Designing Sparse Reliable SLAM: A Graph-Theoretic Approach

Approximation Algorithms for Designing Sparse Graphs with the Maximum Weighted Number of Spanning Trees

---

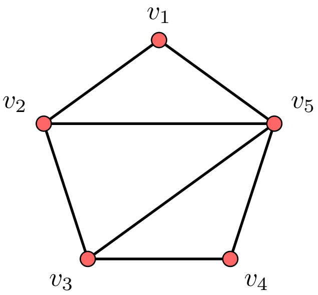
Kasra Khosoussi, Gaurav Sukhatme, Shoudong Huang and Gamini Dissanayake

University of Technology Sydney – University of Southern California

Spanning Trees

A subgraph that

1. includes all vertices
2. is a *tree*
   - connected
   - no cycles (i.e., minimally connected)

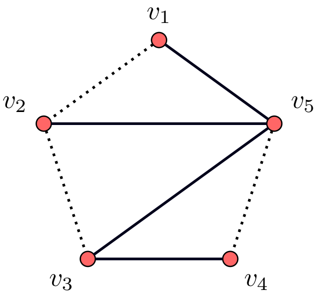


- # of spanning trees: $t(\mathcal{G}) \triangleq |\mathbb{T}(\mathcal{G})|$
- edge-weighted graphs:
  $t_w(\mathcal{G}) \triangleq \sum_{\mathcal{T} \in \mathbb{T}(\mathcal{G})} \prod_{e \in \mathcal{E}(\mathcal{T})} w(e)$
- **Matrix**-**Tree** Theorem:
  $t_w(\mathcal{G}) = \det \mathbf{L}_w(\mathcal{G}) \rightarrow$ reduced Laplacian

Spanning Trees

A subgraph that

1. includes all vertices
2. is a *tree*
   - connected
   - no cycles (i.e., minimally connected)

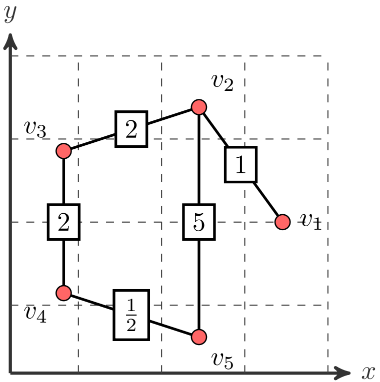


- # of spanning trees: $t(\mathcal{G}) \triangleq |\mathbb{T}(\mathcal{G})|$
- edge-weighted graphs:
  $t_w(\mathcal{G}) \triangleq \sum_{\mathcal{T}\in\mathbb{T}(\mathcal{G})} \prod_{e\in\mathcal{E}(\mathcal{T})} w(e)$
- **Matrix**-**Tree** Theorem:
  $t_w(\mathcal{G}) = \det \mathbf{L}_w(\mathcal{G}) \rightarrow$ reduced Laplacian

Graphical Representation of SLAM

- **Vertices**: n robot poses in d-dimensional space
- **Edges**: noisy pairwise measurements
- **Edge Weights**: measurement precision

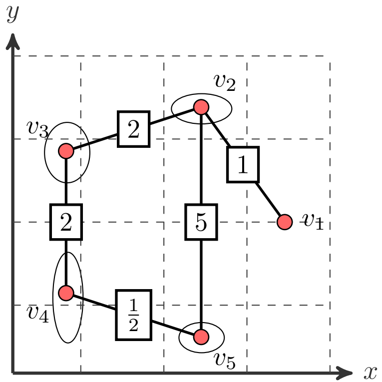
find the "optimal" embedding (drawing) in $\mathrm{SE}(d)^n$



Graphical Representation of SLAM

- **Vertices**: n robot poses in d-dimensional space
- **Edges**: noisy pairwise measurements
- **Edge Weights**: measurement precision

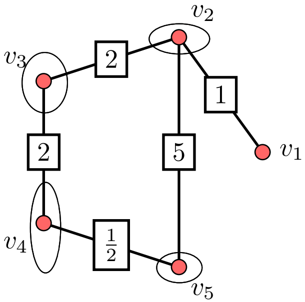
find the "optimal" embedding (drawing) in $\mathrm{SE}(d)^n$



Graphical Representation of SLAM

- **Vertices**: n robot poses in d-dimensional space
- **Edges**: noisy pairwise measurements
- **Edge Weights**: measurement precision

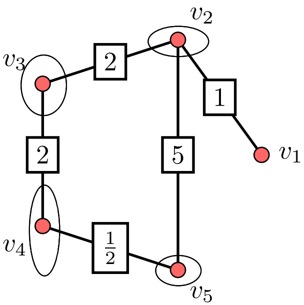
find the "optimal" embedding (drawing) in $\mathrm{SE}(d)^n$



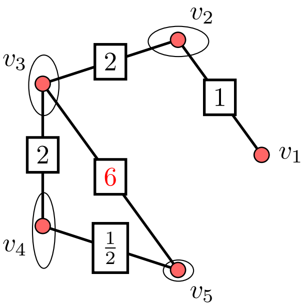
Graphical Representation of SLAM

- **Vertices**: n robot poses in d-dimensional space
- **Edges**: noisy pairwise measurements
- **Edge Weights**: measurement precision

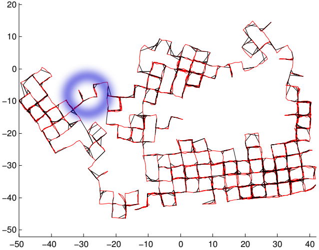
find the “optimal” embedding (drawing) in $\mathrm{SE}(d)^n$

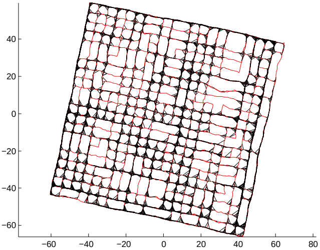


vs.



A Tale of Two Cities: Manhattan vs. City10K





SLAM: estimation over graph (ICRA 2016)

Key observation:

Fisher information ↔ graph Laplacian

volume of uncertainty ellipsoids ↔ weighted number of spanning trees

Theorems:

- Known orientation with dimension d (e.g., $d \in \{2{,}3\}$)

$$\underbrace{\det \mathsf{Cov}[\hat{\mathbf{x}}_{\mathsf{mle}}]}_{\text{hypervolume of uncertainty hyperellipsoid}} = \underbrace{t_w(\mathcal{G})^{-d}}_{\text{weighted number of spanning trees}}$$

- 2D pose-graph:

$$\underbrace{\lim_{\gamma \to 0} \det \mathsf{Cov}[\hat{\mathbf{x}}_{\mathsf{mle}}]}_{\text{hypervolume of uncertainty hyperellipsoid}} = \underbrace{t_{w_p}(\mathcal{G})^{-2} \cdot t_{w_\theta}(\mathcal{G})^{-1}}_{\text{weighted number of spanning trees}}$$

Conclusion:

$$\underbrace{\text{min.}\quad \det \mathsf{Cov}[\hat{\mathbf{x}}_{\mathsf{mle}}]}_{\text{Determinant(D)-optimality}} \Leftrightarrow \underbrace{\text{max.}\ \ t_w(\mathcal{G})}_{\text{tree}(t)\text{-optimality}}$$

Conclusion:

$$\underbrace{\text{min. } \det \mathsf{Cov}[\hat{\mathbf{x}}_{\mathsf{mle}}]}_{\text{Determinant(D)-optimality}} \Leftrightarrow \underbrace{\text{max. } t_w(\mathcal{G})}_{\text{tree}(t)\text{-optimality}}$$

subject to sparsity constraints

Conclusion:

$$\underbrace{\text{min. } \det \mathsf{Cov}[\hat{\mathbf{x}}_{\mathsf{mle}}]}_{\text{Determinant(D)-optimality}} \Leftrightarrow \underbrace{\text{max. } t_w(\mathcal{G})}_{\text{tree}(t)\text{-optimality}}$$

subject to sparsity constraints

This paper:

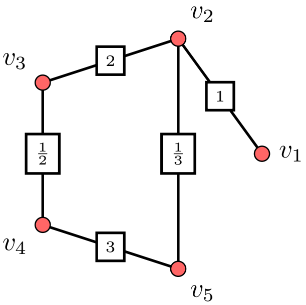
approximation algorithms for designing sparse t-optimal graphs

+

provable guarantees

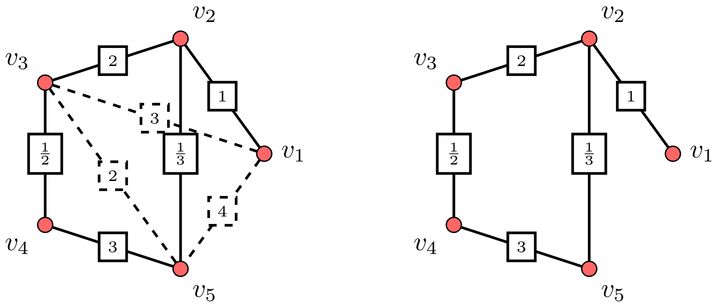
Edge Selection Problem (ESP)

- **P1**: Given a set of c new measurements (edges), pick $k \leq c$.
- **P2**: Given a c-subset of existing measurements (edges), prune $k \leq c$.



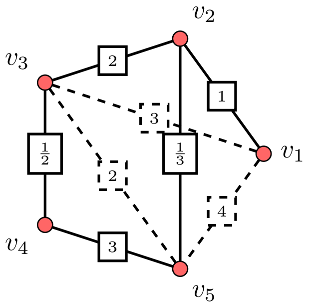
Edge Selection Problem (ESP)

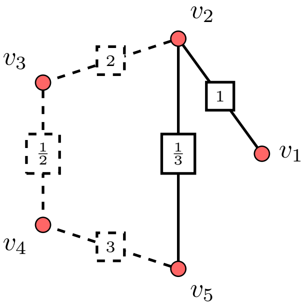
- **P1**: Given a set of c new measurements (edges), pick $k \leq c$.
- **P2**: Given a c-subset of existing measurements (edges), prune $k \leq c$.



Edge Selection Problem (ESP)

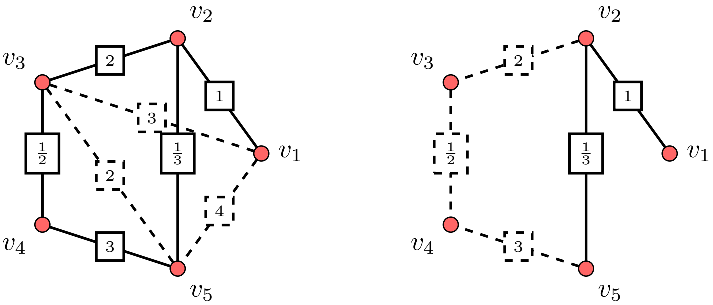
- **P1**: Given a set of c new measurements (edges), pick $k \leq c$.
- **P2**: Given a c-subset of existing measurements (edges), prune $k \leq c$.





Edge Selection Problem (ESP)

- **P1**: Given a set of c new measurements (edges), pick $k \leq c$.
- **P2**: Given a c-subset of existing measurements (edges), prune $k \leq c$.



equivalent classes of problems

k-ESP

1. $\mathcal{G}_{\text{init}} = (\mathcal{V}, \mathcal{E}_{\text{init}}, w)$ — connected
2. $\mathcal{C}$ candidate edges

$$\underset{\mathcal{E} \subseteq \mathcal{C}}{\text{maximize}} \quad \underbrace{\mathcal{T}\text{ree}\left(\mathcal{E}_{\text{init}} \cup \mathcal{E}\right)}_{\text{D-optimality}} \quad \text{s.t.} \quad \underbrace{|\mathcal{E}| = k}_{\text{sparsity}} \tag{k-ESP}$$

k-ESP

1. $\mathcal{G}_{\text{init}} = (\mathcal{V}, \mathcal{E}_{\text{init}}, w)$ — connected
2. $\mathcal{C}$ candidate edges

$$\underset{\mathcal{E} \subseteq \mathcal{C}}{\text{maximize}} \quad \underbrace{\mathcal{T}\text{ree}\left(\mathcal{E}_{\text{init}} \cup \mathcal{E}\right)}_{\text{D-optimality}} \quad \text{s.t.} \quad \underbrace{|\mathcal{E}| = k}_{\text{sparsity}} \qquad (k\text{-ESP})$$

☺ Has been studied in other fields (e.g., network reliability)

k-ESP

1. $\mathcal{G}_{\text{init}} = (\mathcal{V}, \mathcal{E}_{\text{init}}, w)$ — connected
2. $\mathcal{C}$ candidate edges

$$\underset{\mathcal{E} \subseteq \mathcal{C}}{\text{maximize}} \quad \underbrace{\mathcal{T}\text{ree}\left(\mathcal{E}_{\text{init}} \cup \mathcal{E}\right)}_{\text{D-optimality}} \quad \text{s.t.} \quad \underbrace{|\mathcal{E}| = k}_{\text{sparsity}} \tag{k-ESP}$$

- 🙂 Has been studied in other fields (e.g., network reliability)
- 😐 Characterizing t-optimal graphs $\rightarrow$ open problem

k-ESP

1. $\mathcal{G}_{\text{init}} = (\mathcal{V}, \mathcal{E}_{\text{init}}, w)$ — connected
2. $\mathcal{C}$ candidate edges

$$\underset{\mathcal{E} \subseteq \mathcal{C}}{\text{maximize}} \quad \underbrace{\mathcal{T}\text{ree}\left(\mathcal{E}_{\text{init}} \cup \mathcal{E}\right)}_{\text{D-optimality}} \quad \text{s.t.} \quad \underbrace{|\mathcal{E}| = k}_{\text{sparsity}} \qquad (k\text{-ESP})$$

- 🙂 Has been studied in other fields (e.g., network reliability)
- 😐 Characterizing t-optimal graphs $\rightarrow$ open problem
- 🙁 Exhaustive search is not tractable

k-ESP

1. $\mathcal{G}_{\text{init}} = (\mathcal{V}, \mathcal{E}_{\text{init}}, w)$ — connected
2. $\mathcal{C}$ candidate edges

$$\underset{\mathcal{E} \subseteq \mathcal{C}}{\text{maximize}} \quad \underbrace{\mathcal{T}\text{ree}\left(\mathcal{E}_{\text{init}} \cup \mathcal{E}\right)}_{\text{D-optimality}} \quad \text{s.t.} \quad \underbrace{|\mathcal{E}| = k}_{\text{sparsity}} \qquad (k\text{-ESP})$$

- Has been studied in other fields (e.g., network reliability)
- Characterizing t-optimal graphs $\to$ open problem
- Exhaustive search is not tractable

$k = 1$ (IROS'14):

- Optimal candidate: maximum effective resistance R_{eff}
  - R_{eff}: A metric to define a "distance" between two vertices in $\mathcal{G}$

k-ESP

1. $\mathcal{G}_{\text{init}} = (\mathcal{V}, \mathcal{E}_{\text{init}}, w)$ — connected
2. $\mathcal{C}$ candidate edges

$$\underset{\mathcal{E} \subseteq \mathcal{C}}{\text{maximize}} \quad \underbrace{\mathcal{T}\text{ree}\left(\mathcal{E}_{\text{init}} \cup \mathcal{E}\right)}_{\text{D-optimality}} \quad \text{s.t.} \quad \underbrace{|\mathcal{E}| = k}_{\text{sparsity}} \qquad (k\text{-ESP})$$

- ☺ Has been studied in other fields (e.g., network reliability)
- 😐 Characterizing t-optimal graphs $\rightarrow$ open problem
- ☹ Exhaustive search is not tractable

$k = 1$ (IROS'14):

- Optimal candidate: maximum effective resistance R_{eff}
  - R_{eff}: A metric to define a "distance" between two vertices in $\mathcal{G}$
  - Optimal policy: connect the vertices that are furthest from each other

Approximation Alg. 1: Greedy

For k times:

- Compute the effective resistance of the remaining candidates
- Pick the candidate with the highest effective resistance

Approximation Alg. 1: Greedy

For k times:

- Compute the effective resistance of the remaining candidates
- Pick the candidate with the highest effective resistance

$\_\ \_\ \_\ \_\ \_\ \_\ \_\ \_\ \_\ \_$

Approximation Alg. 1: Greedy

For k times:

- Compute the effective resistance of the remaining candidates
- Pick the candidate with the highest effective resistance

— — — — — — — — — —

1 Monotone: iff $f(\mathcal{A} \cup \{x\}) \geq f(\mathcal{A})$

Approximation Alg. 1: Greedy

For k times:

- Compute the effective resistance of the remaining candidates
- Pick the candidate with the highest effective resistance

— — — — — — — — — —

1. Monotone: iff $f(\mathcal{A} \cup \{x\}) \geq f(\mathcal{A})$
2. Submodular: iff f exhibits the *diminishing returns* property, e.g.,

Approximation Alg. 1: Greedy

For k times:

- Compute the effective resistance of the remaining candidates
- Pick the candidate with the highest effective resistance

1. Monotone: iff $f(\mathcal{A} \cup \{x\}) \geq f(\mathcal{A})$
2. Submodular: iff f exhibits the *diminishing returns* property, e.g.,

   wining an  :  vs. 

Approximation Alg. 1: Greedy

For k times:

- Compute the effective resistance of the remaining candidates
- Pick the candidate with the highest effective resistance

1. Monotone: iff $f(\mathcal{A} \cup \{x\}) \geq f(\mathcal{A})$
2. Submodular: iff f exhibits the *diminishing returns* property, e.g.,

   wining an  :  vs. 

Approximation Alg. 1: Greedy

For k times:

- Compute the effective resistance of the remaining candidates
- Pick the candidate with the highest effective resistance

1. Monotone: iff $f(\mathcal{A} \cup \{x\}) \geq f(\mathcal{A})$
2. Submodular: iff f exhibits the *diminishing returns* property, e.g.,

   wining an  :  vs. 

Greedy algorithm → constant-factor approximation with a factor of $(1 - 1/e) \approx 0.63$

Approximation Alg. 1: Greedy

For k times:

- Compute the effective resistance of the remaining candidates
- Pick the candidate with the highest effective resistance

— — — — — — — — — —

1. Monotone: iff $f(\mathcal{A} \cup \{x\}) \geq f(\mathcal{A})$
2. Submodular: iff f exhibits the *diminishing returns* property, e.g.,

wining an  :  vs. 

Greedy algorithm $\rightarrow$ constant-factor approximation with a factor of $(1 - 1/e) \approx 0.63$

— — — — — — — — — —

- **Main Result**: $\mathcal{E} \mapsto \mathcal{T}\text{ree}\,(\mathcal{E}_{\text{init}} \cup \mathcal{E})$ is monotone log-submodular

Approximation Alg. 1: Greedy

For k times:

- Compute the effective resistance of the remaining candidates
- Pick the candidate with the highest effective resistance

1. Monotone: iff $f(\mathcal{A} \cup \{x\}) \geq f(\mathcal{A})$
2. Submodular: iff f exhibits the *diminishing returns* property, e.g.,

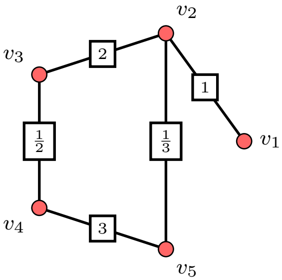
   wining an  :  vs. 

Greedy algorithm $\rightarrow$ constant-factor approximation with a factor of $(1 - 1/e) \approx 0.63$

- **Main Result**: $\mathcal{E} \mapsto \mathcal{T}\text{ree}\,(\mathcal{E}_{\text{init}} \cup \mathcal{E})$ is monotone log-submodular
- Value of greedy is within a constant-factor of OPT (see the paper)

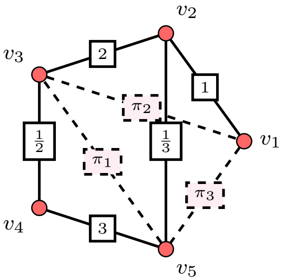
Approximation Alg. 2: Convex relaxation

- Selectors: $\boldsymbol{\pi} \triangleq [\pi_1 \ \dots \ \pi_c]^\top \in \{0{,}1\}^c$
- Pick the ith candidate edge iff $\pi_i = 1$



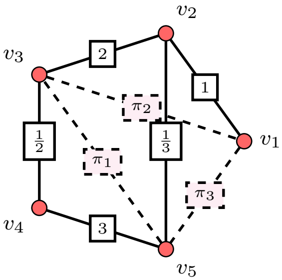
Approximation Alg. 2: Convex relaxation

- Selectors: $\boldsymbol{\pi} \triangleq [\pi_1 \ \ldots \ \pi_c]^\top \in \{0,1\}^c$
- Pick the ith candidate edge iff $\pi_i = 1$



Approximation Alg. 2: Convex relaxation

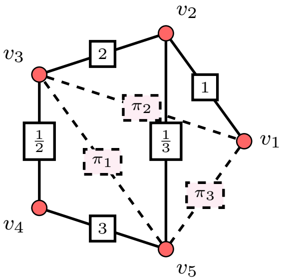
- Selectors: $\boldsymbol{\pi} \triangleq [\pi_1 \ \ldots \ \pi_c]^\top \in \{0,1\}^c$
- Pick the ith candidate edge iff $\pi_i = 1$



$$\begin{aligned} \underset{\boldsymbol{\pi}}{\text{maximize}} \quad & \log \det \mathbf{L}(\boldsymbol{\pi}) \\ \text{subject to} \quad & \sum_{i=1}^{c} \pi_i = k, \\ & \pi_i \in \{0,1\}, \ \forall i \in [c]. \end{aligned}$$

Approximation Alg. 2: Convex relaxation

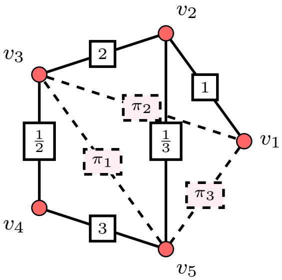
- Selectors: $\boldsymbol{\pi} \triangleq [\pi_1 \ \ldots \ \pi_c]^\top \in \{0,1\}^c$
- Pick the ith candidate edge iff $\pi_i = 1$



$$\begin{aligned} \underset{\boldsymbol{\pi}}{\text{maximize}} \quad & \log \det \mathbf{L}(\boldsymbol{\pi}) \\ \text{subject to} \quad & \sum_{i=1}^{c} \pi_i = k, \\ & \pi_i \in [0,1], \ \forall i \in [c]. \end{aligned}$$

Approximation Alg. 2: Convex relaxation

- Selectors: $\boldsymbol{\pi} \triangleq [\pi_1 \; \dots \; \pi_c]^\top \in \{0,1\}^c$
- Pick the ith candidate edge iff $\pi_i = 1$

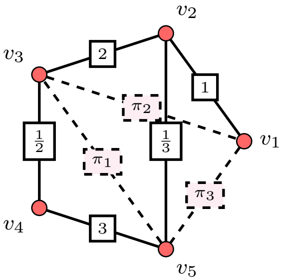


$$
\begin{aligned}
\underset{\boldsymbol{\pi}}{\text{maximize}} \quad & \log \det \mathbf{L}(\boldsymbol{\pi}) \\
\text{subject to} \quad & \sum_{i=1}^{c} \pi_i = k, \\
& \pi_i \in [0,1], \; \forall i \in [c].
\end{aligned}
$$

- Convex Program (MAXDET) $\rightarrow$ `CVX/YALMIP` + `SDPT3`

Approximation Alg. 2: Convex relaxation

- Selectors: $\boldsymbol{\pi} \triangleq [\pi_1 \ \dots \ \pi_c]^\top \in \{0,1\}^c$
- Pick the ith candidate edge iff $\pi_i = 1$



$$\underset{\boldsymbol{\pi}}{\text{maximize}} \quad \log {\color{green}\det \mathbf{L}(\boldsymbol{\pi})}$$

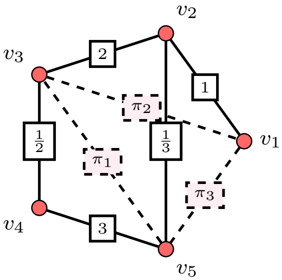
$$\text{subject to} \quad \sum_{i=1}^{c} \pi_i = k,$$

$$\pi_i \in {\color{red}[}0,1{\color{red}]}, \ \forall i \in [c].$$

- Convex Program (MAXDET) $\rightarrow$ `CVX/YALMIP` + `SDPT3`
- $\text{OPT}_{\text{cvx}} \geq \text{OPT}$ where $\text{OPT}_{\text{cvx}} \triangleq \log\det \mathbf{L}(\boldsymbol{\pi}^\star)$

Approximation Alg. 2: Convex relaxation

- Selectors: $\boldsymbol{\pi} \triangleq [\pi_1 \; \dots \; \pi_c]^\top \in \{0,1\}^c$
- Pick the ith candidate edge iff $\pi_i = 1$

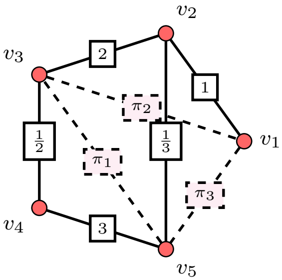


$$\begin{aligned} \underset{\boldsymbol{\pi}}{\text{maximize}} \quad & \log \det \mathbf{L}(\boldsymbol{\pi}) \\ \text{subject to} \quad & \sum_{i=1}^{c} \pi_i = k, \\ & \pi_i \in [0,1], \; \forall i \in [c]. \end{aligned}$$

- Convex Program (MAXDET) $\rightarrow$ `CVX`/`YALMIP` + `SDPT3`
- $\text{OPT}_{\text{cvx}} \geq \text{OPT}$ where $\text{OPT}_{\text{cvx}} \triangleq \log \det \mathbf{L}(\boldsymbol{\pi}^\star)$
- if $\boldsymbol{\pi}^\star \in \{0,1\}^c \Rightarrow$ optimal solution

Approximation Alg. 2: Convex relaxation

- Selectors: $\boldsymbol{\pi} \triangleq [\pi_1 \, \dots \, \pi_c]^\top \in \{0,1\}^c$
- Pick the ith candidate edge iff $\pi_i = 1$

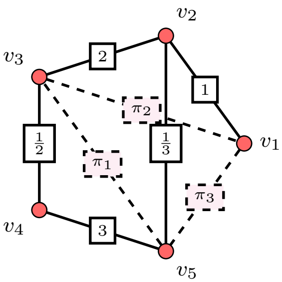


$$
\begin{aligned}
\underset{\boldsymbol{\pi}}{\text{maximize}} \quad & \log \det \mathbf{L}(\boldsymbol{\pi}) \\
\text{subject to} \quad & \sum_{i=1}^{c} \pi_i = k, \\
& \pi_i \in [0,1], \ \forall i \in [c].
\end{aligned}
$$

- Convex Program (MAXDET) $\rightarrow$ `CVX`/`YALMIP` + `SDPT3`
- $\text{OPT}_{\text{cvx}} \geq \text{OPT}$ where $\text{OPT}_{\text{cvx}} \triangleq \log \det \mathbf{L}(\boldsymbol{\pi}^\star)$
- if $\boldsymbol{\pi}^\star \in \{0,1\}^c \Rightarrow$ optimal solution
- otherwise need a **rounding** scheme: $[0,1]^c \ni \boldsymbol{\pi}^\star \mapsto \tilde{\boldsymbol{\pi}} \in \{0,1\}^c$ s.t. $\|\tilde{\boldsymbol{\pi}}\|_0 = k$

Approximation Alg. 2: Convex relaxation

- Selectors: $\boldsymbol{\pi} \triangleq [\pi_1 \ \dots \ \pi_c]^\top \in \{0,1\}^c$
- Pick the ith candidate edge iff $\pi_i = 1$



$$\begin{aligned} \underset{\boldsymbol{\pi}}{\text{maximize}} \quad & \log \det \mathbf{L}(\boldsymbol{\pi}) \\ \text{subject to} \quad & \sum_{i=1}^{c} \pi_i = k, \\ & \pi_i \in [0,1], \ \forall i \in [c]. \end{aligned}$$

- Convex Program (MAXDET) $\rightarrow$ `CVX`/`YALMIP` + `SDPT3`
- $\text{OPT}_{\text{cvx}} \geq \text{OPT}$ where $\text{OPT}_{\text{cvx}} \triangleq \log\det \mathbf{L}(\boldsymbol{\pi}^\star)$
  - if $\boldsymbol{\pi}^\star \in \{0,1\}^c \Rightarrow$ optimal solution
  - otherwise need a **rounding** scheme: $[0,1]^c \ni \boldsymbol{\pi}^\star \mapsto \tilde{\boldsymbol{\pi}} \in \{0,1\}^c$ s.t. $\|\tilde{\boldsymbol{\pi}}\|_0 = k$
  - e.g., pick the k candidate edges that correspond to the k largest $\pi_i^\star$'s

Convex relaxation: a new narrative

- $0 \leq \pi_i \leq 1$: the probability of sampling the ith candidate

Convex relaxation: a new narrative

- $0 \leq \pi_i \leq 1$: the probability of sampling the ith candidate

$$\begin{aligned} \underset{\boldsymbol{\pi}}{\text{maximize}} \quad & \mathbb{E}_{\boldsymbol{\pi}}\left[t_w(\mathcal{G})\right] \\ \text{subject to} \quad & \mathbb{E}_{\boldsymbol{\pi}}\left[|\mathcal{E}|\right] = k, \\ & \pi_i \in [0,1],\ \forall i \in [c]. \end{aligned}$$

Convex relaxation: a new narrative

- $0 \le \pi_i \le 1$: the probability of sampling the ith candidate

$$\begin{aligned} \underset{\boldsymbol{\pi}}{\text{maximize}} \quad & \mathbb{E}_{\boldsymbol{\pi}}\left[t_w(\mathcal{G})\right] \\ \text{subject to} \quad & \mathbb{E}_{\boldsymbol{\pi}}\left[|\mathcal{E}|\right] = k, \\ & \pi_i \in [0,1], \ \forall i \in [c]. \end{aligned}$$

$$\begin{aligned} \underset{\boldsymbol{\pi}}{\text{maximize}} \quad & \log \det \mathbf{L}(\boldsymbol{\pi}) \\ \text{subject to} \quad & \sum_{i=1}^{c} \pi_i = k, \\ & \pi_i \in [0,1], \ \forall i \in [c]. \end{aligned}$$

Convex relaxation: a new narrative

- $0 \le \pi_i \le 1$: the probability of sampling the ith candidate

$$\begin{aligned} \underset{\boldsymbol{\pi}}{\text{maximize}} \quad & \mathbb{E}_{\boldsymbol{\pi}}\left[t_w(\mathcal{G})\right] \\ \text{subject to} \quad & \mathbb{E}_{\boldsymbol{\pi}}\left[|\mathcal{E}|\right] = k, \\ & \pi_i \in [0,1],\ \forall i \in [c]. \end{aligned}$$

$$\begin{aligned} \underset{\boldsymbol{\pi}}{\text{maximize}} \quad & \log \det \mathbf{L}(\boldsymbol{\pi}) \\ \text{subject to} \quad & \sum_{i=1}^{c} \pi_i = k, \\ & \pi_i \in [0,1],\ \forall i \in [c]. \end{aligned}$$

  - relaxation: hard $\rightarrow$ soft objective and constraints

Convex relaxation: a new narrative

- $0 \leq \pi_i \leq 1$: the probability of sampling the ith candidate

$$\begin{aligned} \underset{\boldsymbol{\pi}}{\text{maximize}} \quad & \mathbb{E}_{\boldsymbol{\pi}}\left[t_w(\mathcal{G})\right] \\ \text{subject to} \quad & \mathbb{E}_{\boldsymbol{\pi}}\left[|\mathcal{E}|\right] = k, \\ & \pi_i \in [0,1],\ \forall i \in [c]. \end{aligned}$$

$$\begin{aligned} \underset{\boldsymbol{\pi}}{\text{maximize}} \quad & \log \det \mathbf{L}(\boldsymbol{\pi}) \\ \text{subject to} \quad & \sum_{i=1}^{c} \pi_i = k, \\ & \pi_i \in [0,1],\ \forall i \in [c]. \end{aligned}$$

- relaxation: hard → soft objective and constraints
- justification for deterministic rounding (picking the k largest $\pi_i^\star$)

Convex relaxation: a new narrative

- $0 \leq \pi_i \leq 1$: the probability of sampling the ith candidate

$$\begin{aligned} \underset{\boldsymbol{\pi}}{\text{maximize}} \quad & \mathbb{E}_{\boldsymbol{\pi}}\left[t_w(\mathcal{G})\right] \\ \text{subject to} \quad & \mathbb{E}_{\boldsymbol{\pi}}\left[|\mathcal{E}|\right] = k, \\ & \pi_i \in [0,1],\ \forall i \in [c]. \end{aligned}$$

$$\begin{aligned} \underset{\boldsymbol{\pi}}{\text{maximize}} \quad & \log\det \mathbf{L}(\boldsymbol{\pi}) \\ \text{subject to} \quad & \sum_{i=1}^{c} \pi_i = k, \\ & \pi_i \in [0,1],\ \forall i \in [c]. \end{aligned}$$

- relaxation: hard $\rightarrow$ soft objective and constraints
- justification for deterministic rounding (picking the k largest $\pi_i^\star$)
- randomized rounding schemes

Certifying near-optimality

(notation: $\tau \triangleq \log \mathcal{T}\text{ree}$)

Corollary:

$$\max\left\{\tau_{\text{greedy}}, \tau_{\text{cvx}}\right\} \leq \underbrace{\text{OPT}}_{\text{intractable}} \leq \min\left\{\alpha\tau_{\text{greedy}} + (1-\alpha)\tau_{\text{init}}, \tau^{\star}_{\text{cvx}}\right\}$$

where $\alpha \triangleq (1 - 1/e)^{-1} \approx 1.58$

Certifying near-optimality

(notation: $\tau \triangleq \log \mathcal{T}\text{ree}$)

Corollary:

$$\underbrace{\max \left\{\tau_{\text{greedy}}, \tau_{\text{cvx}}\right\}}_{\mathcal{L}} \leq \text{OPT} \leq \underbrace{\min \left\{\alpha\tau_{\text{greedy}} + (1-\alpha)\tau_{\text{init}}, \tau^{\star}_{\text{cvx}}\right\}}_{\mathcal{U}}$$

where $\alpha \triangleq (1 - 1/e)^{-1} \approx 1.58$

Certifying near-optimality

(notation: $\tau \triangleq \log \mathcal{T}\mathrm{ree}$)

Corollary:

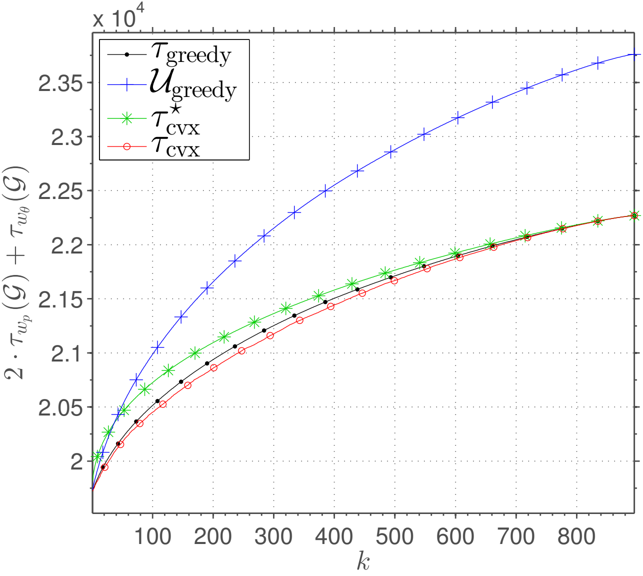
$$\underbrace{\max\left\{\tau_{\mathsf{greedy}}, \tau_{\mathsf{cvx}}\right\}}_{\mathcal{L}} \leq \mathsf{OPT} \leq \underbrace{\min\left\{\alpha\tau_{\mathsf{greedy}} + (1-\alpha)\tau_{\mathsf{init}}, \tau^{\star}_{\mathsf{cvx}}\right\}}_{\mathcal{U}}$$

where $\alpha \triangleq (1 - 1/e)^{-1} \approx 1.58$

- given any (suboptimal) design $\mathcal{A}$,

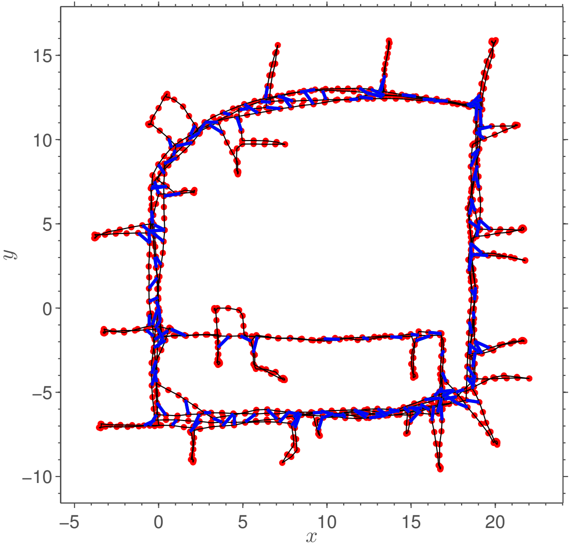
$$\max\left\{0, \mathcal{L} - \tau_{\mathcal{A}}\right\} \leq \underbrace{\mathsf{OPT} - \tau_{\mathcal{A}}}_{\mathsf{gap}} \leq \mathcal{U} - \tau_{\mathcal{A}}$$

Intel Research Lab Dataset



Intel Research Lab Dataset

$k = 161 \quad \tau_{\text{greedy}} = 2.079e+04 \quad \mathcal{U} = 2.097e+04$



Greedy design for $k = 161$ loop-closure edges (18% of candidates)

Other Applications

- **Relevant applications**:
  - Network reliability under random edge failure (e.g., power or communication networks)
  - D-optimal incomplete block designs
  - Molecular physics
  - RNA modelling
  - Estimation over sensor networks (e.g., time synchronization)
  - Connectivity controller for multi-robot systems

Dirty Laundry & Conclusion

Active SLAM: (work in progress — not addressed here)

- Dimensionality reduction for D-optimal planning:

  sensitive to topology and not to a particular embedding
- Hierarchical planning: topology $\rightarrow$ embedding
- Advantages:
  - Agnostic to sensor readings and not confined to a particular embedding
  - Robust against local minima and linearization errors
  - A compact and almost lossless representation

Contributions:

- A new submodular graph invariant: $\log \mathcal{T}\text{ree}$
- First near-optimal approximation algorithms for designing t-optimal graphs
- A new narrative for MAXDET-like convex relaxation
- Near-optimality certificates

Thank you!

1. Maximizing the Weighted Number of Spanning Trees (arXiv)
2. Tree-Connectivity: Evaluating the Graphical Structure of SLAM (ICRA'16)
3. Good, Bad and Ugly Graphs for SLAM (RSS'15 Workshop)
4. Novel Insights Into the Impact of Graph Structure on SLAM (IROS'14)

kasra.github.io

δ-**ESP***

$$\begin{aligned} \underset{\mathcal{E} \subseteq \mathcal{C}}{\text{minimize}} \quad & |\mathcal{E}| \\ \text{subject to} \quad & \text{tree}_w(\mathcal{E}_{\text{init}} \cup \mathcal{E}) \geq \delta. \end{aligned}$$

δ-**ESP***

$$\underset{\mathcal{E} \subseteq \mathcal{C}}{\text{minimize}} \quad |\mathcal{E}|$$

$$\text{subject to} \quad \text{tree}_w(\mathcal{E}_{\text{init}} \cup \mathcal{E}) \geq \delta.$$

- Performance guarantees

$$k_{\text{greedy}} \leq \zeta^* k_{\mathrm{OPT}}$$

δ-ESP*

$$\begin{aligned} \underset{\mathcal{E} \subseteq \mathcal{C}}{\text{minimize}} \quad & |\mathcal{E}| \\ \text{subject to} \quad & \text{tree}_w(\mathcal{E}_{\text{init}} \cup \mathcal{E}) \geq \delta. \end{aligned}$$

- Performance guarantees

$$k_{\text{greedy}} \leq \zeta^* k_{\mathrm{OPT}}$$

- Near-optimality certificates

$$\max \left\{ \left\lceil \frac{1}{\zeta^*} k_{\text{greedy}} \right\rceil, \left\lceil \sum_{i=1}^{c} \pi_i^\star \right\rceil \right\} \leq k_{\text{OPT}} \leq \min \left\{ k_{\text{greedy}}, k_{\text{cvx}} \right\}$$

δ-ESP*

$$\underset{\mathcal{E}\subseteq\mathcal{C}}{\text{minimize}} \quad |\mathcal{E}|$$
$$\text{subject to} \quad \mathsf{tree}_w(\mathcal{E}_{\mathsf{init}} \cup \mathcal{E}) \geq \delta.$$

- Performance guarantees

$$k_{\mathsf{greedy}} \leq \zeta^* k_{\mathrm{OPT}}$$

- Near-optimality certificates

$$\underbrace{\max\left\{\left\lceil \frac{1}{\zeta^*} k_{\mathsf{greedy}} \right\rceil, \left\lceil \sum_{i=1}^{c} \pi_i^\star \right\rceil\right\}}_{\mathcal{L}} \leq k_{\mathsf{OPT}} \leq \underbrace{\min\left\{k_{\mathsf{greedy}}, k_{\mathsf{cvx}}\right\}}_{\mathcal{U}}.$$

δ-ESP*

$$\underset{\mathcal{E} \subseteq \mathcal{C}}{\text{minimize}} \quad |\mathcal{E}|$$

$$\text{subject to} \quad \text{tree}_w(\mathcal{E}_{\text{init}} \cup \mathcal{E}) \geq \delta.$$

- Performance guarantees

$$k_{\text{greedy}} \leq \zeta^* k_{\text{OPT}}$$

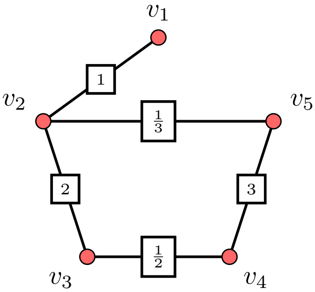
- Near-optimality certificates

$$\underbrace{\max \left\{ \left\lceil \frac{1}{\zeta^*} k_{\text{greedy}} \right\rceil, \left\lceil \sum_{i=1}^{c} \pi_i^\star \right\rceil \right\}}_{\mathcal{L}} \leq k_{\text{OPT}} \leq \underbrace{\min \left\{ k_{\text{greedy}}, k_{\text{cvx}} \right\}}_{\mathcal{U}}.$$

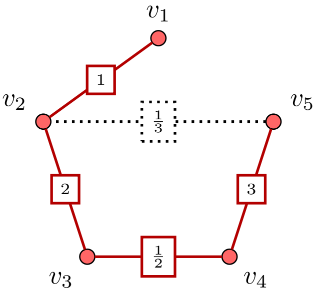
- Given any design with $k_{\mathcal{A}}$ edges,

$$\min \left\{ 0, k_{\mathcal{A}} - \mathcal{U} \right\} \leq \underbrace{k_{\mathcal{A}} - k_{\text{OPT}}}_{\text{gap}} \leq k_{\mathcal{A}} - \mathcal{L}$$

Kirchhoff's Matrix-Tree Theorems

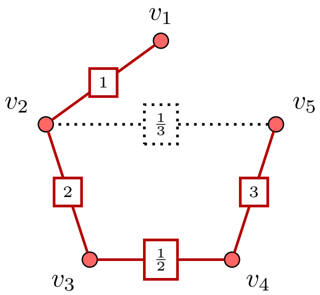


Kirchhoff's Matrix-Tree Theorems



$$\mathbb{V}_w(\mathcal{T}) = 1 \times 2 \times \frac{1}{2} \times 3 = 3$$

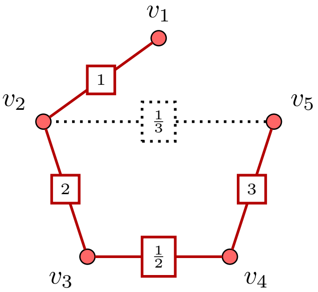
Kirchhoff's Matrix-Tree Theorems



$$\mathbb{V}_w(\mathcal{T}) = 1 \times 2 \times \frac{1}{2} \times 3 = 3$$

$$t_w(\mathcal{G}) \triangleq \sum_{\mathcal{T} \in \mathbb{T}(\mathcal{G})} \mathbb{V}_w(\mathcal{T})$$

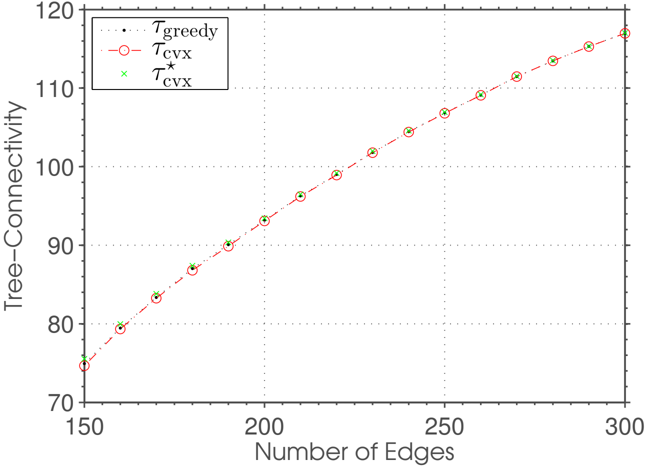
Kirchhoff's Matrix-Tree Theorems



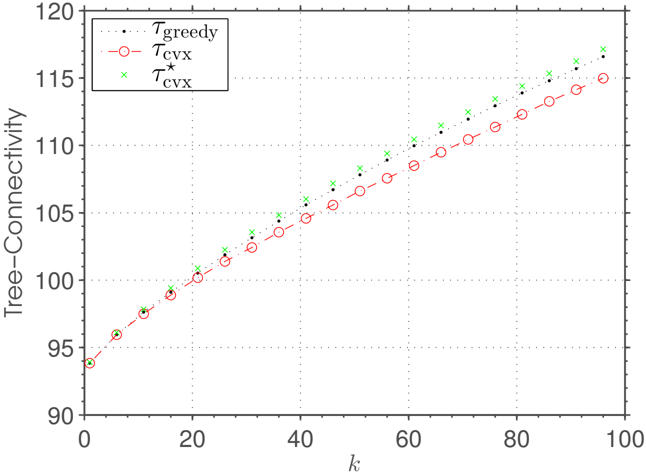
$$\mathbb{V}_w(\mathcal{T}) = 1 \times 2 \times \frac{1}{2} \times 3 = 3$$

$$t_w(\mathcal{G}) \triangleq \sum_{\mathcal{T} \in \mathbb{T}(\mathcal{G})} \mathbb{V}_w(\mathcal{T}) \overset{(\mathsf{MT})}{=} \det \mathbf{L}_w$$

$\mathbf{L}_w$: reduced weighted graph Laplacian



Varying $|\mathcal{E}_{\text{init}}|$ for $k = 5$ and $|\mathcal{V}| = 50$



Varying k for $|\mathcal{V}| = 50$ and $|\mathcal{E}_{\text{init}}| = 200$