

Dual problem

$$d^* = \max_{\nu \geq 0,\, \lambda} d(\lambda, \nu)$$

Lagrangian duality: $d^* \leq p^*$

Strong duality: $d^* = p^*$

Observe: If strong duality holds and x^* and (λ^*, ν^*) are primal and dual solutions, then:

$$p^* = f(x^*) = \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda^*, \nu^*) = d(\lambda^*, \nu^*) = d^*.$$

$\Rightarrow$ x^* is an *unconstrained minimizer* of $\mathcal{L}(x, \lambda^*, \nu^*)$.

$\Rightarrow$ Can recover *primal* solution x^* from (convex) dual solution (λ^*, ν^*) if minimizing $\mathcal{L}(x, \lambda^*, \nu^*)$ is easy.

Lagrangian duality: Summary

Primal problem

$$p^* = \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \left\{ \begin{array}{c} g(x) = 0 \\ h(x) \leq 0 \end{array} \right\}$$

Lagrangian duality: Summary

Primal problem

$$p^* = \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \left\{ \begin{aligned} g(x) &= 0 \\ h(x) &\leq 0 \end{aligned} \right\}$$

Lagrangian: $\mathcal{L}(x, \lambda, \nu) \triangleq f(x) + \lambda^{\mathsf{T}} g(x) + \nu^{\mathsf{T}} h(x).$

Lagrangian duality: Summary

Primal problem

$$p^* = \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \left\{ \begin{array}{l} g(x) = 0 \\ h(x) \leq 0 \end{array} \right\}$$

Lagrangian: $\mathcal{L}(x, \lambda, \nu) \triangleq f(x) + \lambda^\mathsf{T} g(x) + \nu^\mathsf{T} h(x)$.

Dual function: $d(\lambda, \nu) \triangleq \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \nu)$.

Lagrangian duality: Summary

Primal problem

$$p^* = \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \left\{ \begin{array}{c} g(x) = 0 \\ h(x) \leq 0 \end{array} \right\}$$

Lagrangian: $\mathcal{L}(x, \lambda, \nu) \triangleq f(x) + \lambda^\mathsf{T} g(x) + \nu^\mathsf{T} h(x)$.

Dual function: $d(\lambda, \nu) \triangleq \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \nu)$.

Lagrangian duality: $d(\lambda, \nu) \leq f(x)$ for all $\nu \geq 0$, λ, feasible x
$\Rightarrow$ Dual function provides *lower bounds* on p^*

Lagrangian duality: Summary

Primal problem

$$p^* = \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \left\{ \begin{array}{c} g(x) = 0 \\ h(x) \leq 0 \end{array} \right\}$$

Dual problem

$$d^* = \max_{\nu \geq 0,\, \lambda} d(\lambda, \nu)$$

Lagrangian: $\mathcal{L}(x, \lambda, \nu) \triangleq f(x) + \lambda^\mathsf{T} g(x) + \nu^\mathsf{T} h(x)$.

Dual function: $d(\lambda, \nu) \triangleq \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \nu)$.

Lagrangian duality: $d(\lambda, \nu) \leq f(x)$ for all $\nu \geq 0$, λ, feasible x
$\Rightarrow$ Dual function provides *lower bounds* on p^*

Dual problem:

- Provides *tightest* achievable lower bound $d^* \leq p^*$
- $d(\lambda, \nu)$ *concave* $\Rightarrow$ dual problem *convex*

In this talk

1. **Lagrangian duality**: lower bounds for optimization problems via *convex relaxation*
2. **Solution verification**: *certificates of optimality* for pose-graph SLAM
3. **SE-Sync**: Fast *global* pose-graph optimization

Assessing the quality of a pose-graph SLAM solution

Main idea: Use Lagrangian duality to *bound the suboptimality* of a candidate pose-graph SLAM solution.

Assessing the quality of a pose-graph SLAM solution

Main idea: Use Lagrangian duality to *bound the suboptimality* of a candidate pose-graph SLAM solution.

Let $x \in \mathrm{SE}(d)^n$ be a candidate solution of pose-graph MLE:

$$p^* = \min_{\substack{t_i \in \mathbb{R}^d \\ R_i \in \mathrm{SO}(d)}} \sum_{(i,j) \in \vec{\mathcal{E}}} \kappa_{ij} \| R_j - R_i \tilde{R}_{ij} \|_F^2 + \tau_{ij} \left\| t_j - t_i - R_i \tilde{t}_{ij} \right\|_2^2 .$$

Assessing the quality of a pose-graph SLAM solution

Main idea: Use Lagrangian duality to *bound the suboptimality* of a candidate pose-graph SLAM solution.

Let $x \in \mathrm{SE}(d)^n$ be a candidate solution of pose-graph MLE:

$$p^* = \min_{\substack{t_i \in \mathbb{R}^d \\ R_i \in \mathrm{SO}(d)}} \sum_{(i,j) \in \vec{\mathcal{E}}} \kappa_{ij} \| R_j - R_i \tilde{R}_{ij} \|_F^2 + \tau_{ij} \| t_j - t_i - R_i \tilde{t}_{ij} \|_2^2 .$$

Lagrangian duality: $d^* \leq p^* \leq f(x)$.

Assessing the quality of a pose-graph SLAM solution

Main idea: Use Lagrangian duality to *bound the suboptimality* of a candidate pose-graph SLAM solution.

Let $x \in \mathrm{SE}(d)^n$ be a candidate solution of pose-graph MLE:

$$p^* = \min_{\substack{t_i \in \mathbb{R}^d \\ R_i \in \mathrm{SO}(d)}} \sum_{(i,j) \in \vec{\mathcal{E}}} \kappa_{ij} \| R_j - R_i \tilde{R}_{ij} \|_F^2 + \tau_{ij} \left\| t_j - t_i - R_i \tilde{t}_{ij} \right\|_2^2 .$$

Lagrangian duality: $d^* \leq p^* \leq f(x)$. This implies:

$$f(x) - p^* \leq f(x) - d^*.$$

Assessing the quality of a pose-graph SLAM solution

Main idea: Use Lagrangian duality to *bound the suboptimality* of a candidate pose-graph SLAM solution.

Let $x \in \mathrm{SE}(d)^n$ be a candidate solution of pose-graph MLE:

$$p^* = \min_{\substack{t_i \in \mathbb{R}^d \\ R_i \in \mathrm{SO}(d)}} \sum_{(i,j) \in \vec{\mathcal{E}}} \kappa_{ij} \| R_j - R_i \tilde{R}_{ij} \|_F^2 + \tau_{ij} \left\| t_j - t_i - R_i \tilde{t}_{ij} \right\|_2^2.$$

Lagrangian duality: $d^* \leq p^* \leq f(x)$. This implies:

$$f(x) - p^* \leq f(x) - d^*.$$

- Left-hand side is *suboptimality* of x.

Assessing the quality of a pose-graph SLAM solution

Main idea: Use Lagrangian duality to *bound the suboptimality* of a candidate pose-graph SLAM solution.

Let $x \in \mathrm{SE}(d)^n$ be a candidate solution of pose-graph MLE:

$$p^* = \min_{\substack{t_i \in \mathbb{R}^d \\ R_i \in \mathrm{SO}(d)}} \sum_{(i,j) \in \vec{\mathcal{E}}} \kappa_{ij} \|R_j - R_i \tilde{R}_{ij}\|_F^2 + \tau_{ij} \left\| t_j - t_i - R_i \tilde{t}_{ij} \right\|_2^2.$$

Lagrangian duality: $d^* \leq p^* \leq f(x)$. This implies:

$$f(x) - p^* \leq f(x) - d^*.$$

- Left-hand side is *suboptimality* of x.
- Right-hand side can be computed from (convex) *dual problem*.

Assessing the quality of a pose-graph SLAM solution

Main idea: Use Lagrangian duality to *bound the suboptimality* of a candidate pose-graph SLAM solution.

Let $x \in \mathrm{SE}(d)^n$ be a candidate solution of pose-graph MLE:

$$p^* = \min_{\substack{t_i \in \mathbb{R}^d \\ R_i \in \mathrm{SO}(d)}} \sum_{(i,j) \in \vec{\mathcal{E}}} \kappa_{ij} \| R_j - R_i \tilde{R}_{ij} \|_F^2 + \tau_{ij} \left\| t_j - t_i - R_i \tilde{t}_{ij} \right\|_2^2 .$$

Lagrangian duality: $d^* \leq p^* \leq f(x)$. This implies:

$$f(x) - p^* \leq f(x) - d^*.$$

- Left-hand side is *suboptimality* of x.
- Right-hand side can be computed from (convex) *dual problem*.

$\Rightarrow$ We can *bound global suboptimality* of x using d^*!

Deriving the dual problem I: Simplifying the MLE

Pose-graph MLE

$$p^*_{\mathrm{MLE}} = \min_{\substack{t_i \in \mathbb{R}^d \\ R_i \in \mathrm{SO}(d)}} \sum_{(i,j) \in \vec{\mathcal{E}}} \kappa_{ij} \| R_j - R_i \tilde{R}_{ij} \|_F^2 + \tau_{ij} \left\| t_j - t_i - R_i \tilde{t}_{ij} \right\|_2^2$$

Deriving the dual problem I: Simplifying the MLE

Pose-graph MLE

$$p^*_{\mathrm{MLE}} = \min_{\substack{t_i \in \mathbb{R}^d \\ R_i \in \mathrm{SO}(d)}} \sum_{(i,j) \in \vec{\mathcal{E}}} \kappa_{ij} \|R_j - R_i \tilde{R}_{ij}\|_F^2 + \tau_{ij} \|t_j - t_i - R_i \tilde{t}_{ij}\|_2^2$$

Deriving the dual problem I: Simplifying the MLE

Pose-graph MLE

$$p^*_{\text{MLE}} = \min_{\substack{t_i \in \mathbb{R}^d \\ R_i \in \mathrm{SO}(d)}} \sum_{(i,j) \in \vec{\mathcal{E}}} \kappa_{ij} \| R_j - R_i \tilde{R}_{ij} \|_F^2 + \tau_{ij} \| t_j - t_i - R_i \tilde{t}_{ij} \|_2^2$$

Solving for $t \triangleq (t_1, \ldots, t_n)$ in terms of $R \triangleq (R_1, \ldots, R_n)$ using a generalized Schur complement...

Deriving the dual problem I: Simplifying the MLE

Pose-graph MLE

$$p^*_{\text{MLE}} = \min_{\substack{t_i \in \mathbb{R}^d \\ R_i \in \text{SO}(d)}} \sum_{(i,j) \in \vec{\mathcal{E}}} \kappa_{ij} \| R_j - R_i \tilde{R}_{ij} \|_F^2 + \tau_{ij} \left\| t_j - t_i - R_i \tilde{t}_{ij} \right\|_2^2$$

Solving for $t \triangleq (t_1, \ldots, t_n)$ in terms of $R \triangleq (R_1, \ldots, R_n)$ using a generalized Schur complement...

[Lots of algebra...]

Deriving the dual problem I: Simplifying the MLE

Pose-graph MLE

$$p^*_{\text{MLE}} = \min_{\substack{t_i \in \mathbb{R}^d \\ R_i \in \text{SO}(d)}} \sum_{(i,j)\in\vec{\mathcal{E}}} \kappa_{ij} \| R_j - R_i \tilde{R}_{ij} \|_F^2 + \tau_{ij} \left\| t_j - t_i - R_i \tilde{t}_{ij} \right\|_2^2$$

Solving for $t \triangleq (t_1, \ldots, t_n)$ in terms of $R \triangleq (R_1, \ldots, R_n)$ using a generalized Schur complement:

Simplified pose-graph MLE

$$p^*_{\text{MLE}} = \min_{R \in \text{SO}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

Deriving the dual problem II: Orthogonal relaxation

Simplified pose-graph MLE

$$p^*_{\text{MLE}} = \min_{R \in \text{SO}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

Deriving the dual problem II: Orthogonal relaxation

Simplified pose-graph MLE

$$p^*_{\text{MLE}} = \min_{R \in \text{SO}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$$\text{SO}(d) = \left\{ R \in \mathbb{R}^{d \times d} \mid R^\mathsf{T} R = I_d,\ \det(R) = +1 \right\}$$

Deriving the dual problem II: Orthogonal relaxation

Simplified pose-graph MLE

$$p^*_{\text{MLE}} = \min_{R \in \text{SO}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$$\text{SO}(d) = \left\{ R \in \mathbb{R}^{d \times d} \mid R^\mathsf{T} R = I_d, \ \det(R) = +1 \right\}$$

Relaxing the determinantal constraint...

Deriving the dual problem II: Orthogonal relaxation

Simplified pose-graph MLE

$$p^*_{\mathrm{MLE}} = \min_{R \in \mathrm{SO}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$$\mathrm{SO}(d) = \left\{ R \in \mathbb{R}^{d \times d} \mid R^\mathsf{T} R = I_d, \ \det(R) = +1 \right\}$$

Relaxing the determinantal constraint:

Orthogonally-relaxed MLE

$$p_{\mathrm{O}^*=\min_{R \in \mathrm{O}(d)^n}} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

NB: $\mathrm{SO}(d)^n \subset \mathrm{O}(d)^n \Rightarrow p_{\mathrm{O}^* \le p^*_{\mathrm{MLE}}}$.

Deriving the dual problem III: Constructing the dual

Relaxed MLE (intrinsic)

$$p_{O^*} = \min_{R \in O(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

Deriving the dual problem III: Constructing the dual

Relaxed MLE (intrinsic)

$$p_{\mathrm{O}^*} = \min_{R \in \mathrm{O}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$$\mathrm{O}(d) = \left\{ R \in \mathbb{R}^{d \times d} \mid R^\mathsf{T} R = I_d \right\}$$

Deriving the dual problem III: Constructing the dual

Relaxed MLE (intrinsic)

$$p_{\mathrm{O}^*} = \min_{R \in \mathrm{O}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$\Longrightarrow$

Relaxed MLE (extrinsic)

$$p_{\mathrm{O}^*} = \min_{R \in \mathbb{R}^{d \times dn}} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$
$$\text{s.t.} \quad R_i^\mathsf{T} R_i = I_d \quad i = 1, \dots, n$$

$$\mathrm{O}(d) = \left\{ R \in \mathbb{R}^{d \times d} \mid R^\mathsf{T} R = I_d \right\}$$

Deriving the dual problem III: Constructing the dual

Relaxed MLE (intrinsic)

$$p_{\mathrm{O}^*} = \min_{R \in \mathrm{O}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$\Longrightarrow$

Relaxed MLE (extrinsic)

$$p_{\mathrm{O}^*} = \min_{R \in \mathbb{R}^{d \times dn}} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$$\text{s.t.} \quad R_i^\mathsf{T} R_i = I_d \quad i = 1, \ldots, n$$

Dual function:

$$d(\Lambda_1, \ldots, \Lambda_n) \triangleq \inf_{R \in \mathbb{R}^{d \times dn}} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R) + \sum_{i=1}^{n} \operatorname{tr}\left(\Lambda_i (I_d - R_i^\mathsf{T} R_i)\right)$$

Deriving the dual problem III: Constructing the dual

Relaxed MLE (intrinsic)

$$p_{\mathrm{O}^*} = \min_{R \in \mathrm{O}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$\Longrightarrow$

Relaxed MLE (extrinsic)

$$p_{\mathrm{O}^*} = \min_{R \in \mathbb{R}^{d \times dn}} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$
$$\text{s.t.} \quad R_i^\mathsf{T} R_i = I_d \quad i = 1, \ldots, n$$

Dual function:

$$d(\Lambda_1, \ldots, \Lambda_n) \triangleq \inf_{R \in \mathbb{R}^{d \times dn}} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R) + \sum_{i=1}^{n} \operatorname{tr}\left(\Lambda_i (I_d - R_i^\mathsf{T} R_i)\right)$$

Matricized form: For $\Lambda = \operatorname{Diag}(\Lambda_1, \ldots, \Lambda_n)$:

$$d(\Lambda) = \inf_{R \in \mathbb{R}^{d \times dn}} \operatorname{tr}\left((\tilde{Q} - \Lambda) R^\mathsf{T} R\right) + \operatorname{tr}(\Lambda)$$

Deriving the dual problem III: Constructing the dual

Relaxed MLE (intrinsic)

$$p_{O^*} = \min_{R \in O(d)^n} \operatorname{tr}(\tilde{Q} R^T R)$$

$\Longrightarrow$

Relaxed MLE (extrinsic)

$$p_{O^*} = \min_{R \in \mathbb{R}^{d \times dn}} \operatorname{tr}(\tilde{Q} R^T R)$$
$$\text{s.t.} \quad R_i^T R_i = I_d \quad i = 1, \dots, n$$

Dual function:

$$d(\Lambda_1, \dots, \Lambda_n) \triangleq \inf_{R \in \mathbb{R}^{d \times dn}} \operatorname{tr}(\tilde{Q} R^T R) + \sum_{i=1}^{n} \operatorname{tr}\left(\Lambda_i (I_d - R_i^T R_i)\right)$$

Matricized form: For $\Lambda = \operatorname{Diag}(\Lambda_1, \dots, \Lambda_n)$:

$$d(\Lambda) = \inf_{R \in \mathbb{R}^{d \times dn}} \operatorname{tr}\left((\tilde{Q} - \Lambda) R^T R\right) + \operatorname{tr}(\Lambda)$$

Deriving the dual problem III: Constructing the dual

Relaxed MLE (intrinsic)

$$p_{\mathrm{O}^*} = \min_{R \in \mathrm{O}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$\Longrightarrow$

Relaxed MLE (extrinsic)

$$p_{\mathrm{O}^*} = \min_{R \in \mathbb{R}^{d \times dn}} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$$\text{s.t.} \quad R_i^\mathsf{T} R_i = I_d \quad i = 1, \ldots, n$$

Dual function:

$$d(\Lambda_1, \ldots, \Lambda_n) \triangleq \inf_{R \in \mathbb{R}^{d \times dn}} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R) + \sum_{i=1}^{n} \operatorname{tr}\left(\Lambda_i (I_d - R_i^\mathsf{T} R_i)\right)$$

Matricized form: For $\Lambda = \operatorname{Diag}(\Lambda_1, \ldots, \Lambda_n)$:

$$d(\Lambda) = \inf_{R \in \mathbb{R}^{d \times dn}} \operatorname{tr}\left((\tilde{Q} - \Lambda) R^\mathsf{T} R\right) + \operatorname{tr}(\Lambda) = \begin{cases} \operatorname{tr}(\Lambda), & \tilde{Q} - \Lambda \succeq 0 \\ -\infty, & \tilde{Q} - \Lambda \not\succeq 0 \end{cases}$$

Deriving the dual problem III: Constructing the dual

Relaxed MLE (intrinsic)

$$p_{\mathrm{O}^*} = \min_{R \in \mathrm{O}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$\Longrightarrow$

Relaxed MLE (extrinsic)

$$p_{\mathrm{O}^*} = \min_{R \in \mathbb{R}^{d \times dn}} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$$\text{s.t.} \quad R_i^\mathsf{T} R_i = I_d \quad i = 1, \dots, n$$

Dual function:

$$d(\Lambda_1, \dots, \Lambda_n) \triangleq \inf_{R \in \mathbb{R}^{d \times dn}} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R) + \sum_{i=1}^{n} \operatorname{tr}\left(\Lambda_i (I_d - R_i^\mathsf{T} R_i)\right)$$

Matricized form: For $\Lambda = \operatorname{Diag}(\Lambda_1, \dots, \Lambda_n)$:

$$d(\Lambda) = \inf_{R \in \mathbb{R}^{d \times dn}} \operatorname{tr}\left((\tilde{Q} - \Lambda) R^\mathsf{T} R\right) + \operatorname{tr}(\Lambda) = \begin{cases} \operatorname{tr}(\Lambda), & \tilde{Q} - \Lambda \succeq 0 \\ -\infty, & \tilde{Q} - \Lambda \not\succeq 0 \end{cases}$$

Deriving the dual problem III: Constructing the dual

Relaxed MLE (intrinsic)

$$p_{\mathrm{O}^*} = \min_{R \in \mathrm{O}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$\implies$

Relaxed MLE (extrinsic)

$$p_{\mathrm{O}^*} = \min_{R \in \mathbb{R}^{d \times dn}} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$$\text{s.t.} \quad R_i^\mathsf{T} R_i = I_d \quad i = 1, \dots, n$$

Dual function:

$$d(\Lambda_1, \dots, \Lambda_n) \triangleq \inf_{R \in \mathbb{R}^{d \times dn}} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R) + \sum_{i=1}^{n} \operatorname{tr}\left(\Lambda_i (I_d - R_i^\mathsf{T} R_i)\right)$$

Matricized form: For $\Lambda = \operatorname{Diag}(\Lambda_1, \dots, \Lambda_n)$:

$$d(\Lambda) = \inf_{R \in \mathbb{R}^{d \times dn}} \operatorname{tr}\left((\tilde{Q} - \Lambda) R^\mathsf{T} R\right) + \operatorname{tr}(\Lambda) = \begin{cases} \operatorname{tr}(\Lambda), & \tilde{Q} - \Lambda \succeq 0 \\ -\infty, & \tilde{Q} - \Lambda \not\succeq 0 \end{cases}$$

Dual problem:

$$d^* = \max_{\Lambda \in \mathrm{SBD}(d,n)} \operatorname{tr}(\Lambda) \quad \text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0$$

Deriving the dual problem IV: Putting it all together

Pose-graph MLE

$$p^*_{\text{MLE}} = \min_{X \in \text{SE}(d)^n} \operatorname{tr}(\tilde{M} X^\mathsf{T} X)$$

$$\Downarrow$$

Orthogonally-relaxed MLE

$$p_{O^*=\min_{R \in O(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)}$$

$$\Downarrow$$

Lagrangian dual

$$d^* = \max_{\Lambda \in \text{SBD}(d,n)} \operatorname{tr}(\Lambda)$$
$$\text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0$$

Lower bounds: $d^* \leq p_{O^* \leq p^*_{\text{MLE}}}$

Suboptimality upper bound:

$$f(x) - p^*_{\text{MLE}} \leq f(x) - d^* \quad \forall x \in \text{SE}(d)^n$$

Deriving the dual problem IV: Putting it all together

Pose-graph MLE

$$p^*_{\text{MLE}} = \min_{X \in \text{SE}(d)^n} \operatorname{tr}(\tilde{M} X^\mathsf{T} X)$$

$\Downarrow$

Orthogonally-relaxed MLE

$$p_{\text{O}^*} = \min_{R \in \text{O}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$\Downarrow$

Lagrangian dual

$$d^* = \max_{\Lambda \in \text{SBD}(d,n)} \operatorname{tr}(\Lambda) \quad \text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0$$

Lower bounds: $d^* \leq p_{\text{O}^*} \leq p^*_{\text{MLE}}$

Suboptimality upper bound:

$$f(x) - p^*_{\text{MLE}} \leq f(x) - d^* \quad \forall x \in \text{SE}(d)^n$$

Key Question: How *tight* are these?

Deriving the dual problem IV: Putting it all together

Pose-graph MLE

$$p^*_{\text{MLE}} = \min_{X \in \text{SE}(d)^n} \operatorname{tr}(\tilde{M} X^{\mathsf{T}} X)$$

$$\Downarrow$$

Orthogonally-relaxed MLE

$$p_{\text{O}^* = \min_{R \in \text{O}(d)^n} \operatorname{tr}(\tilde{Q} R^{\mathsf{T}} R)}$$

$$\Downarrow$$

Lagrangian dual

$$d^* = \max_{\Lambda \in \text{SBD}(d,n)} \operatorname{tr}(\Lambda)$$

$$\text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0$$

Lower bounds: $d^* \leq p_{\text{O}^* \leq p^*_{\text{MLE}}}$

Suboptimality upper bound:

$$f(x) - p^*_{\text{MLE}} \leq f(x) - d^* \quad \forall x \in \text{SE}(d)^n$$

Theorem (Rosen et al. 2016)

Let $\underline{Q}$ be the data matrix constructed from noiseless measurements. Then $\exists \beta \triangleq \beta(\underline{Q}) > 0$ s.t. if $\|\underline{Q} - \tilde{Q}\| < \beta$, then $d^* = p_{\text{O}^* = p^*_{\text{MLE}}}$.

Deriving the dual problem IV: Putting it all together

Pose-graph MLE

$$p^*_{\text{MLE}} = \min_{X \in \text{SE}(d)^n} \operatorname{tr}(\tilde{M} X^\mathsf{T} X)$$

$\Downarrow$

Orthogonally-relaxed MLE

$$p_{\text{O}^*=\min_{R \in \text{O}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)}$$

$\Downarrow$

Lagrangian dual

$$d^* = \max_{\Lambda \in \text{SBD}(d,n)} \operatorname{tr}(\Lambda) \quad \text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0$$

Lower bounds: $d^* \leq p_{\text{O}^* \leq p^*_{\text{MLE}}}$

Suboptimality upper bound:

$$f(x) - p^*_{\text{MLE}} \leq f(x) - d^* \quad \forall x \in \text{SE}(d)^n$$

Theorem (Rosen et al. 2016)

Let $\underline{Q}$ be the data matrix constructed from noiseless measurements. Then $\exists \beta \triangleq \beta(\underline{Q}) > 0$ s.t. if $\|\underline{Q} - \tilde{Q}\| < \beta$, then $d^* = p_{\text{O}^*=p^*_{\text{MLE}}}$.

- *Guarantees strong duality* for sufficiently small noise
- $f(x) = d^*$ *certifies* x's optimality

Pose-graph solution verification (version 1.0)[2]

Input: Candidate pose-graph SLAM solution $x = (t, R) \in \mathrm{SE}(d)^n$
Output: Upper-bound on x's suboptimality

- Solve dual problem:

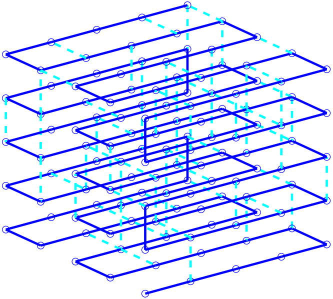
$$d^* = \max_{\Lambda \in \mathrm{SBD}(d,n)} \operatorname{tr}(\Lambda)$$
$$\text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0.$$

- **Return** $f(x) - d^*$

[2] L. Carlone et al. "Lagrangian Duality in 3D SLAM: Verification Techniques and Optimal Solutions". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. Hamburg, Germany, Sept. 2015.

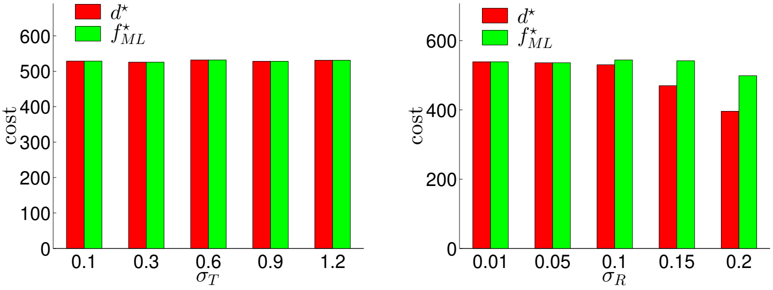
Experiments: When does strong duality hold?

We simulate a small 3D grid map, varying measurement noise...



Experiments: When does strong duality hold?

Tightness of the relaxation exhibits a *phase transition* as a function of the rotational measurement noise:

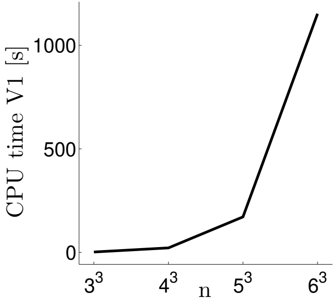


(**NB:** .1 rad $\approx$ 6 degrees)

Technical challenge: Scalability

General-purpose semidefinite programming solvers don't scale well.

$$d^* = \max_{\Lambda \in \mathrm{SBD}(d,n)} \operatorname{tr}(\Lambda) \quad \text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0$$



Typical pose-graphs are 1–2 orders of magnitude larger than this.

Exploiting strong duality

Assume:

1. Strong duality holds
2. $R^* \in \mathrm{SO}(d)^n$ and $\Lambda^* \in \mathrm{SBD}(d, n)$ are primal-dual optimal pair

Exploiting strong duality

Assume:

1. Strong duality holds
2. $R^* \in \mathrm{SO}(d)^n$ and $\Lambda^* \in \mathrm{SBD}(d,n)$ are primal-dual optimal pair

Then: R^* is an *unconstrained minimizer* of $\mathcal{L}(R, \Lambda^*)$, where:

$$\mathcal{L}(R, \Lambda) \triangleq \mathrm{tr}(\tilde{Q}R^\mathsf{T}R) + \sum_{i=1}^{n} \mathrm{tr}\left(\Lambda_i(I_d - R_i^\mathsf{T}R_i)\right).$$

Exploiting strong duality

Assume:

1. Strong duality holds
2. $R^* \in \mathrm{SO}(d)^n$ and $\Lambda^* \in \mathrm{SBD}(d,n)$ are primal-dual optimal pair

Then: R^* is an *unconstrained minimizer* of $\mathcal{L}(R, \Lambda^*)$, where:

$$\mathcal{L}(R, \Lambda) \triangleq \mathrm{tr}(\tilde{Q} R^\mathsf{T} R) + \sum_{i=1}^{n} \mathrm{tr}\left(\Lambda_i (I_d - R_i^\mathsf{T} R_i)\right).$$

Therefore:

$$\nabla_R \mathcal{L}(R^*, \Lambda^*) = 0.$$

Since Lagrangian is *affine* in Λ, this is *linear* equation in Λ^*.

Exploiting strong duality

Assume:

1. Strong duality holds
2. $R^* \in \mathrm{SO}(d)^n$ and $\Lambda^* \in \mathrm{SBD}(d, n)$ are primal-dual optimal pair

Then: R^* is an *unconstrained minimizer* of $\mathcal{L}(R, \Lambda^*)$, where:

$$\mathcal{L}(R, \Lambda) \triangleq \operatorname{tr}(\tilde{Q} R^\mathsf{T} R) + \sum_{i=1}^{n} \operatorname{tr}\left(\Lambda_i (I_d - R_i^\mathsf{T} R_i)\right).$$

Therefore:

$$\nabla_R \mathcal{L}(R^*, \Lambda^*) = 0.$$

Since Lagrangian is *affine* in Λ, this is *linear* equation in Λ^*.

Therefore: *If* $R \in \mathrm{SO}(d)^n$ is optimal, *and* strong duality holds, we can *construct* the corresponding dual certificate Λ from R:

$$\Lambda(R) = \frac{1}{2} \operatorname{BlockDiag}_d \left(\tilde{Q} R^\mathsf{T} R + R^\mathsf{T} R \tilde{Q}\right).$$

Pose-graph solution verification (version 2.0)[3]

Input: Candidate pose-graph SLAM solution $x = (t, R) \in \mathrm{SE}(d)^n$
Output: Upper-bound on x's suboptimality

- Construct candidate dual certificate:

$$\Lambda(R) = \frac{1}{2}\,\mathrm{BlockDiag}_d\left(\tilde{Q}R^\mathsf{T}R + R^\mathsf{T}R\tilde{Q}\right).$$

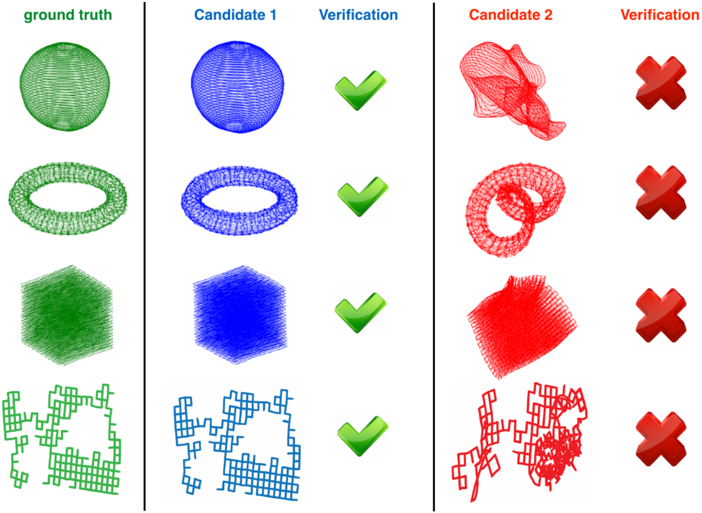
- Compute minimum eigenvalue $\lambda_{\min}$ of $\tilde{Q} - \Lambda(R)$.

- If $\lambda_{\min} \geq 0$ ($\Leftrightarrow \Lambda(R)$ is dual feasible):
  Return $f(x) - \mathrm{tr}(\Lambda(R))$

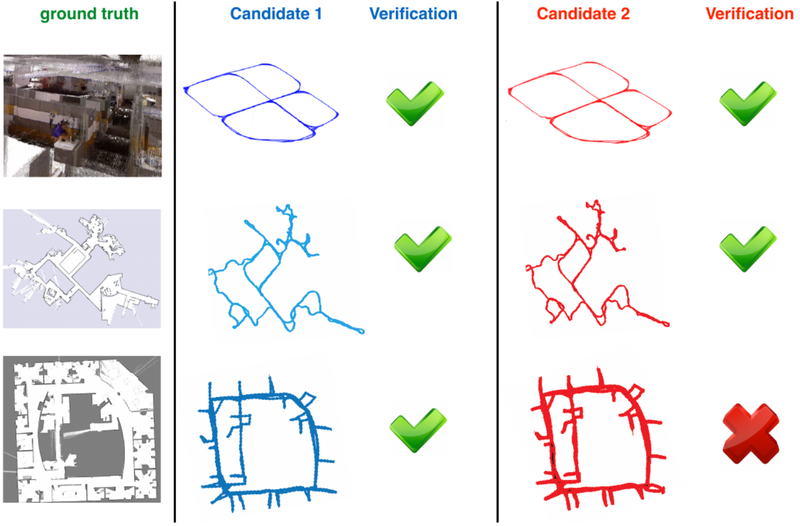
- Else:
  Return $+\infty$

[3] L. Carlone et al. "Lagrangian Duality in 3D SLAM: Verification Techniques and Optimal Solutions". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. Hamburg, Germany, Sept. 2015.

Experiments: Simulated data



Experiments: Real data



In this talk

1. **Lagrangian duality**: lower bounds for optimization problems via *convex relaxation*
2. **Solution verification**: *certificates of optimality* for pose-graph SLAM
3. **SE-Sync**: Fast *global* pose-graph optimization

From verification to global optimization

The story so far: We can *certify* the optimality of $\hat{x} \in \mathrm{SE}(d)^n$ whenever *strong duality* holds.

But: How can we *obtain* such an $\hat{x}$?

From verification to global optimization

The story so far: We can *certify* the optimality of $\hat{x} \in \mathrm{SE}(d)^n$ whenever *strong duality* holds.

But: How can we *obtain* such an $\hat{x}$?

Observe: Any $\hat{x}$ that could be *certified* (using strong duality) could also be *directly computed* (from a solution Λ of the dual).

From verification to global optimization

The story so far: We can *certify* the optimality of $\hat{x} \in \mathrm{SE}(d)^n$ whenever *strong duality* holds.

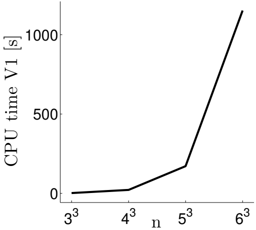
But: How can we *obtain* such an $\hat{x}$?

Observe: Any $\hat{x}$ that could be *certified* (using strong duality) could also be *directly computed* (from a solution Λ of the dual).

Key question: Can we do this *efficiently*?

SDP redux: What makes SDPs hard?

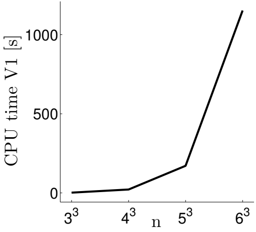
$$d^* = \max_{\Lambda \in \mathrm{SBD}(d,n)} \operatorname{tr}(\Lambda)$$
$$\text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0$$



[4] L. Vandenberghe et al. "Semidefinite Programming". In: *SIAM Review* 38.1 (Mar. 1996), pp. 49–95.

SDP redux: What makes SDPs hard?

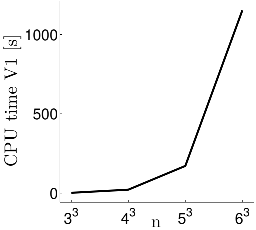
$$d^* = \max_{\Lambda \in \mathrm{SBD}(d,n)} \operatorname{tr}(\Lambda)$$
$$\text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0$$



[4] L. Vandenberghe et al. "Semidefinite Programming". In: *SIAM Review* 38.1 (Mar. 1996), pp. 49–95.

SDP redux: What makes SDPs hard?

$$d^* = \max_{\Lambda \in \mathrm{SBD}(d,n)} \operatorname{tr}(\Lambda)$$
$$\text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0$$

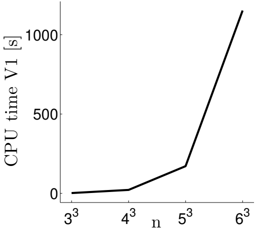


General SDP of order n:

[4] L. Vandenberghe et al. "Semidefinite Programming". In: *SIAM Review* 38.1 (Mar. 1996), pp. 49–95.

SDP redux: What makes SDPs hard?

$$d^* = \max_{\Lambda \in \mathrm{SBD}(d,n)} \mathrm{tr}(\Lambda)$$
$$\text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0$$



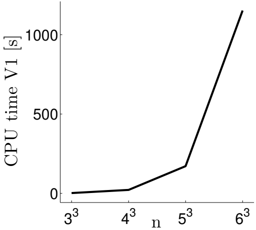
General SDP of order n:

$\Rightarrow$ Matrices have dimension $O(n^2)$

[4] L. Vandenberghe et al. "Semidefinite Programming". In: *SIAM Review* 38.1 (Mar. 1996), pp. 49–95.

SDP redux: What makes SDPs hard?

$$d^* = \max_{\Lambda \in \mathrm{SBD}(d,n)} \operatorname{tr}(\Lambda)$$
$$\text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0$$



General SDP of order n:

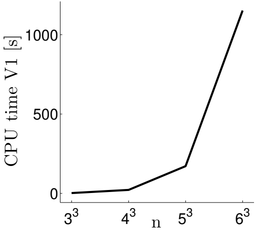
$\Rightarrow$ Matrices have dimension $O(n^2)$

$\Rightarrow$ Newton system has dimension $O(n^4)$[4]

[4] L. Vandenberghe et al. "Semidefinite Programming". In: *SIAM Review* 38.1 (Mar. 1996), pp. 49–95.

SDP redux: What makes SDPs hard?

$$d^* = \max_{\Lambda \in \mathrm{SBD}(d,n)} \operatorname{tr}(\Lambda)$$
$$\text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0$$



General SDP of order n:

$\Rightarrow$ Matrices have dimension $O(n^2)$

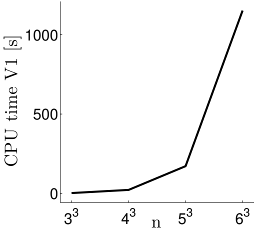
$\Rightarrow$ Newton system has dimension $O(n^4)$[4]

$\Rightarrow$ Solving Newton system (via factorization) has $O(n^6)$ cost

[4] L. Vandenberghe et al. "Semidefinite Programming". In: *SIAM Review* 38.1 (Mar. 1996), pp. 49–95.

SDP redux: What makes SDPs hard?

$$d^* = \max_{\Lambda \in \mathrm{SBD}(d,n)} \operatorname{tr}(\Lambda) \quad \text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0$$



General SDP of order n:

$\Rightarrow$ Matrices have dimension $O(n^2)$

$\Rightarrow$ Newton system has dimension $O(n^4)$[4]

$\Rightarrow$ Solving Newton system (via factorization) has $O(n^6)$ cost

But: Maybe we can build a *specialized* solver for PGO?

[4] L. Vandenberghe et al. "Semidefinite Programming". In: *SIAM Review* 38.1 (Mar. 1996), pp. 49–95.

SDPception

Simplified pose-graph MLE

$$p^*_{\text{MLE}} = \min_{R \in \text{SO}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

SDPception

Simplified pose-graph MLE

$$p^*_{\text{MLE}} = \min_{R \in \text{SO}(d)^n} \operatorname{tr}(\tilde{Q} R^{\mathsf{T}} R)$$

$$\Downarrow$$

Lagrangian dual

$$\begin{aligned} d^* = \max_{\Lambda \in \text{SBD}(d,n)} \ & \operatorname{tr}(\Lambda) \\ \text{s.t.} \quad & \tilde{Q} - \Lambda \succeq 0 \end{aligned}$$

SDPception

Simplified pose-graph MLE

$$p^*_{\mathrm{MLE}} = \min_{R \in \mathrm{SO}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$$\Downarrow$$

Lagrangian dual

$$d^* = \max_{\Lambda \in \mathrm{SBD}(d,n)} \operatorname{tr}(\Lambda) \quad \text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0$$

SDPception

Simplified pose-graph MLE

$$p^*_{\mathrm{MLE}} = \min_{R \in \mathrm{SO}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$$\Downarrow$$

Lagrangian dual

$$d^* = \max_{\Lambda \in \mathrm{SBD}(d,n)} \operatorname{tr}(\Lambda)$$
$$\text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0$$

$$\Downarrow$$

Dual of Lagrangian dual

$$d^* = \min_{Z \succeq 0} \operatorname{tr}(\tilde{Q} Z)$$
$$\text{s.t.} \quad \operatorname{Diag}_d(Z) = (I_d, \dots, I_d)$$

SDPception

Simplified pose-graph MLE

$$p^*_{\mathrm{MLE}} = \min_{R \in \mathrm{SO}(d)^n} \operatorname{tr}(\tilde{Q} R^{\mathsf{T}} R)$$

$$\Downarrow$$

Lagrangian dual

$$d^* = \max_{\Lambda \in \mathrm{SBD}(d,n)} \operatorname{tr}(\Lambda)$$
$$\text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0$$

$$\Downarrow$$

Dual of Lagrangian dual

$$d^* = \min_{Z \succeq 0} \operatorname{tr}(\tilde{Q} Z)$$
$$\text{s.t.} \quad \mathrm{Diag}_d(Z) = (I_d, \dots, I_d)$$

Question: How does the dual of the dual relate to the (primal) MLE?

SDPception

Simplified pose-graph MLE

$$p^*_{\mathrm{MLE}} = \min_{R \in \mathrm{SO}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$$\Downarrow$$

Lagrangian dual

$$\begin{aligned} d^* = & \max_{\Lambda \in \mathrm{SBD}(d,n)} \operatorname{tr}(\Lambda) \\ & \text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0 \end{aligned}$$

$$\Downarrow$$

Dual of Lagrangian dual

$$d^* = \min_{Z \succeq 0} \operatorname{tr}(\tilde{Q} Z)$$

$$\text{s.t.} \quad \mathrm{Diag}_d(Z) = (I_d, \dots, I_d)$$

SDPception

Simplified pose-graph MLE

$$p^*_{\mathrm{MLE}} = \min_{R \in \mathrm{SO}(d)^n} \operatorname{tr}(\tilde{Q} R^{\mathsf{T}} R)$$

$$\Downarrow$$

Lagrangian dual

$$d^* = \max_{\Lambda \in \mathrm{SBD}(d,n)} \operatorname{tr}(\Lambda)$$
$$\text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0$$

$$\Downarrow$$

Dual of Lagrangian dual

$$d^* = \min_{Z \succeq 0} \operatorname{tr}(\tilde{Q} Z)$$
$$\text{s.t.} \quad \operatorname{Diag}_d(Z) = (I_d, \ldots, I_d)$$

$$R^{\mathsf{T}} R = \begin{pmatrix} I_d & * & \cdots & * \\ * & I_d & & * \\ \vdots & & \ddots & \vdots \\ * & * & \cdots & I_d \end{pmatrix} \succeq 0$$

SDPception

Simplified pose-graph MLE

$$p^*_{\text{MLE}} = \min_{R \in \text{SO}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$\Downarrow$

Lagrangian dual

$$d^* = \max_{\Lambda \in \text{SBD}(d,n)} \operatorname{tr}(\Lambda)$$

$$\text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0$$

$\Downarrow$

Dual of Lagrangian dual

$$d^* = \min_{Z \succeq 0} \operatorname{tr}(\tilde{Q} Z)$$

$$\text{s.t.} \quad \text{Diag}_d(Z) = (I_d, \dots, I_d)$$

$$R^\mathsf{T} R = \begin{pmatrix} I_d & * & \cdots & * \\ * & I_d & & * \\ \vdots & & \ddots & \vdots \\ * & * & \cdots & I_d \end{pmatrix} \succeq 0$$

SDPception

Simplified pose-graph MLE

$$p^*_{\text{MLE}} = \min_{R \in \text{SO}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$$\Downarrow$$

Lagrangian dual

$$d^* = \max_{\Lambda \in \text{SBD}(d,n)} \operatorname{tr}(\Lambda)$$

$$\text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0$$

$$\Downarrow$$

Dual of Lagrangian dual

$$d^* = \min_{Z \succeq 0} \operatorname{tr}(\tilde{Q} Z)$$

$$\text{s.t.} \quad \operatorname{Diag}_d(Z) = (I_d, \dots, I_d)$$

$$R^\mathsf{T} R = \begin{pmatrix} I_d & * & \cdots & * \\ * & I_d & & * \\ \vdots & & \ddots & \vdots \\ * & * & \cdots & I_d \end{pmatrix} \succeq 0$$

$\Rightarrow$ Dual of dual is a *convex relaxation* by *expanding MLE's feasible set*

SDPception

Simplified pose-graph MLE

$$p^*_{\text{MLE}} = \min_{R \in \text{SO}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$\Downarrow$

Lagrangian dual

$$d^* = \max_{\Lambda \in \text{SBD}(d,n)} \operatorname{tr}(\Lambda) \quad \text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0$$

$\Downarrow$

Dual of Lagrangian dual

$$d^* = \min_{Z \succeq 0} \operatorname{tr}(\tilde{Q} Z) \quad \text{s.t.} \quad \operatorname{Diag}_d(Z) = (I_d, \ldots, I_d)$$

Theorem (Rosen et al. 2016)

Let $\underline{Q}$ be the data matrix constructed from noiseless measurements. Then $\exists \beta \triangleq \beta(\underline{Q}) > 0$ s.t. if $\|\underline{Q} - \tilde{Q}\| < \beta$:

- *SDP has a unique solution Z^**
- *$Z^* = {R^*}^\mathsf{T} R^*$, where $R^* \in \text{SO}(d)^n$ is a global minimizer of MLE.*

SDPception

Simplified pose-graph MLE

$$p^*_{\text{MLE}} = \min_{R \in \text{SO}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$$\Downarrow$$

Lagrangian dual

$$d^* = \max_{\Lambda \in \text{SBD}(d,n)} \operatorname{tr}(\Lambda)$$
$$\text{s.t.} \quad \tilde{Q} - \Lambda \succeq 0$$

$$\Downarrow$$

Dual of Lagrangian dual

$$d^* = \min_{Z \succeq 0} \operatorname{tr}(\tilde{Q} Z)$$
$$\text{s.t.} \quad \text{Diag}_d(Z) = (I_d, \ldots, I_d)$$

Theorem (Rosen et al. 2016)

Let $\underline{Q}$ be the data matrix constructed from noiseless measurements. Then $\exists \beta \triangleq \beta(\underline{Q}) > 0$ s.t. if $\|\underline{Q} - \tilde{Q}\| < \beta$:

- *SDP has a unique solution Z^**
- *$Z^* = {R^*}^\mathsf{T} R^*$, where $R^* \in \text{SO}(d)^n$ is a global minimizer of MLE.*

$\Rightarrow$ We can compute *global minima* of MLE by solving SDP

$\Rightarrow$ Solutions of SDP are *low-rank*

Exploiting low-rank structure

We expect a *low-rank* solution $Z^* = {Y^*}^\mathsf{T} Y^*$ for:

$$p^*_{\text{SDP}} = \min_{Z \in \mathbb{S}_+^{dn}} \operatorname{tr}(\tilde{Q}Z) \quad \text{s.t.} \quad \operatorname{Diag}_d(Z) = (I_d, \ldots, I_d).$$

[5] Burer et al. 2003; Burer et al. 2005; Burer et al. 2006.

Exploiting low-rank structure

We expect a *low-rank* solution $Z^* = {Y^*}^\mathsf{T} Y^*$ for:

$$p^*_{\mathrm{SDP}} = \min_{Z \in \mathbb{S}^{dn}_+} \operatorname{tr}(\tilde{Q} Z) \quad \text{s.t.} \quad \operatorname{Diag}_d(Z) = (I_d, \ldots, I_d).$$

Main idea: Replace Z with its *low-rank factorization* $Y^\mathsf{T} Y$:[5]

$$p^*_{\mathrm{SDPLR}} = \min_{Y \in \mathbb{R}^{r \times dn}} \operatorname{tr}(\tilde{Q} Y^\mathsf{T} Y) \quad \text{s.t.} \quad \operatorname{Diag}_d(Y^\mathsf{T} Y) = (I_d, \ldots, I_d)$$

[5] Burer et al. 2003; Burer et al. 2005; Burer et al. 2006.

Exploiting low-rank structure

We expect a *low-rank* solution $Z^* = {Y^*}^\mathsf{T} Y^*$ for:

$$p^*_{\mathrm{SDP}} = \min_{Z \in \mathbb{S}^{dn}_+} \operatorname{tr}(\tilde{Q} Z) \quad \text{s.t.} \quad \operatorname{Diag}_d(Z) = (I_d, \dots, I_d).$$

Main idea: Replace Z with its *low-rank factorization* $Y^\mathsf{T} Y$:[5]

$$p^*_{\mathrm{SDPLR}} = \min_{Y \in \mathbb{R}^{r \times dn}} \operatorname{tr}(\tilde{Q} Y^\mathsf{T} Y) \quad \text{s.t.} \quad \operatorname{Diag}_d(Y^\mathsf{T} Y) = (I_d, \dots, I_d)$$

Payoffs:

- $Y^\mathsf{T} Y \succeq 0$ for *all* Y $\Rightarrow$ PSD constraint is *redundant*
  $\Rightarrow$ Rank-restricted factorization is an *NLP* (vs. SDP)

[5] Burer et al. 2003; Burer et al. 2005; Burer et al. 2006.

Exploiting low-rank structure

We expect a *low-rank* solution $Z^* = Y^{*\mathsf{T}} Y^*$ for:

$$p^*_{\mathrm{SDP}} = \min_{Z \in \mathbb{S}^{dn}_+} \operatorname{tr}(\tilde{Q} Z) \quad \text{s.t.} \quad \operatorname{Diag}_d(Z) = (I_d, \dots, I_d).$$

Main idea: Replace Z with its *low-rank factorization* $Y^{\mathsf{T}} Y$:[5]

$$p^*_{\mathrm{SDPLR}} = \min_{Y \in \mathbb{R}^{r \times dn}} \operatorname{tr}(\tilde{Q} Y^{\mathsf{T}} Y) \quad \text{s.t.} \quad \operatorname{Diag}_d(Y^{\mathsf{T}} Y) = (I_d, \dots, I_d)$$

Payoffs:

- $Y^{\mathsf{T}} Y \succeq 0$ for *all* $Y \Rightarrow$ PSD constraint is *redundant*
  $\Rightarrow$ Rank-restricted factorization is an *NLP* (vs. SDP)
- Y is *much lower-dimensional* than Z for $r \ll dn$

[5] Burer et al. 2003; Burer et al. 2005; Burer et al. 2006.

Exploiting low-rank structure

We expect a *low-rank* solution $Z^* = Y^{*\mathsf{T}} Y^*$ for:

$$p^*_{\mathrm{SDP}} = \min_{Z \in \mathbb{S}^{dn}_+} \operatorname{tr}(\tilde{Q} Z) \quad \text{s.t.} \quad \operatorname{Diag}_d(Z) = (I_d, \dots, I_d).$$

Main idea: Replace Z with its *low-rank factorization* $Y^\mathsf{T} Y$:[5]

$$p^*_{\mathrm{SDPLR}} = \min_{Y \in \mathbb{R}^{r \times dn}} \operatorname{tr}(\tilde{Q} Y^\mathsf{T} Y) \quad \text{s.t.} \quad \operatorname{Diag}_d(Y^\mathsf{T} Y) = (I_d, \dots, I_d)$$

Payoffs:

- $Y^\mathsf{T} Y \succeq 0$ for *all* Y $\Rightarrow$ PSD constraint is *redundant*
  $\Rightarrow$ Rank-restricted factorization is an *NLP* (vs. SDP)
- Y is *much lower-dimensional* than Z for $r \ll dn$
- $r \geq \operatorname{rank}(Z^*)$ for some optimal Z^* $\Rightarrow$ $(Y^*)^\mathsf{T} Y^*$ solves SDP.

[5] Burer et al. 2003; Burer et al. 2005; Burer et al. 2006.

Exploiting geometric structure[6]

Rank-restricted SDP, NLP form

$$p^*_{\mathrm{SDPLR}} = \min_{Y \in \mathbb{R}^{r \times dn}} \operatorname{tr}(\tilde{Q} Y^\mathsf{T} Y)$$

$$\text{s.t.} \quad \operatorname{BlockDiag}_d(Y^\mathsf{T} Y) = (I_d, \ldots, I_d)$$

[6] N. Boumal. "A Riemannian Low-Rank Method for Optimization Over Semidefinite Matrices with Block-Diagonal Constraints". arXiv preprint: arXiv:1506.00575v2. 2015

Exploiting geometric structure[6]

Rank-restricted SDP, NLP form

$$p^*_{\text{SDPLR}} = \min_{Y \in \mathbb{R}^{r \times dn}} \operatorname{tr}(\tilde{Q} Y^\mathsf{T} Y)$$

$$\text{s.t.} \quad \operatorname{BlockDiag}_d(Y^\mathsf{T} Y) = (I_d, \ldots, I_d)$$

The constraints are equivalent to:

$$Y_i^\mathsf{T} Y_i = I_d, \quad Y_i \in \mathbb{R}^{r \times d}.$$

[6] N. Boumal. "A Riemannian Low-Rank Method for Optimization Over Semidefinite Matrices with Block-Diagonal Constraints". arXiv preprint: arXiv:1506.00575v2. 2015

Exploiting geometric structure[6]

Rank-restricted SDP, NLP form

$$p^*_{\text{SDPLR}} = \min_{Y \in \mathbb{R}^{r \times dn}} \operatorname{tr}(\tilde{Q} Y^\mathsf{T} Y)$$

$$\text{s.t.} \quad \operatorname{BlockDiag}_d(Y^\mathsf{T} Y) = (I_d, \dots, I_d)$$

The constraints are equivalent to:

$$Y_i^\mathsf{T} Y_i = I_d, \quad Y_i \in \mathbb{R}^{r \times d}.$$

Notice: This is the definition of the *Stiefel manifold*:

$$\operatorname{St}(k, n) \triangleq \left\{ Y \in \mathbb{R}^{n \times k} \mid Y^\mathsf{T} Y = I_k \right\}.$$

[6] N. Boumal. "A Riemannian Low-Rank Method for Optimization Over Semidefinite Matrices with Block-Diagonal Constraints". arXiv preprint: arXiv:1506.00575v2. 2015

Exploiting geometric structure[6]

Rank-restricted SDP, NLP form

$$p^*_{\text{SDPLR}} = \min_{Y \in \mathbb{R}^{r \times dn}} \operatorname{tr}(\tilde{Q} Y^\mathsf{T} Y)$$

$$\text{s.t.} \quad \operatorname{BlockDiag}_d(Y^\mathsf{T} Y) = (I_d, \ldots, I_d)$$

Riemannian rank-restricted NLP

$$p^*_{\text{SDPLR}} = \min_{Y \in \operatorname{St}(d,r)^n} \operatorname{tr}(\tilde{Q} Y^\mathsf{T} Y)$$

The constraints are equivalent to:

$$Y_i^\mathsf{T} Y_i = I_d, \quad Y_i \in \mathbb{R}^{r \times d}.$$

Notice: This is the definition of the *Stiefel manifold*:

$$\operatorname{St}(k, n) \triangleq \left\{ Y \in \mathbb{R}^{n \times k} \mid Y^\mathsf{T} Y = I_k \right\}.$$

[6] N. Boumal. "A Riemannian Low-Rank Method for Optimization Over Semidefinite Matrices with Block-Diagonal Constraints". arXiv preprint: arXiv:1506.00575v2. 2015

Exploiting geometric structure[6]

Rank-restricted SDP, NLP form

$$p^*_{\mathrm{SDPLR}} = \min_{Y \in \mathbb{R}^{r \times dn}} \operatorname{tr}(\tilde{Q} Y^\mathsf{T} Y)$$

$$\text{s.t.} \quad \operatorname{BlockDiag}_d(Y^\mathsf{T} Y) = (I_d, \dots, I_d)$$

Riemannian rank-restricted NLP

$$p^*_{\mathrm{SDPLR}} = \min_{Y \in \mathrm{St}(d,r)^n} \operatorname{tr}(\tilde{Q} Y^\mathsf{T} Y)$$

The constraints are equivalent to:

$$Y_i^\mathsf{T} Y_i = I_d, \quad Y_i \in \mathbb{R}^{r \times d}.$$

Notice: This is the definition of the *Stiefel manifold*:

$$\mathrm{St}(k, n) \triangleq \left\{ Y \in \mathbb{R}^{n \times k} \mid Y^\mathsf{T} Y = I_k \right\}.$$

Payoff: This is an *unconstrained* optimization problem.

[6] N. Boumal. “A Riemannian Low-Rank Method for Optimization Over Semidefinite Matrices with Block-Diagonal Constraints”. arXiv preprint: arXiv:1506.00575v2. 2015

Ensuring global optimality

Simplified pose-graph MLE

$$p^*_{\mathrm{MLE}} = \min_{R \in \mathrm{SO}(d)^n} \mathrm{tr}(\tilde{Q} R^\mathsf{T} R)$$

$$\Downarrow$$

Semidefinite relaxation

$$p^*_{\mathrm{SDP}} = \min_{Z \in \mathbb{S}^{dn}_+} \mathrm{tr}(\tilde{Q} Z)$$

$$\text{s.t.} \quad \mathrm{BlockDiag}_d(Z) = (I_d, \ldots, I_d)$$

$$\Downarrow$$

Riemannian rank-restricted NLP

$$p^*_{\mathrm{SDPLR}} = \min_{Y \in \mathrm{St}(d,r)^n} \mathrm{tr}(\tilde{Q} Y^\mathsf{T} Y)$$

Ensuring global optimality

Simplified pose-graph MLE

$$p^*_{\text{MLE}} = \min_{R \in \mathrm{SO}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$$\Downarrow$$

Semidefinite relaxation

$$p^*_{\text{SDP}} = \min_{Z \in \mathbb{S}^{dn}_+} \operatorname{tr}(\tilde{Q} Z)$$

$$\text{s.t.} \quad \operatorname{BlockDiag}_d(Z) = (I_d, \ldots, I_d)$$

$$\Downarrow$$

Riemannian rank-restricted NLP

$$p^*_{\text{SDPLR}} = \min_{Y \in \mathrm{St}(d,r)^n} \operatorname{tr}(\tilde{Q} Y^\mathsf{T} Y)$$

Question:
What have we actually gained?

Ensuring global optimality

Simplified pose-graph MLE

$$p^*_{\text{MLE}} = \min_{R \in \text{SO}(d)^n} \text{tr}(\tilde{Q} R^\mathsf{T} R)$$

$$\Downarrow$$

Semidefinite relaxation

$$p^*_{\text{SDP}} = \min_{Z \in \mathbb{S}^{dn}_+} \text{tr}(\tilde{Q} Z)$$

$$\text{s.t.} \quad \text{BlockDiag}_d(Z) = (I_d, \ldots, I_d)$$

$$\Downarrow$$

Riemannian rank-restricted NLP

$$p^*_{\text{SDPLR}} = \min_{Y \in \text{St}(d,r)^n} \text{tr}(\tilde{Q} Y^\mathsf{T} Y)$$

Proposition (Boumal et al. 2016)

If $Y \in \text{St}(d, r)^n$ is a rank deficient 2nd-order critical point for rank-restricted NLP, then:

- *Y is a global minimizer of Riemannian NLP*
- *$Z^* = Y^\mathsf{T} Y$ is a solution of the semidefinite relaxation.*

Ensuring global optimality

Simplified pose-graph MLE

$$p^*_{\text{MLE}} = \min_{R \in \text{SO}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$$\Downarrow$$

Semidefinite relaxation

$$p^*_{\text{SDP}} = \min_{Z \in \mathbb{S}_+^{dn}} \operatorname{tr}(\tilde{Q} Z)$$

$$\text{s.t.} \quad \text{BlockDiag}_d(Z) = (I_d, \dots, I_d)$$

$$\Downarrow$$

Riemannian rank-restricted NLP

$$p^*_{\text{SDPLR}} = \min_{Y \in \text{St}(d,r)^n} \operatorname{tr}(\tilde{Q} Y^\mathsf{T} Y)$$

Proposition (Boumal et al. 2016)

If $Y \in \text{St}(d, r)^n$ is a rank deficient 2nd-order critical point for rank-restricted NLP, then:

- *Y is a global minimizer of Riemannian NLP*
- *$Z^* = Y^\mathsf{T} Y$ is a solution of the semidefinite relaxation.*

$\Rightarrow$ We can use (fast) *local* search to find *globally optimal* solutions!

SE-Sync

Simplified pose-graph MLE

$$p^*_{\text{MLE}} = \min_{R \in \text{SO}(d)^n} \text{tr}(\tilde{Q} R^\mathsf{T} R)$$

$$\Downarrow$$

SDP relaxation

$$p^*_{\text{SDP}} = \min_{Z \in \mathbb{S}^{dn}_+} \text{tr}(\tilde{Q} Z)$$

$$\text{s.t.} \quad \text{Diag}_d(Z) = (I_d, \ldots, I_d)$$

$$\Downarrow$$

Riemannian rank-restricted NLP

$$p^*_{\text{SDPLR}} = \min_{Y \in \text{St}(d,r)^n} \text{tr}(\tilde{Q} Y^\mathsf{T} Y)$$

The SE-Sync algorithm:

1. Find low-rank factor Y^* using fast (2nd-order) NLP method in Riemannian Staircase.

SE-Sync

Simplified pose-graph MLE

$$p^*_{\text{MLE}} = \min_{R \in \text{SO}(d)^n} \text{tr}(\tilde{Q} R^\mathsf{T} R)$$

$$\Downarrow$$

SDP relaxation

$$p^*_{\text{SDP}} = \min_{Z \in \mathbb{S}^{dn}_+} \text{tr}(\tilde{Q} Z)$$

$$\text{s.t.} \quad \text{Diag}_d(Z) = (I_d, \ldots, I_d)$$

$$\Downarrow$$

Riemannian rank-restricted NLP

$$p^*_{\text{SDPLR}} = \min_{Y \in \text{St}(d,r)^n} \text{tr}(\tilde{Q} Y^\mathsf{T} Y)$$

The SE-Sync algorithm:

1. Find low-rank factor Y^* using fast (2nd-order) NLP method in Riemannian Staircase.
2. Compute SDP lower bound: $p^*_{\text{SDP}} = \text{tr}(\tilde{Q} {Y^*}^\mathsf{T} Y^*)$.

SE-Sync

Simplified pose-graph MLE

$$p^*_{\text{MLE}} = \min_{R \in \text{SO}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$\Downarrow$

SDP relaxation

$$p^*_{\text{SDP}} = \min_{Z \in \mathbb{S}^{dn}_+} \operatorname{tr}(\tilde{Q} Z)$$

$$\text{s.t.} \quad \operatorname{Diag}_d(Z) = (I_d, \ldots, I_d)$$

$\Downarrow$

Riemannian rank-restricted NLP

$$p^*_{\text{SDPLR}} = \min_{Y \in \text{St}(d,r)^n} \operatorname{tr}(\tilde{Q} Y^\mathsf{T} Y)$$

The SE-Sync algorithm:

1. Find low-rank factor Y^* using fast (2nd-order) NLP method in Riemannian Staircase.
2. Compute SDP lower bound: $p^*_{\text{SDP}} = \operatorname{tr}(\tilde{Q} Y^{*\mathsf{T}} Y^*)$.
3. Round $Y^* \to \hat{R} \in \text{SO}(d)^n$ using truncated SVD.

SE-Sync

Simplified pose-graph MLE

$$p^*_{\mathrm{MLE}} = \min_{R \in \mathrm{SO}(d)^n} \operatorname{tr}(\tilde{Q} R^\mathsf{T} R)$$

$$\Downarrow$$

SDP relaxation

$$p^*_{\mathrm{SDP}} = \min_{Z \in \mathbb{S}^{dn}_+} \operatorname{tr}(\tilde{Q} Z)$$

$$\text{s.t.} \quad \operatorname{Diag}_d(Z) = (I_d, \dots, I_d)$$

$$\Downarrow$$

Riemannian rank-restricted NLP

$$p^*_{\mathrm{SDPLR}} = \min_{Y \in \mathrm{St}(d,r)^n} \operatorname{tr}(\tilde{Q} Y^\mathsf{T} Y)$$

The SE-Sync algorithm:

1. Find low-rank factor Y^* using fast (2nd-order) NLP method in Riemannian Staircase.
2. Compute SDP lower bound: $p^*_{\mathrm{SDP}} = \operatorname{tr}(\tilde{Q} Y^{*\mathsf{T}} Y^*)$.
3. Round $Y^* \to \hat{R} \in \mathrm{SO}(d)^n$ using truncated SVD.
4. Return $\{\hat{R}, p^*_{\mathrm{SDP}}\}$.

SE-Sync

Simplified pose-graph MLE

$$p^*_{\text{MLE}} = \min_{R \in \text{SO}(d)^n} \text{tr}(\tilde{Q} R^\mathsf{T} R)$$

$$\Downarrow$$

SDP relaxation

$$p^*_{\text{SDP}} = \min_{Z \in \mathbb{S}^{dn}_+} \text{tr}(\tilde{Q} Z)$$

$$\text{s.t.} \quad \text{Diag}_d(Z) = (I_d, \ldots, I_d)$$

$$\Downarrow$$

Riemannian rank-restricted NLP

$$p^*_{\text{SDPLR}} = \min_{Y \in \text{St}(d,r)^n} \text{tr}(\tilde{Q} Y^\mathsf{T} Y)$$

The SE-Sync algorithm:

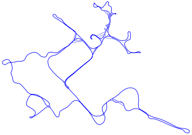
1. Find low-rank factor Y^* using fast (2nd-order) NLP method in Riemannian Staircase.
2. Compute SDP lower bound: $p^*_{\text{SDP}} = \text{tr}(\tilde{Q} {Y^*}^\mathsf{T} Y^*)$.
3. Round $Y^* \to \hat{R} \in \text{SO}(d)^n$ using truncated SVD.
4. Return $\{\hat{R}, p^*_{\text{SDP}}\}$.

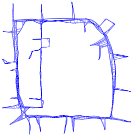
Payoff: If strong duality holds:

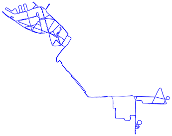
- $\hat{R}$ is *globally optimal*
- $\text{tr}(\tilde{Q} \hat{R}^\mathsf{T} \hat{R}) = p^*_{\text{SDP}}$ *certifies* it

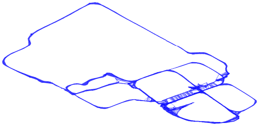
Experimental results: Large-scale SLAM benchmarks

	# Poses	PDL-GN (GTSAM)		SE-Sync		
		Objective value	Time [s]	Objective value	Time [s]	Rel. suboptimality
csail	1045	3.170×10^{1}	0.029	3.170×10^{1}	0.010	7.844×10^{-16}
intel	1728	5.235×10^{1}	0.120	5.235×10^{1}	0.071	1.357×10^{-16}
ais2klinik	15115	1.885×10^{2}	12.472	1.885×10^{2}	1.981	2.412×10^{-15}
garage	1661	1.263×10^{0}	0.415	1.263×10^{0}	0.468	1.618×10^{-14}
cubicle	5750	7.171×10^{2}	2.456	7.171×10^{2}	0.754	2.061×10^{-15}
rim	10195	5.461×10^{3}	6.803	5.461×10^{3}	2.256	5.663×10^{-15}


csail

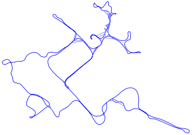

intel

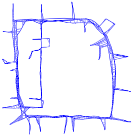

ais2klinik

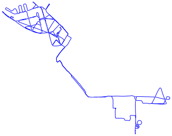

rim

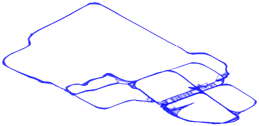
Experimental results: Large-scale SLAM benchmarks

	# Poses	PDL-GN (GTSAM)		SE-Sync		
		Objective value	Time [s]	Objective value	Time [s]	Rel. suboptimality
csail	1045	3.170×10^1	0.029	3.170×10^1	0.010	7.844×10^{-16}
intel	1728	5.235×10^1	0.120	5.235×10^1	0.071	1.357×10^{-16}
ais2klinik	15115	1.885×10^2	12.472	1.885×10^2	1.981	2.412×10^{-15}
garage	1661	1.263×10^0	0.415	1.263×10^0	0.468	1.618×10^{-14}
cubicle	5750	7.171×10^2	2.456	7.171×10^2	0.754	2.061×10^{-15}
rim	10195	5.461×10^3	6.803	5.461×10^3	2.256	5.663×10^{-15}


csail

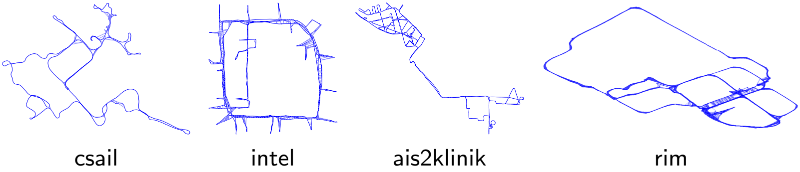

intel


ais2klinik


rim

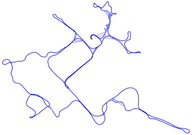
Experimental results: Large-scale SLAM benchmarks

	# Poses	PDL-GN (GTSAM) Objective value	PDL-GN (GTSAM) Time [s]	SE-Sync Objective value	SE-Sync Time [s]	SE-Sync Rel. suboptimality
csail	1045	3.170×10^{1}	0.029	3.170×10^{1}	0.010	7.844×10^{-16}
intel	1728	5.235×10^{1}	0.120	5.235×10^{1}	0.071	1.357×10^{-16}
ais2klinik	15115	1.885×10^{2}	12.472	1.885×10^{2}	1.981	2.412×10^{-15}
garage	1661	1.263×10^{0}	0.415	1.263×10^{0}	0.468	1.618×10^{-14}
cubicle	5750	7.171×10^{2}	2.456	7.171×10^{2}	0.754	2.061×10^{-15}
rim	10195	5.461×10^{3}	6.803	5.461×10^{3}	2.256	5.663×10^{-15}

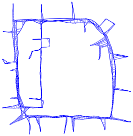


Experimental results: Large-scale SLAM benchmarks

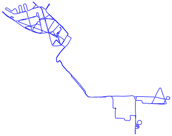
	# Poses	PDL-GN (GTSAM)		SE-Sync		
		Objective value	Time [s]	Objective value	Time [s]	Rel. suboptimality
csail	1045	3.170×10^{1}	0.029	3.170×10^{1}	0.010	7.844×10^{-16}
intel	1728	5.235×10^{1}	0.120	5.235×10^{1}	0.071	1.357×10^{-16}
ais2klinik	15115	1.885×10^{2}	12.472	1.885×10^{2}	1.981	2.412×10^{-15}
garage	1661	1.263×10^{0}	0.415	1.263×10^{0}	0.468	1.618×10^{-14}
cubicle	5750	7.171×10^{2}	2.456	7.171×10^{2}	0.754	2.061×10^{-15}
rim	10195	5.461×10^{3}	6.803	5.461×10^{3}	2.256	5.663×10^{-15}



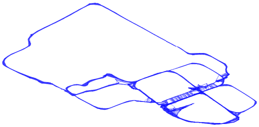
csail



intel

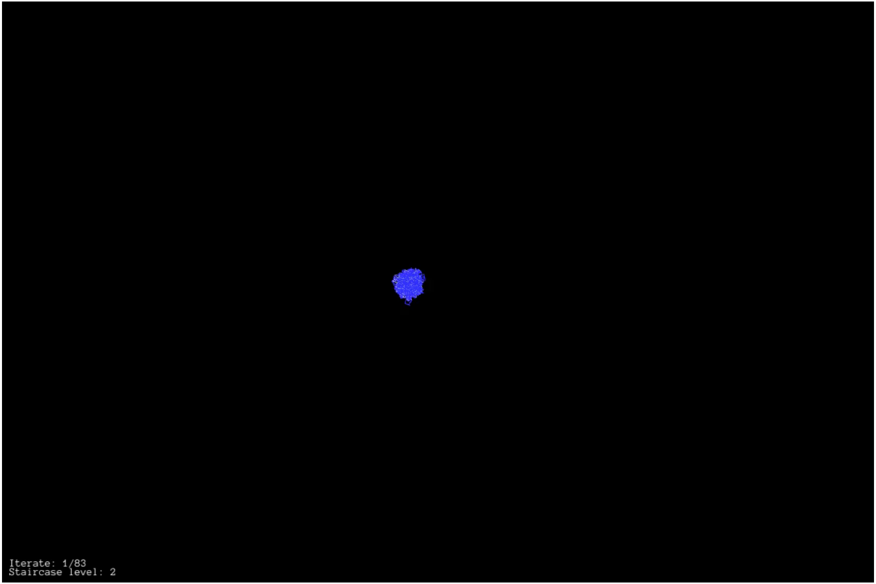


ais2klinik



rim

Visualization: Grid world



Certifiably Correct SLAM

Main Takeaway: *Convex relaxation* is a powerful tool!

This talk:

- Lagrangian duality & Lagrangian relaxation
- Pose-graph solution verification
- **SE-Sync:** first *practical*, *provably correct* SLAM algorithm[7]

[7] D.M. Rosen et al. "SE-Sync: A Certifiably Correct Algorithm for Synchronization over the Special Euclidean Group". In: *Intl. J. of Robotics Research* 38.2–3 (Mar. 2019), pp. 95–125.

[8] D.M. Rosen et al. "Advances in Inference and Representation for Simultaneous Localization and Mapping". In: *Annu. Rev. Control Robot. Auton. Syst.* (2021). (To appear).

Certifiably Correct SLAM

Main Takeaway: *Convex relaxation* is a powerful tool!

This talk:

- Lagrangian duality & Lagrangian relaxation
- Pose-graph solution verification
- **SE-Sync:** first *practical*, *provably correct* SLAM algorithm[7]

Current research frontier: *Certifiably correct perception*
$\Rightarrow$ 3D reconstruction, distributed SLAM + BA, sensor cal., etc...

[7] D.M. Rosen et al. "SE-Sync: A Certifiably Correct Algorithm for Synchronization over the Special Euclidean Group". In: *Intl. J. of Robotics Research* 38.2–3 (Mar. 2019), pp. 95–125.

[8] D.M. Rosen et al. "Advances in Inference and Representation for Simultaneous Localization and Mapping". In: *Annu. Rev. Control Robot. Auton. Syst.* (2021). (To appear).

Certifiably Correct SLAM

Main Takeaway: *Convex relaxation* is a powerful tool!

This talk:

- Lagrangian duality & Lagrangian relaxation
- Pose-graph solution verification
- **SE-Sync:** first *practical*, *provably correct* SLAM algorithm[7]

Current research frontier: *Certifiably correct perception*
$\Rightarrow$ 3D reconstruction, distributed SLAM + BA, sensor cal., etc...

More info: `https://david-m-rosen.github.io`

[7] D.M. Rosen et al. "SE-Sync: A Certifiably Correct Algorithm for Synchronization over the Special Euclidean Group". In: *Intl. J. of Robotics Research* 38.2–3 (Mar. 2019), pp. 95–125.

[8] D.M. Rosen et al. "Advances in Inference and Representation for Simultaneous Localization and Mapping". In: *Annu. Rev. Control Robot. Auton. Syst.* (2021). (To appear).

Certifiably Correct SLAM

Main Takeaway: *Convex relaxation* is a powerful tool!

This talk:

- Lagrangian duality & Lagrangian relaxation
- Pose-graph solution verification
- **SE-Sync:** first *practical*, *provably correct* SLAM algorithm[7]

Current research frontier: *Certifiably correct perception*
$\Rightarrow$ 3D reconstruction, distributed SLAM + BA, sensor cal., etc...

More info: `https://david-m-rosen.github.io`

New survey on certifiably correct perception (and other stuff!)[8]

---

[7] D.M. Rosen et al. "SE-Sync: A Certifiably Correct Algorithm for Synchronization over the Special Euclidean Group". In: *Intl. J. of Robotics Research* 38.2–3 (Mar. 2019), pp. 95–125.

[8] D.M. Rosen et al. "Advances in Inference and Representation for Simultaneous Localization and Mapping". In: *Annu. Rev. Control Robot. Auton. Syst.* (2021). (To appear).