Designing Sparse Reliable Pose-Graph SLAM: A Graph-Theoretic Approach

Kasra Khosoussi[1], Gaurav S. Sukhatme[2]
Shoudong Huang[1], and Gamini Dissanayake[1]

[1] Centre for Autonomous Systems
University of Technology Sydney
Sydney, NSW 2007, Australia
`kasra.khosoussi@uts.edu.au`

[2] Department of Computer Science
University of Southern California
Los Angeles, CA 90089, USA

Abstract. In this paper, we aim to design sparse D-optimal (determinant-optimal) pose-graph SLAM problems through the synthesis of sparse graphs with the maximum weighted number of spanning trees. Characterizing graphs with the maximum number of spanning trees is an open problem in general. To tackle this problem, several new theoretical results are established in this paper, including the monotone log-submodularity of the weighted number of spanning trees. By exploiting these structures, we design a complementary pair of near-optimal efficient approximation algorithms with provable guarantees. Our theoretical results are validated using random graphs and a publicly available pose-graph SLAM dataset.

Keywords: Number of Spanning Trees, D-Optimal Pose-Graph SLAM, Approximation Algorithms

1 Introduction

Graphs arise in modelling numerous phenomena across science and engineering. In particular, estimation-on-graph (EoG) is a class of (maximum likelihood) estimation problems with a natural graphical representation that arise especially in robotics and sensor networks. In such problems, each vertex corresponds to an unknown state, and each edge corresponds to a relative noisy measurement between the corresponding states. Simultaneous localization and mapping (SLAM) and sensor network localization (SNL) are two well-studied EoGs.

Designing sparse, yet "well-connected" graphs is a subtle task that frequently arises in various domains. First, note that graph sparsity—in EoGs and many other contexts—lead to computational efficiency. Hence, maintaining sparsity is often crucial. It is useful to see graph connectivity as a spectrum, as we often need to compare the connectivity of connected graphs. In engineering, well-connected

graphs often exhibit desirable qualities such as *reliability*, *robustness*, and *resilience* to noise, outliers, and link failures. More specifically, a well-connected EoG is more resilient to a fixed noise level and results in a more reliable estimate (i.e., smaller estimation-error covariance in the Loewner ordering sense). Consequently, maintaining a sufficient connectivity is also critical. Needless to say, sparsity is, by its very essence, at odds with well-connectivity. This is the case in SLAM, where there is a trade-off between the cost of inference and the reliability of the resulting estimate. This problem is not new. Measurement selection and pose-graph pruning have been extensively studied in the SLAM literature (see, e.g., [21, 12]). However, in this paper we take a novel graph-theoretic approach by reducing the problem of designing sparse reliable SLAM problems to the purely combinatorial problem of synthesizing sparse, yet well-connected graphs. In what follows, we briefly justify this approximate reduction.

First, note that by estimation reliability we refer to the standard D-optimality criterion, defined as the determinant of the (asymptotic) maximum likelihood estimator covariance matrix. D-optimality is a standard and popular design criterion; see, e.g., [13, 16] and the references therein. Next, we have to specify how we measure graph connectivity. Among the existing combinatorial and spectral graph connectivity criteria, the number of spanning trees (sometimes referred to as *graph complexity* or *tree-connectivity*) stands out: despite its combinatorial origin, it can also be characterized solely by the spectrum of the graph Laplacian [9]. In [16, 17, 18], we shed light on the connection between the D-criterion in SLAM—and some other EoGs—and the tree-connectivity of the underlying graph. Our theoretical and empirical results demonstrate that, under some standard conditions, D-optimality in SLAM is *significantly* influenced by the tree-connectivity of the graph underneath. Therefore, one can accurately estimate the D-criterion without using any information about the robot's trajectory or realized measurements (see Section 3). Intuitively speaking, our approach can be seen as a dimensionality reduction scheme for designing D-optimal SLAM problems from the joint space of trajectories and graph topologies to only the space of graph topologies [18].

Although this work is specifically motivated by the SLAM problem, designing sparse graphs with the maximum tree-connectivity has several other important applications. For example, it has been shown that tree-connectivity is associated with the D-optimal incomplete block designs [7, 5, 1]. Moreover, tree-connectivity is a major factor in maximizing the connectivity of certain random graphs that model unreliable networks under random link failure (*all-terminal network reliability*) [15, 30]. In particular, a classic result in network reliability theory states that if the *uniformly-most reliable* network exits, it must have the maximum tree-connectivity among all graphs with the same size [2, 23, 3].

Known Results. Graphs with the maximum weighted number of spanning trees among a family of graphs with the same vertex set are called *t-optimal*. The problem of characterizing unweighted *t*-optimal graphs among the set of graphs with n vertices and m edges remains open and has been solved *only* for specific pairs of n and m; see, e.g., [27, 5, 14, 25]. The span of these special cases

is too narrow for the types of graphs that typically arise in SLAM and sensor networks. Furthermore, in many cases the (n,m) constraint alone is insufficient for describing the true set of feasible graphs and cannot capture implicit practical constraints that exist in SLAM. Finally, it is not clear how these results can be extended to the case of (edge) weighted graphs, which are essential for representing SLAM problems, where the weight of each edge represents the precision of the corresponding pairwise measurement [18].

Contributions. This paper addresses the problem of designing sparse t-optimal graphs with the ultimate goal of designing D-optimal pose-graph SLAM problems. First and foremost, we formulate a combinatorial optimization problem that captures the measurement selection and measurement pruning scenarios in SLAM. Next, we prove that the weighted number of spanning trees, under certain conditions, is a monotone log-submodular function of the edge set. To the best of our knowledge, this is a new result in graph theory. Using this result, we prove that the greedy algorithm is near-optimal. In our second approximation algorithm, we formulate this problem as an integer program that admits a straightforward convex relaxation. Our analysis sheds light on the performance of a simple deterministic rounding procedure that have also been used in more general contexts. The proposed approximation algorithms provide near-optimality certificates. The proposed graph synthesis framework can be readily applied to any application where maximizing tree-connectivity is desired.

Notation. Throughout this paper, bold lower-case and upper-case letters are reserved for vectors and matrices, respectively. The standard basis for $\mathbb{R}^n$ is denoted by $\{\mathbf{e}_i^n\}_{i=1}^n$. Sets are shown by upper-case letters. $|\cdot|$ denotes the set cardinality. For any finite set $\mathcal{W}$, $\binom{\mathcal{W}}{k}$ is the set of all k-subsets of $\mathcal{W}$. We use $[n]$ to denote the set $\{1,2,\ldots,n\}$. The eigenvalues of symmetric matrix $\mathbf{M}$ are denoted by $\lambda_1(\mathbf{M}) \leq \cdots \leq \lambda_n(\mathbf{M})$. $\mathbf{1}$, $\mathbf{I}$ and $\mathbf{0}$ denote the vector of all ones, the identity and the zero matrices with appropriate sizes, respectively. $\mathbf{S}_1 \succ \mathbf{S}_2$ (resp. $\mathbf{S}_1 \succeq \mathbf{S}_2$) means $\mathbf{S}_1 - \mathbf{S}_2$ is positive definite (resp. positive semidefinite). Finally, $\mathrm{diag}(\mathbf{W}_1,\ldots,\mathbf{W}_k)$ is the block-diagonal matrix whose main diagonal blocks are $\mathbf{W}_1,\ldots,\mathbf{W}_k$.

2 Preliminaries

Graph Matrices. Throughout this paper, we usually refer to undirected graphs $\mathcal{G} = (\mathcal{V},\mathcal{E})$ with n vertices (labeled with $[n]$) and m edges. With a little abuse of notation, we call $\widetilde{\mathbf{A}} \in \{-1,0,1\}^{n \times m}$ the incidence matrix of $\mathcal{G}$ after choosing an arbitrary orientation for its edges. The Laplacian matrix of $\mathcal{G}$ is defined as $\widetilde{\mathbf{L}} \triangleq \widetilde{\mathbf{A}}\widetilde{\mathbf{A}}^\top$. $\widetilde{\mathbf{L}}$ can be written as $\sum_{i=1}^m \widetilde{\mathbf{L}}_{e_i}$ in which $\widetilde{\mathbf{L}}_{e_i}$ is the *elementary Laplacian* associated with edge $e_i = \{u_i,v_i\}$, where the (u_i,u_i) and (v_i,v_i) entries are 1, and the (u_i,v_i) and (v_i,u_i) entries are -1. Anchoring $v_0 \in \mathcal{V}$ is equivalent to removing the row associated with v_0 from $\widetilde{\mathbf{A}}$. Anchoring v_0 results in the *reduced* incidence matrix $\mathbf{A}$ and the *reduced* Laplacian matrix $\mathbf{L} \triangleq \mathbf{A}\mathbf{A}^\top$. $\mathbf{L}$ is also known as the *Dirichlet*. We may assign positive weights to the edges of $\mathcal{G}$ via $w : \mathcal{E} \to \mathbb{R}_{>0}$. Let $\mathbf{W} \in \mathbb{R}^{m \times m}$ be the diagonal matrix whose (i,i) entry is equal to the weight

of the ith edge. The *weighted* Laplacian (resp. reduced *weighted* Laplacian) is then defined as $\widetilde{\mathbf{L}}_w \triangleq \widetilde{\mathbf{A}}\mathbf{W}\widetilde{\mathbf{A}}^\top$ (resp. $\mathbf{L}_w \triangleq \mathbf{A}\mathbf{W}\mathbf{A}^\top$). Note that the (reduced) unweighted Laplacian is a special case of the (reduced) weighted Laplacian with $\mathbf{W} = \mathbf{I}_m$ (i.e., when all edges have unit weight).

Spanning Trees. A spanning tree of $\mathcal{G}$ is a spanning subgraph of $\mathcal{G}$ that is also a tree. Let $\mathbb{T}_\mathcal{G}$ denote the set of all spanning trees of $\mathcal{G}$. $t(\mathcal{G}) \triangleq |\mathbb{T}_\mathcal{G}|$ denotes the number of spanning trees in $\mathcal{G}$. As a generalization, for graphs whose edges are weighted by $w : \mathcal{E} \to \mathbb{R}_{>0}$, we define the *weighted number of spanning trees*,

$$t_w(\mathcal{G}) \triangleq \sum_{\mathcal{T} \in \mathbb{T}_\mathcal{G}} \mathbb{V}_w(\mathcal{T}). \tag{1}$$

We call $\mathbb{V}_w : \mathbb{T}_\mathcal{G} \to \mathbb{R}_{>0}$ the *tree value function* and define it as the product of the edge weights along a spanning tree. Notice that for unit edge weights, $t_w(\mathcal{G})$ coincides with $t(\mathcal{G})$. Thus, unless explicitly stated otherwise, we generally assume the graph is weighted. To prevent overflow and underflow, it is more convenient to work with $\log t_w(\mathcal{G})$. We formally define *tree-connectivity* as,

$$\tau_w(\mathcal{G}) \triangleq \begin{cases} \log t_w(\mathcal{G}) & \text{if } \mathcal{G} \text{ is connected,} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

For the purpose of this work, without loss of generality we can assume $w(e) \geq 1$ for all $e \in \mathcal{E}$, and thus $\tau_w(\mathcal{G}) \geq 0$.[3] The equality occurs only when either $\mathcal{G}$ is not connected, or when $\mathcal{G}$ is a tree whose all edges have unit weight. Kirchhoff's seminal matrix-tree theorem is a classic result in spectral graph theory. This theorem relates the spectrum of the Laplacian matrix of graph to its number of spanning trees. The original matrix-tree theorem states that,

$$t(\mathcal{G}) = \det \mathbf{L} \tag{3}$$

$$= \frac{1}{n} \prod_{i=2}^{n} \lambda_i(\widetilde{\mathbf{L}}). \tag{4}$$

Here $\mathbf{L}$ is the reduced Laplacian after anchoring an arbitrary vertex. Kirchhoff's matrix-tree theorem has been generalized to the case of edge-weighted graphs. According to the generalized theorem, $t_w(\mathcal{G}) = \det \mathbf{L}_w = \frac{1}{n} \prod_{i=2}^{n} \lambda_i(\widetilde{\mathbf{L}}_w)$.

Submodularity. Suppose $\mathcal{W}$ is a finite set. Consider a set function $\xi : 2^\mathcal{W} \to \mathbb{R}$. ξ is called:

1. *normalized* iff $\xi(\varnothing) = 0$.
2. *monotone* iff $\xi(\mathcal{B}) \geq \xi(\mathcal{A})$ for every $\mathcal{A}$ and $\mathcal{B}$ s.t. $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{W}$.
3. *submodular* iff for every $\mathcal{A}$ and $\mathcal{B}$ s.t. $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{W}$ and $\forall s \in \mathcal{W} \setminus \mathcal{B}$ we have,

$$\xi(\mathcal{A} \cup \{s\}) - \xi(\mathcal{A}) \geq \xi(\mathcal{B} \cup \{s\}) - \xi(\mathcal{B}). \tag{5}$$

---

[3] Replacing any $w : \mathcal{E} \to \mathbb{R}_{\geq 0}$ with $w' : \mathcal{E} \to \mathbb{R}_{\geq 1} : e \mapsto \alpha_w w(e)$ for a sufficiently large constant α_w does not affect the set of t-optimal graphs.

3 D-Optimality via Graph Synthesis

In this section, we discuss the connection between D-optimality and t-optimality in SLAM by briefly reviewing the results in [16, 17, 18]. Consider the 2-D pose-graph SLAM problem where each measurement consists of the rotation (angle) and translation between a pair of robot poses over time, corrupted by an independently-drawn additive zero-mean Gaussian noise. According to our model, the covariance matrix of the noise vector corrupting the ith measurement can be written as $\mathrm{diag}(\sigma_{p_i}^2 \mathbf{I}_2, \sigma_{\theta_i}^2)$, where $\sigma_{p_i}^2$ and $\sigma_{\theta_i}^2$ denote the translational and rotational noise variances, respectively. As mentioned earlier, SLAM, as an EoG problem, admits a natural graphical representation $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in which poses correspond to graph vertices and edges correspond to the relative measurements between the corresponding poses. Furthermore, measurement precisions are incorporated into our model by assigning positive weights to the edges of $\mathcal{G}$. Note that for each edge we have two separate weight functions w_p and w_θ, defined as $w_p : e_i \mapsto \sigma_{p_i}^{-2}$ and $w_\theta : e_i \mapsto \sigma_{\theta_i}^{-2}$.

Let $\mathbb{Var}[\hat{\mathbf{x}}_{\mathsf{mle}}]$ be the asymptotic covariance matrix of the maximum likelihood estimator (Cramér-Rao lower bound) for estimating the trajectory $\mathbf{x}$. In [16, 17, 18], we investigated the impact of graph topology on the D-optimality criterion ($\det \mathbb{Var}[\hat{\mathbf{x}}_{\mathsf{mle}}]$) in SLAM. The results presented in [18] are threefold. First, in [18, Proposition 2] it is proved that

$$-2\,\tau_{w_p}(\mathcal{G}) - \log\det(\mathbf{L}_{w_\theta} + \gamma\mathbf{I}) \leq \log\det \mathbb{Var}[\hat{\mathbf{x}}_{\mathsf{mle}}] \leq -2\,\tau_{w_p}(\mathcal{G}) - \tau_{w_\theta}(\mathcal{G}) \quad (6)$$

in which γ is a parameter whose value depends on the maximum distance between the neighbouring robot poses normalized by $\sigma_{p_i}^2$'s; e.g., this parameter shrinks by reducing the distance between the neighbouring poses, or by reducing the precision of the translational measurements (see [18, Remark 2]). Next, based on this result, it is easy to see that [18, Theorem 5],

$$\lim_{\gamma \to 0^+} \log\det \mathbb{Var}[\hat{\mathbf{x}}_{\mathsf{mle}}] = -2\,\tau_{w_p}(\mathcal{G}) - \tau_{w_\theta}(\mathcal{G}). \quad (7)$$

Note that the expression above depends only on the graphical representation of the problem. Finally, the empirical observations and Monte Carlo simulations based on a number of synthetic and real datasets indicate that the RHS of (7) provides a reasonable estimate for $\log\det \mathbb{Var}[\hat{\mathbf{x}}_{\mathsf{mle}}]$ even in the non-asymptotic regime where γ is not negligible. In what follows, we demonstrate how these results can be used in a graph-theoretic approach to the D-optimal measurement selection and pruning problems.

Measurement Selection. Maintaining sparsity is essential for computational efficiency in SLAM, especially in long-term autonomy. Sparsity can be preserved by implementing a measurement selection policy to asses the significance of new or existing measurements. Such a vetting process can be realized by (i) assessing the significance of any new measurement before adding it to the graph, and/or (ii) pruning a subset of the acquired measurements if their contribution is deemed to be insufficient. These ideas have been investigated in the literature; for the former approach see, e.g., [13, 26], and see, e.g., [21, 12] for the latter.

Now consider the D-optimal measurement selection problem whose goal is to select the optimal k-subset of measurements such that the resulting $\log \det \mathbb{V}\mathrm{ar}[\hat{\mathbf{x}}_{\mathsf{mle}}]$ is minimized. This problem is closely related to the D-optimal sensor selection problem for which two successful approximation algorithms have been proposed in [13] and [26] under the assumption of linear sensor models. The measurement models in SLAM are nonlinear. Nevertheless, we can still use [13, 26] after linearizing the measurement model. Note that the Fisher information matrix and $\log \det \mathbb{V}\mathrm{ar}[\hat{\mathbf{x}}_{\mathsf{mle}}]$ in SLAM depend on the true $\mathbf{x}$. Since the true value of $\mathbf{x}$ is not available, in practice these terms are approximated by evaluating the Jacobian matrix at the estimate obtained by maximizing the log-likelihood function using an iterative solver.

An alternative approach would be to replace $\log \det \mathbb{V}\mathrm{ar}[\hat{\mathbf{x}}_{\mathsf{mle}}]$ with a graph-theoretic objective function based on (7). Note that this is equivalent to reducing the original problem into a graph synthesis problem. The graphical approach has the following advantages:

1. *Robustness*: Maximum likelihood estimation in SLAM boils down to solving a non-convex optimization problem via iterative solvers. These solvers are subject to local minima. Hence, the approximated $\log \det \mathbb{V}\mathrm{ar}[\hat{\mathbf{x}}_{\mathsf{mle}}]$ can be highly inaccurate and lead to misleading designs if the Jacobian is evaluated at a local minimum (see [18, Section VI] for an example). The graph-theoretic objective function based on (7), however, is independent of the trajectory $\mathbf{x}$ and, therefore, is robust to such convergence errors.

2. *Flexibility*: To directly compute $\log \det \mathbb{V}\mathrm{ar}[\hat{\mathbf{x}}_{\mathsf{mle}}]$, we first need a nominal or estimated trajectory $\mathbf{x}$. Furthermore, for the latter we also need to know the realization of relative measurements. Therefore, any design or decisions made in this way will be confined to a particular trajectory. On the contrary, the graphical approach requires only the knowledge of the topology of the graph, and thus is more flexible. Note that the t-optimal topology corresponds to a range of trajectories. Therefore, the graphical approach enables us to assess the D-optimality of a particular design with minimum information and without relying on any particular—planned, nominal or estimated—trajectory.

We will investigate the problem of designing t-optimal graphs in Section 4.

4 Synthesis of Near-t-Optimal Graphs

Problem Formulation. In this section, we formulate and tackle the combinatorial optimization problem of designing sparse graphs with the maximum weighted tree-connectivity. Since the decision variables are the edges of the graph, it is more convenient to treat the weighted tree-connectivity as a function of the edge set of the graph for a given set of vertices $(\mathcal{V} = [n])$ and a positive weight function $w : \binom{[n]}{2} \to \mathbb{R}_{\geq 1}$. $\mathsf{tree}_{n,w} : 2^{\binom{[n]}{2}} \to \mathbb{R}_{\geq 0} : \mathcal{E} \mapsto \tau_w([n],\mathcal{E})$ takes as input a set of edges $\mathcal{E}$ and returns the weighted tree-connectivity of graph $([n],\mathcal{E})$. To simplify our notation, hereafter we drop n and/or w from $\mathsf{tree}_{n,w}$ (and similar terms) whenever n and/or w are clear from the context.

Problem 1 (k-ESP). Suppose the following are given:

- a *base graph* $\mathcal{G}_{\text{init}} = ([n], \mathcal{E}_{\text{init}})$
- a weight function $w : \binom{[n]}{2} \to \mathbb{R}_{\geq 1}$
- a set of c *candidate* edges (either $\mathcal{C}^+$ or $\mathcal{C}^-$)
- an integer $k \leq c$

Consider the following edge selection problems (ESP):

$\diamond$ k-ESP$^+$

$$\underset{\mathcal{E} \subseteq \mathcal{C}^+ \subseteq \binom{[n]}{2} \setminus \mathcal{E}_{\text{init}}}{\text{maximize}} \quad \mathsf{tree}(\mathcal{E}_{\text{init}} \cup \mathcal{E}) \quad \text{subject to} \quad |\mathcal{E}| = k. \tag{8}$$

$\diamond$ k-ESP$^-$

$$\underset{\mathcal{E} \subseteq \mathcal{C}^- \subseteq \mathcal{E}_{\text{init}}}{\text{maximize}} \quad \mathsf{tree}(\mathcal{E}_{\text{init}} \setminus \mathcal{E}) \quad \text{subject to} \quad |\mathcal{E}| = k. \tag{9}$$

Remark 1. It is easy to see that any instance of (8) can be expressed as an instance of (9) and vice versa. Therefore, without loss of generality, in this work we only consider k-ESP$^+$.

1-ESP$^+$. Consider the simple case of $k = 1$. $\Delta_{uv} \triangleq \mathbf{a}_{uv} \mathbf{L}^{-1} \mathbf{a}_{uv}$ is known as the *effective resistance* between vertices u and v. Here $\mathbf{a}_{uv} \in \{-1, 0, 1\}^{n-1}$ is the vector $\mathbf{e}_u^n - \mathbf{e}_v^n$ after crossing out the entry that corresponds to the anchored vertex. Effective resistance has emerged from several other contexts as a key factor; see, e.g., [8]. In [19, Lemma 3.1] it is shown that the optimal choice in 1-ESP$^+$ is the candidate edge with the maximum $w(e)\Delta_e$. The effective resistance can be efficiently computed by performing a Cholesky decomposition on the reduced weighted Laplacian matrix of the base graph $\mathbf{L}_{\text{init}}$ and solving a triangular linear system (see [19]). In the worst case and for a dense base graph 1-ESP$^+$ can be solved in $O(n^3 + c n^2)$ time.

4.1 Approximation Algorithms for k-ESP$^+$

Solving the general case of k-ESP$^+$ by exhaustive search requires examining $\binom{c}{k}$ graphs. This is not practical even when c is bounded (e.g., for $c = 30$ and $k = 10$ we need to perform more than 3×10^7 Cholesky factorizations). Here we propose a complementary pair of approximation algorithms.

I: Greedy. The greedy algorithm finds an approximate solution to k-ESP$^+$ by decomposing it into a sequence of k 1-ESP$^+$ problems, each of which can be solved using the procedure outlined above. After solving each subproblem, the optimal edge is moved from the candidate set to the base graph. The next 1-ESP$^+$ subproblem is defined using the updated candidate set and the updated

base graph. If the graph is dense, a naive implementation of the greedy algorithm requires less than $O(kcn^3)$ operations. An efficient implementation of this approach that requires $O(n^3 + kcn^2)$ time is described in [19, Algorithm 1].

Analysis. Let $\mathcal{G}_{\text{init}} = ([n], \mathcal{E}_{\text{init}})$ be a *connected* base graph and $w : \binom{[n]}{2} \to \mathbb{R}_{\geq 1}$. Consider the following function.

$$\mathfrak{X}_w : \mathcal{E} \mapsto \mathsf{tree}(\mathcal{E} \cup \mathcal{E}_{\text{init}}) - \mathsf{tree}(\mathcal{E}_{\text{init}}). \tag{10}$$

In k-ESP$^+$, we restrict the domain of $\mathfrak{X}_w$ to $2^{\mathcal{C}^+}$. Note that $\mathsf{tree}(\mathcal{E}_{\text{init}})$ is a constant and, therefore, we can express the objective function in k-ESP$^+$ using $\mathfrak{X}_w$,

$$\operatorname*{maximize}_{\mathcal{E} \subseteq \mathcal{C}^+} \quad \mathfrak{X}_w(\mathcal{E}) \quad \text{subject to} \quad |\mathcal{E}| = k. \tag{11}$$

Theorem 1. $\mathfrak{X}_w$ *is normalized, monotone and submodular.*

Proof. Omitted due to space limitation—see the technical report [19].

Maximizing an arbitrary monotone submodular function subject to a cardinality constraint can be NP-hard in general (see, e.g., the Maximum Coverage problem [11]). Nemhauser et al. [24] in their seminal work have shown that the greedy algorithm is a constant-factor approximation algorithm with a factor of $\eta \triangleq (1 - 1/e) \approx 0.63$ for any (normalized) monotone submodular function subject to a cardinality constraint. Let OPT be the optimum value of (8), $\mathcal{E}_{\text{greedy}}$ be the edges selected by the greedy algorithm, $\tau_{\text{greedy}} \triangleq \mathsf{tree}(\mathcal{E}_{\text{greedy}} \cup \mathcal{E}_{\text{init}})$ and $\tau_{\text{init}} \triangleq \mathsf{tree}(\mathcal{E}_{\text{init}})$.

Corollary 1. $\tau_{\text{greedy}} \geq \eta\,\mathsf{OPT} + (1 - \eta)\,\tau_{\text{init}}.$

II: Convex Relaxation. In this section, we design an approximation algorithm for k-ESP$^+$ through convex relaxation. We begin by assigning an auxiliary variable $0 \leq \pi_i \leq 1$ to each candidate edge $e_i \in \mathcal{C}^+$. The idea is to reformulate the problem such that finding the optimal set of candidate edges is equivalent to finding the optimal value for π_i's. Let $\boldsymbol{\pi} \triangleq [\pi_1 \ \pi_2 \ \cdots \ \pi_c]^\top$ be the stacked vector of auxiliary variables. Define,

$$\mathbf{L}_w(\boldsymbol{\pi}) \triangleq \mathbf{L}_{\text{init}} + \sum_{e_i \in \mathcal{C}^+} \pi_i w(e_i) \mathbf{L}_{e_i} = \mathbf{A} \mathbf{W}^\pi \mathbf{A}^\top, \tag{12}$$

where $\mathbf{L}_{e_i}$ is the reduced elementary Laplacian, $\mathbf{A}$ is the reduced incidence matrix of $\mathcal{G}_\bullet \triangleq ([n], \mathcal{E}_{\text{init}} \cup \mathcal{C}^+)$, and $\mathbf{W}^\pi$ is the diagonal matrix of edge weights assigned by the following weight function,

$$w^\pi(e_i) = \begin{cases} \pi_i w(e_i) & e_i \in \mathcal{C}^+, \\ w(e_i) & e_i \notin \mathcal{C}^+. \end{cases} \tag{13}$$

Lemma 1. *If $\mathcal{G}_{\text{init}}$ is connected, $\mathbf{L}_w(\boldsymbol{\pi})$ is positive definite for any $\boldsymbol{\pi} \in [0,1]^c$.*

As before, for convenience we assume $\mathcal{G}_{\text{init}}$ is connected. Consider the following optimization problems over $\boldsymbol{\pi}$.

$$
\begin{aligned}
\underset{\boldsymbol{\pi}}{\text{maximize}} \quad & \log \det \mathbf{L}_w(\boldsymbol{\pi}) \\
\text{subject to} \quad & \|\boldsymbol{\pi}\|_0 = k, \\
& 0 \le \pi_i \le 1,\ \forall i \in [c].
\end{aligned} \tag{P_1}
$$

$$
\begin{aligned}
\underset{\boldsymbol{\pi}}{\text{maximize}} \quad & \log \det \mathbf{L}_w(\boldsymbol{\pi}) \\
\text{subject to} \quad & \|\boldsymbol{\pi}\|_1 = k, \\
& \pi_i \in \{0,1\},\ \forall i \in [c].
\end{aligned} \tag{P_1'}
$$

P_1 is equivalent to our original definition of k-ESP$^+$. First, note that from the generalized matrix-tree theorem we know that the objective function is equal to the weighted tree-connectivity of graph $\mathcal{G}_\bullet = ([n], \mathcal{E}_{\text{init}} \cup \mathcal{C}^+)$ whose edges are weighted by w^π. The auxiliary variables act as selectors: the ith candidate edge is selected iff $\pi_i = 1$. The combinatorial difficulty of k-ESP$^+$ here is embodied in the non-convex ℓ_0-norm constraint. It is easy to see that in P_1, at the optimal solution, auxiliary variables take binary values. This is why the integer program P_1' is equivalent to P_1. A natural choice for relaxing P_1' is to replace $\pi_i \in \{0,1\}$ with $0 \le \pi_i \le 1$; i.e.,

$$
\begin{aligned}
\underset{\boldsymbol{\pi}}{\text{maximize}} \quad & \log \det \mathbf{L}_w(\boldsymbol{\pi}) \\
\text{subject to} \quad & \|\boldsymbol{\pi}\|_1 = k, \\
& 0 \le \pi_i \le 1,\ \forall i \in [c].
\end{aligned} \tag{P_2}
$$

The feasible set of P_2 contains that of P_1'. Hence, the optimum value of P_2 is an upper bound for the optimum of P_1 (or, equivalently, P_1'). Note that the ℓ_1-norm constraint here is identical to $\sum_{i=1}^c \pi_i = k$. P_2 is a convex optimization problem since the objective function (tree-connectivity) is concave and the constraints are linear and affine in $\boldsymbol{\pi}$. In fact, P_2 is an instance of the MAXDET problem [29] subject to additional affine constraints on $\boldsymbol{\pi}$. It is worth noting that P_2 can be reached also by relaxing the non-convex ℓ_0-norm constraint in P_1 into the convex ℓ_1-norm constraint $\|\boldsymbol{\pi}\|_1 = k$. Furthermore, P_2 is also closely related to a ℓ_1-regularised variant of MAXDET,

$$
\begin{aligned}
\underset{\boldsymbol{\pi}}{\text{maximize}} \quad & \log \det \mathbf{L}_w(\boldsymbol{\pi}) - \lambda \|\boldsymbol{\pi}\|_1 \\
\text{subject to} \quad & 0 \le \pi_i \le 1,\ \forall i \in [c].
\end{aligned} \tag{P_3}
$$

This problem is a penalized form of P_2; these two problems are equivalent for some positive value of λ. Problem P_3 is also a convex optimization problem for any non-negative λ. The ℓ_1-norm in P_3 penalizes the loss of sparsity, while the log-determinant rewards stronger tree-connectivity. λ is a parameter that controls the sparsity of the resulting graph; i.e., a larger λ yields a sparser vector of selectors $\boldsymbol{\pi}$. P_3 is closely related to graphical lasso [6]. P_2 (and P_3) can be solved globally in polynomial time using interior-point methods [4, 13]. After finding a globally optimal solution $\boldsymbol{\pi}^\star$ for the relaxed problem P_2, we ultimately need to map it into a feasible $\boldsymbol{\pi}$ for P_1'; i.e., choosing k edges from the candidate set $\mathcal{C}^+$.

Lemma 2. $\boldsymbol{\pi}^\star$ *is an optimal solution for k-ESP$^+$ iff $\boldsymbol{\pi}^\star \in \{0,1\}^c$.*

Rounding. In general, $\boldsymbol{\pi}^{\star}$ may contain fractional values that need to be mapped into feasible integral values for P'_1 by a *rounding procedure* that sets k auxiliary variables to one and others to zero. The most intuitive deterministic rounding policy is to pick the k edges with the largest $\pi_i^{\star}$'s.

The idea behind the convex relaxation technique described so far can be seen as a graph-theoretic special case of the algorithm proposed in [13]. However, it is not clear yet how the solution of the relaxed convex problem P_2 is related to that of the original non-convex k-ESP$^+$ in the integer program P'_1. To answer this question, consider the following randomized strategy. We may attempt to find a suboptimal solution for k-ESP$^+$ by randomly sampling candidates. In this case, for the ith candidate edge, we flip a coin whose probability of heads is π_i (independent of other candidates). We then select that candidate edge if the coin lands on head.

Theorem 2. *Let the random variables k^* and t_w^* denote, respectively, the number of chosen candidate edges and the corresponding weighted number of spanning trees achieved by the above randomized algorithm. Then,*

$$\mathbb{E}\left[k^*\right] = \sum_{i=1}^{c} \pi_i, \tag{14}$$

$$\mathbb{E}\left[t_w^*\right] = \det \mathbf{L}_w(\boldsymbol{\pi}). \tag{15}$$

Proof. See [19] for the proof.[4]

According to Theorem 2, the randomized algorithm described above on average selects $\sum_{i=1}^{c} \pi_i$ candidate edges and achieves $\det \mathbf{L}_w(\boldsymbol{\pi})$ weighted number of spanning trees in expectation. Note that these two terms appear in the constraints and objective of the relaxed problem P_2, respectively. Therefore, the relaxed problem can be interpreted as the problem of finding the optimal sampling probabilities $\boldsymbol{\pi}$ for the randomized algorithm described above. This offers a new narrative:

Corollary 2. *The objective in P_2 is to find the optimal probabilities $\boldsymbol{\pi}^{\star}$ for sampling edges from $\mathcal{C}^+$ such that the weighted number of spanning trees is maximized in* expectation, *while the* expected *number of newly selected edges is equal to k.*

In other words, P_2 can be seen as a convex relaxation of P_1 at the expense of maximizing the objective and satisfying the constraint, both *in expectation*. This new interpretation can be used as a basis for designing randomized rounding procedures based on the randomized technique described above. If one uses $\boldsymbol{\pi}^{\star}$ (the fractional solution of the relaxed problem P_2) in the aforementioned randomized rounding scheme, Theorem 2 ensures that, on average, such a method attains $\det \mathbf{L}(\boldsymbol{\pi}^{\star})$ by picking k new edges in expectation. Finally, we note that this new

---

[4] A generalized version of this theorem that covers the more general case of [13] is proved in [19].

interpretation sheds light on why the deterministic rounding policy described earlier performs well in practice. Note that randomly sampling candidate edges with the probabilities in $\boldsymbol{\pi}^\star$ does not necessarily result in a feasible solution for P_1'. That being said, consider every feasible outcome in which exactly k candidate edges are selected by the randomized algorithm with probabilities in $\boldsymbol{\pi}^\star$. It is easy to show that the deterministic procedure described earlier (picking k candidates with the largest $\pi_i^\star$'s) is in fact selecting the most probable feasible outcome (given that exactly k candidates have been selected).

Near-Optimality Certificates. It is impractical to compute OPT via exhaustive search in large problems. Nevertheless, the approximation algorithms described above yield lower and upper bounds for OPT that can be quite tight in practice. Let $\tau_{\mathrm{cvx}}^\star$ be the optimum value of P_2. Moreover, let τ_{cvx} be the suboptimal value obtained after rounding the solution of P_2 (e.g., picking the k largest $\pi_i^\star$'s). The following corollary readily follows from the analysis of the greedy and convex approximation algorithms.

Corollary 3.

$$\max\left\{\tau_{\mathrm{greedy}}, \tau_{\mathrm{cvx}}\right\} \leq \mathsf{OPT} \leq \min\left\{\mathcal{U}_{\mathrm{greedy}}, \tau_{\mathrm{cvx}}^\star\right\} \tag{16}$$

where $\mathcal{U}_{\mathrm{greedy}} \triangleq \zeta\tau_{\mathrm{greedy}} + (1-\zeta)\tau_{\mathrm{init}}$ *in which* $\zeta \triangleq \eta^{-1} \approx 1.58$.

The bounds in Corollary 3 can be computed by running the greedy and convex relaxation algorithms. Whenever OPT is beyond reach, the upper bound can be used to asses the quality of any feasible design. Let $\mathcal{S}$ be an arbitrary k-subset of $\mathcal{C}^+$ and $\tau_{\mathcal{S}} \triangleq \mathsf{tree}(\mathcal{S} \cup \mathcal{E}_{\mathrm{init}})$. $\mathcal{S}$ can be, e.g., the solution of greedy algorithm, the solution of P_2 after rounding, an existing design (e.g., an existing pose-graph problem) or a suboptimal solution proposed by a third party. Let $\mathcal{L}$ and $\mathcal{U}$ denote the lower and upper bounds in (16), respectively. From Corollary 3 we have,

$$\max\left\{0, \mathcal{L} - \tau_{\mathcal{S}}\right\} \leq \underbrace{\mathsf{OPT} - \tau_{\mathcal{S}}}_{\text{optimality gap}} \leq \mathcal{U} - \tau_{\mathcal{S}}. \tag{17}$$

Therefore, $\mathcal{U} - \tau_{\mathcal{S}}$ (or similarly, $\mathcal{U}/\tau_{\mathcal{S}} \geq \mathsf{OPT}/\tau_{\mathcal{S}}$) can be used as a near-optimality certificate for an arbitrary design $\mathcal{S}$.

Two Weight Functions. In the synthesis problem studied so far, it was implicitly assumed that each edge is weighted by a single weight function. This is not necessarily the case in SLAM, where each measurement has two components, each of which has its own precision, i.e., w_p and w_θ in (7). Hence, we need to revisit the synthesis problem in a more general setting, where multiple weight functions assign weights, simultaneously, to a single edge. It turns out that the proposed approximation algorithms and their analyses can be easily generalized to handle this case.

1. *Greedy Algorithm*: For the greedy algorithm, we just need to replace $\mathcal{X}_w$ with $\mathcal{Y}_w : \mathcal{E} \mapsto 2\,\mathcal{X}_{w_p}(\mathcal{E}) + \mathcal{X}_{w_\theta}(\mathcal{E})$; see (7). Note that $\mathcal{Y}_w$ is a linear combination of

normalized monotone submodular functions with positive weights, and therefore is also normalized, monotone and submodular.

2. *Convex Relaxation*: The convex relaxation technique can be generalized to the case of multi-weighted edges by replacing the concave objective function $\log \det \mathbf{L}_w(\boldsymbol{\pi})$ with $2 \log \det \mathbf{L}_{w_p}(\boldsymbol{\pi}) + \log \det \mathbf{L}_{w_\theta}(\boldsymbol{\pi})$, which is also concave.

Remark 2. Recall that our goal was to design sparse, yet reliable SLAM problems. So far we considered the problem of designing D-optimal SLAM problems with a given number of edges. The dual approach would be to find the sparsest SLAM problem such that the determinant of the estimation-error covariance is less than a desired threshold. Take for example the following scenario: find the sparsest SLAM problem by selecting loop-closure measurements from a given set of candidates such that the resulting D-criterion is 50% smaller than that of dead reckoning. The dual problem can be written as,

$$\underset{\mathcal{E} \subseteq \mathcal{C}^+}{\text{minimize}} \quad |\mathcal{E}| \quad \text{subject to} \quad \mathcal{X}_w(\mathcal{E}) \geq \tau_{\mathsf{min}}. \tag{18}$$

in which τ_{min} is given. In [19] we have shown that our proposed approximation algorithms and their analyses can be easily modified to address the dual problem. Due to space limitation, we have to refrain from discussing the dual problem in this paper.

4.2 Experimental Results

The proposed algorithms were implemented in MATLAB. P_2 is modelled using YALMIP [22] and solved using SDPT3 [28].

Random Graphs. Figure 1 illustrates the performance of our approximate algorithms in randomly generated graphs. The set of candidates in these experiments is $\mathcal{C}^+ = \binom{[n]}{2} \setminus \mathcal{E}_{\text{init}}$. Figures 1a and 1b show the resulting tree-connectivity as a function of the number of randomly generated edges for a fixed $k = 5$ and, respectively, $n = 20$ and $n = 50$. Our results indicate that both algorithms exhibit remarkable performances for $k = 5$. Note that computing OPT by exhaustive search is only feasible in small instances such as Figure 1a. Nevertheless, computing the exact OPT is not crucial for evaluating our approximate algorithms, as Corollary 3 guarantees that $\tau^\star_{\text{greedy}} \leq \text{OPT} \leq \tau^\star_{\text{cvx}}$; i.e., the space between each black · and the corresponding green ×. Figure 1c shows the results obtained for varying k. The optimality gap for τ_{cvx} gradually grows as the planning horizon k increases. Our greedy algorithm, however, still yields a near-optimal approximation.

Real Pose-Graph Dataset. We also evaluated the proposed algorithms on the Intel Research Lab dataset as a popular pose-graph SLAM benchmark.[5] In this scenario, $\mathcal{E}_{\text{init}}$ is chosen to be the set of odometry edges, and $\mathcal{C}^+$ is the set
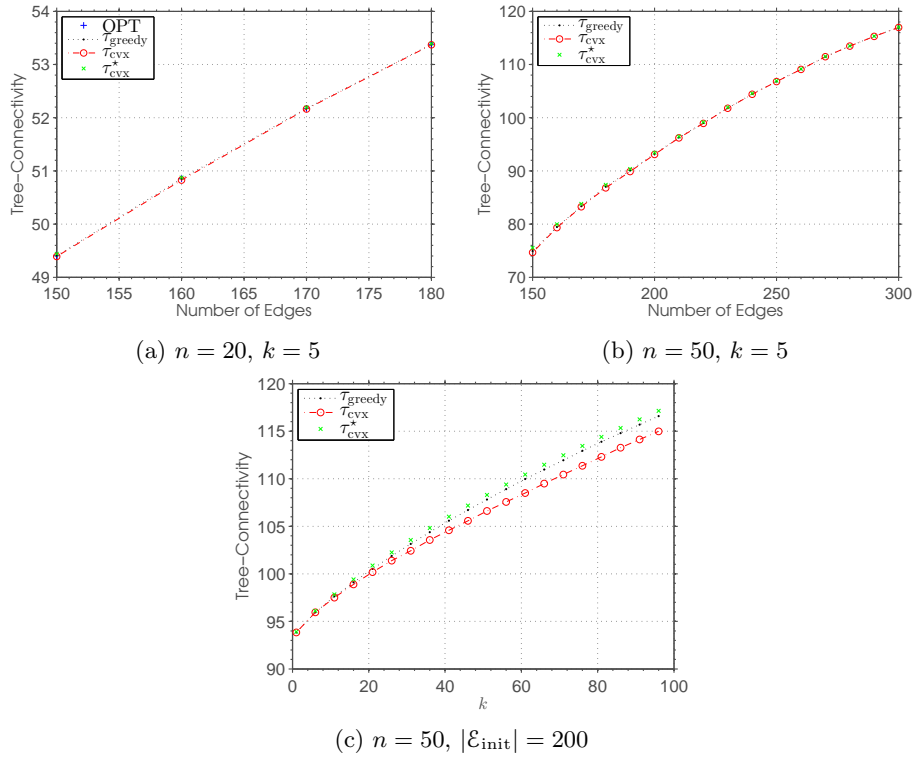
---

[5] https://svn.openslam.org/data/svn/g2o/trunk/data/2d/intel/intel.g2o

(a) $n = 20$, $k = 5$
(b) $n = 50$, $k = 5$

(c) $n = 50$, $|\mathcal{E}_{\text{init}}| = 200$

Fig. 1: k-ESP$^+$ on randomly generated graphs with $\mathcal{C}^+ = \binom{[n]}{2} \setminus \mathcal{E}_{\text{init}}$.

of loop closures. The parameters in this graph are $n = 943$, $|\mathcal{E}_{\text{init}}| = 942$ and $|\mathcal{C}^+| = 895$. Note that computing the true OPT via exhaustive search is clearly impractical; e.g., for $k = 100$, there are more than 10^{134} possible graphs. For the edge weights, we are using the original information (precisions) reported in the dataset. Since the translational and rotational measurements have different precisions, two weight functions—w_p and w_θ—assign weights to each edge of the graph, and the objective is to maximize $2\,\tau_{w_p}(\mathcal{G}) + \tau_{w_\theta}(\mathcal{G})$. Figure 2 shows the resulting objective value for the greedy and convex relaxation approximation algorithms, as well as the upper bounds ($\mathcal{U}$) in Corollary 3.[6] According to Figure 2, both algorithms have successfully found near-t-optimal (near-D-optimal) designs. The greedy algorithm has outperformed the convex relaxation with the simple deterministic (sorting) rounding procedure. For small values of k, the upper bound $\mathcal{U}$ on OPT is given by $\mathcal{U}_{\text{greedy}}$ (blue curve). However, for $k \geq 60$, the convex relaxation provides a significantly tighter upper bound on OPT (green curve). In this dataset, YALMIP+SDPT3 on an Intel Core i5-2400 operating at 3.1 GHz can solve the convex program in about 20-50 seconds, while a naive

---

[6] See also https://youtu.be/5JZF2QiRbDE for a visualization.
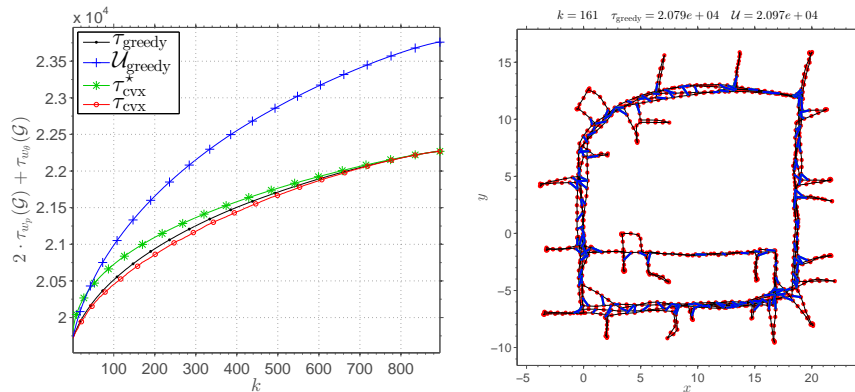
(a) Performance of the proposed approximation algorithms.

(b) Greedy design for $k = 161$ loop closures (out of 895). Loop-closure edges are shown in blue.

Fig. 2: k-ESP$^+$ for pose-graph SLAM on the Intel Research Lab dataset.

implementation of the greedy algorithm (without using rank-one updates) can solve the case with $k = 400$ in about 25 seconds.

5   Conclusion

We presented a graph-theoretic approach to the problem of designing sparse reliable (i.e., near-D-optimal) pose-graph SLAM. This paper demonstrated that this problem boils down to a combinatorial optimization problem whose goal is to find a sparse graph with the maximum weighted number of spanning trees. The problem of characterizing t-optimal graphs is an open problem with—to the best of our knowledge—no known efficient algorithm. We designed two efficient approximation algorithms with provable guarantees and near-optimality certificates. First and foremost, we introduced a new submodular graph invariant, i.e., weighted tree-connectivity. This was used to guarantee that the greedy algorithm is a constant-factor approximation algorithm for this problem with a factor of $(1 - 1/e)$ (up to a constant normalizer). In another approach, we formulated the original combinatorial optimization problem as an integer program that admits a natural convex relaxation. We discussed deterministic and randomized rounding schemes. Our analysis sheds light on the connection between the original and the relaxed problems. Finally, we evaluated the performance of the proposed approximation algorithms using random graphs and a real pose-graph SLAM dataset. Although this paper specifically targeted SLAM, we note that the proposed algorithms can be readily used to synthesize near-t-optimal graphs in any domain where maximizing tree-connectivity is useful. See, e.g., [10, 20, 3, 19] for applications in Chemistry, RNA modelling, network reliability under random link failure and estimation over sensor networks, respectively.

Bibliography

[1] Bailey, R.A., Cameron, P.J.: Combinatorics of optimal designs. Surveys in Combinatorics **365** (2009) 19–73

[2] Bauer, D., Boesch, F.T., Suffel, C., Van Slyke, R.: On the validity of a reduction of reliable network design to a graph extremal problem. Circuits and Systems, IEEE Transactions on **34**(12) (1987) 1579–1581

[3] Boesch, F.T., Satyanarayana, A., Suffel, C.L.: A survey of some network reliability analysis and synthesis results. Networks **54**(2) (2009) 99–107

[4] Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge university press (2004)

[5] Cheng, C.S.: Maximizing the total number of spanning trees in a graph: two related problems in graph theory and optimum design theory. Journal of Combinatorial Theory, Series B **31**(2) (1981) 240–248

[6] Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso. Biostatistics **9**(3) (2008) 432–441

[7] Gaffke, N.: D-optimal block designs with at most six varieties. Journal of Statistical Planning and Inference **6**(2) (1982) 183–200

[8] Ghosh, A., Boyd, S., Saberi, A.: Minimizing effective resistance of a graph. SIAM review **50**(1) (2008) 37–66

[9] Godsil, C., Royle, G.: Algebraic graph theory. Graduate Texts in Mathematics Series. Springer London, Limited (2001)

[10] Gutman, I., Mallion, R., Essam, J.: Counting the spanning trees of a labelled molecular-graph. Molecular Physics **50**(4) (1983) 859–877

[11] Hochbaum, D.S.: Approximation algorithms for NP-hard problems. PWS Publishing Co. (1996)

[12] Huang, G., Kaess, M., Leonard, J.J.: Consistent sparsification for graph optimization. In: Mobile Robots (ECMR), 2013 European Conference on, IEEE (2013) 150–157

[13] Joshi, S., Boyd, S.: Sensor selection via convex optimization. Signal Processing, IEEE Transactions on **57**(2) (2009) 451–462

[14] Kelmans, A.K.: On graphs with the maximum number of spanning trees. Random Structures & Algorithms **9**(1-2) (1996) 177–192

[15] Kelmans, A.K., Kimelfeld, B.: Multiplicative submodularity of a matrix's principal minor as a function of the set of its rows and some combinatorial applications. Discrete Mathematics **44**(1) (1983) 113–116

[16] Khosoussi, K., Huang, S., Dissanayake, G.: Novel insights into the impact of graph structure on SLAM. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2014. (2014) 2707–2714

[17] Khosoussi, K., Huang, S., Dissanayake, G.: Good, bad and ugly graphs for SLAM. RSS Workshop on the problem of mobile sensors (2015)

[18] Khosoussi, K., Huang, S., Dissanayake, G.: Tree-connectivity: A metric to evaluate the graphical structure of SLAM problems. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2016)

[19] Khosoussi, K., Sukhatme, G.S., Huang, S., Dissanayake, G.: Maximizing the weighted number of spanning trees: Near-t-optimal graphs. arXiv:1604.01116 (2016)

[20] Kim, N., Petingi, L., Schlick, T.: Network theory tools for RNA modeling. WSEAS transactions on mathematics **9**(12) (2013) 941

[21] Kretzschmar, H., Stachniss, C., Grisetti, G.: Efficient information-theoretic graph pruning for graph-based slam with laser range finders. In: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. (2011) 865 –871

[22] Löfberg, J.: Yalmip : A toolbox for modeling and optimization in MATLAB. In: Proceedings of the CACSD Conference, Taipei, Taiwan (2004)

[23] Myrvold, W.: Reliable network synthesis: Some recent developments. In: Proceedings of International Conference on Graph Theory, Combinatorics, Algorithms, and Applications. (1996)

[24] Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions - I. Mathematical Programming **14**(1) (1978) 265–294

[25] Petingi, L., Rodriguez, J.: A new technique for the characterization of graphs with a maximum number of spanning trees. Discrete mathematics **244**(1) (2002) 351–373

[26] Shamaiah, M., Banerjee, S., Vikalo, H.: Greedy sensor selection: Leveraging submodularity. In: 49th IEEE conference on decision and control (CDC), IEEE (2010) 2572–2577

[27] Shier, D.: Maximizing the number of spanning trees in a graph with n nodes and m edges. Journal Research National Bureau of Standards, Section B **78** (1974) 193–196

[28] Tütüncü, R.H., Toh, K.C., Todd, M.J.: Solving semidefinite-quadratic-linear programs using sdpt3. Mathematical programming **95**(2) (2003) 189–217

[29] Vandenberghe, L., Boyd, S., Wu, S.P.: Determinant maximization with linear matrix inequality constraints. SIAM journal on matrix analysis and applications **19**(2) (1998) 499–533

[30] Weichenberg, G., Chan, V.W., Médard, M.: High-reliability topological architectures for networks under stress. Selected Areas in Communications, IEEE Journal on **22**(9) (2004) 1830–1845