Article

The International Journal of
Robotics Research
2019, Vol. 38(2-3) 260–298
© The Author(s) 2019
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0278364918823086
journals.sagepub.com/home/ijr

Reliable Graphs for SLAM

Kasra Khosoussi[1], Matthew Giamou[1], Gaurav S Sukhatme[2], Shoudong Huang[3], Gamini Dissanayake[3] and Jonathan P How[1]

Abstract

Estimation-over-graphs (EoG) is a class of estimation problems that admit a natural graphical representation. Several key problems in robotics and sensor networks, including sensor network localization, synchronization over a group, and simultaneous localization and mapping (SLAM) fall into this category. We pursue two main goals in this work. First, we aim to characterize the impact of the graphical structure of SLAM and related problems on estimation reliability. We draw connections between several notions of graph connectivity and various properties of the underlying estimation problem. In particular, we establish results on the impact of the weighted number of spanning trees on the D-optimality criterion in 2D SLAM. These results enable agents to evaluate estimation reliability based only on the graphical representation of the EoG problem. We then use our findings and study the problem of designing sparse SLAM problems that lead to reliable maximum likelihood estimates through the synthesis of sparse graphs with the maximum weighted tree connectivity. Characterizing graphs with the maximum number of spanning trees is an open problem in general. To tackle this problem, we establish several new theoretical results, including the monotone log-submodularity of the weighted number of spanning trees. We exploit these structures and design a complementary greedy–convex pair of efficient approximation algorithms with provable guarantees. The proposed synthesis framework is applied to various forms of the measurement selection problem in resource-constrained SLAM. Our algorithms and theoretical findings are validated using random graphs, existing and new synthetic SLAM benchmarks, and publicly available real pose-graph SLAM datasets.

Keywords

SLAM, measurement selection, resource-constrained SLAM, estimation over graphs, pose-graph pruning, number of spanning trees, graph complexity

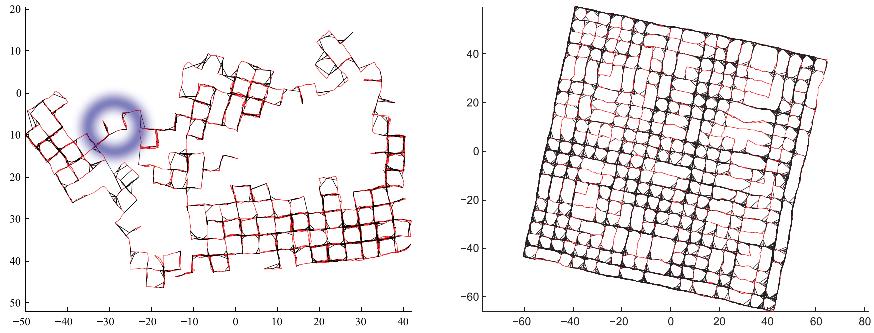
1. Introduction

Estimation-over-graphs (EoG) is a rich class of estimation problems that admit a natural graphical representation. In these problems, each vertex corresponds to an unknown state, and each edge corresponds to a noisy relative measurement between the corresponding states. Simultaneous localization and mapping (SLAM), synchronization problems over various groups, and sensor network localization are several notable examples of EoGs with important applications in robotics and sensor networks. An instance of EoG is often characterized by a measurement model, a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and a set of noisy measurements $z : \mathcal{E} \to \mathbb{M}$. The objective then is to find an optimal *drawing* of graph $\hat{} : \mathcal{V} \to \mathbb{X}$ in the state space $\mathbb{X}$, consistent with the given measurement model and measurements (e.g., in the maximum likelihood (ML) sense). For example, in d-dimensional pose-graph SLAM ($d \in \{2, 3\}$), both $\mathbb{X}$ and $\mathbb{M}$ correspond to the special Euclidean group SE(d). In this work, we explore and exploit the overlooked interplay between the graphical and estimation-theoretic facets of SLAM and several other EoGs.

The graphical representation of EoGs provides a compact overview of the underlying estimation problem. In particular, *graph connectivity* is related to the notion of *redundancy* in estimation. Intuitively, an EoG with a well-connected topology is expected to be more *resilient* to a fixed level of noise. A simple example is illustrated in Figure 1. This figure shows the ML estimates (MLEs) for two popular synthetic pose-graph SLAM datasets whose ground truths resemble grids. In this example, these

[1] Laboratory for Information and Decision Systems (LIDS), Massachusetts Institute of Technology, Cambridge, MA, USA
[2] Department of Computer Science Viterbi School of Engineering, University of Southern California, Los Angeles, CA, USA
[3] Centre for Autonomous Systems (CAS), University of Technology Sydney, Sydney, Australia

Corresponding author:
Kasra Khosoussi, Laboratory for Information and Decision Systems (LIDS), Massachusetts Institute of Technology, Cambridge, MA 02139, USA.
Email: kasra@mit.edu



(a) Manhattan (Olson, 2008) (b) City10K (Kaess et al., 2008)

Fig. 1. MLEs for two datasets with identical noise models. The bridges highlighted in the Manhattan dataset are indicative of its weak connectivity.

datasets are regenerated with identical noise regimes. Estimation errors and confidence ellipses in City10K (left) are notably smaller than those in Manhattan (right). This difference can be attributed mainly to the better connectivity of the City10K dataset.

Any well-defined graph connectivity measure must be monotone (i.e., non-decreasing) in the edge set. Therefore, graph connectivity can be maximized by simply adding all available edges to the graph (or, equivalently, collecting all possible measurements). Nevertheless, adding edges places a computational burden on the so-called *back-end* which is responsible for solving the underlying optimization problem. Thus, arbitrarily selecting new edges is not sustainable and prioritization based on *information content* is inevitable. This is especially important in the case of resource-constrained platforms, where there is often a hard limit on the number of measurements that can be collected, processed, and utilized (see, e.g., Carlone and Karaman, 2017; Davison, 2005; Giamou et al., 2018; Huang et al., 2013; Paull et al., 2015; Tian et al., 2018).

This paper aims to answer the following questions.

1. *Analysis*: How precisely does the graph topology influence the underlying estimation problem?
2. *Synthesis*: For a suitable connectivity measure that characterizes a certain aspect of estimation reliability, how can one design sparse, yet well-connected topologies that lead to a tractable (i.e., budget-feasible) and reliable estimation?

Answering the first question enables us to characterize desirable graph topologies for an EoG problem such as SLAM from an *intrinsic perspective* and independent of any geometric aspects of the problem (i.e., realized measurements or a graph drawing). Subsequently, in the second question we seek to design efficient algorithms for synthesizing such graphs under sparsity constraints as a proxy for the resource constraints arising in solving the underlying estimation problem.

In the SLAM literature, the impact of graph connectivity on estimation error is intuitively understood through the notion of loop closure "size." Closing a "larger" loop is preferred over closing "smaller" loops as it reduces more of the uncertainty in a robot's belief over its trajectory and the map (Bailey and Durrant-Whyte, 2006). We formalize this intuition by relating different measures of graph connectivity to several aspects of estimation reliability. Aside from presenting novel insights into an overlooked aspect of SLAM, our theoretical developments ultimately provide us with a unique set of tools for reasoning about estimation error based only on the topology of the underlying graph $\mathcal{G}$, i.e., without knowing the realized measurements, any particular drawing of the graph, or solving the inference problem. This is in contrast to the generic estimation-theoretic approach, in which the graphical facet of the EoG problem is neglected, and instead, the Fisher information matrix (FIM) is computed by evaluating the Jacobian at the current MLE of, e.g., a robot's trajectory. In comparison, the proposed graph-theoretic framework is robust to common convergence issues and linearization errors by virtue of not depending on a specific (estimated or nominal) trajectory, and demands less computational efforts by not requiring the solution of the resulting optimization problem and operating on a minimal representation of the problem. It is also more flexible when planning over longer horizons because it is independent of any particular metric realization of (current and/or future) trajectories and measurements.

The motivation behind our work originates from the work of Olson and Kaess (2009), where they insightfully highlighted the crucial role played by graph connectivity in SLAM. Olson and Kaess (2009) empirically observed that as the average degree in pose graphs increases, the value of the log-likelihood cost function at the MLE approaches its value at ground truth. We start our analysis by presenting a formal explanation for this observation. We then establish a connection between the graph Laplacian matrix and the FIM, which fully characterize, respectively, the graphical and estimation-theoretic facets of EoGs. This insight enables us to establish stronger links between the two aspects of SLAM based on the algebraic connectivity (Godsil and Royle, 2001) (Fiedler value) and the weighted number of spanning trees (WST). 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 (Godsil and Royle, 2001). As we shall see in this paper, tree connectivity is strongly tied to the standard D-optimality criterion (D-criterion for short), defined as the determinant of the (asymptotic) ML estimator covariance matrix. Consequently, one can accurately estimate the determinant of the estimation error covariance matrix using only the topology of the (weighted) graph underneath.

Based on the insight highlighted above, we address the second question, i.e., minimizing the determinant of the estimation error covariance in SLAM while maintaining computational tractability through the synthesis of sparse graphs with maximum tree connectivity. In particular, we tackle several forms of the measurement selection problem in which one seeks to select a D-optimal k-subset of loop closures. This problem has been studied extensively in the recent SLAM literature (see, e.g., Davison, 2005; Huang et al., 2013; Ila et al., 2010; Kaess and Dellaert, 2009). However, here we take a rather different graph-theoretic approach by exploiting the aforementioned connection between tree connectivity and D-optimality.

Graphs with the maximum 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., Cheng, 1981; Kelmans, 1996; Petingi and Rodriguez, 2002; Shier, 1974). The span of these special cases is too narrow for the types of graphs that typically arise in SLAM and related applications. Furthermore, the (n, m) constraint is usually insufficient for describing the true set of (physically) realizable graphs and cannot capture implicit physical and operational constraints that exist in these problems. Finally, it is not immediately clear how these results can be extended to the case of (edge) weighted graphs. The edge weights are essential parts of EoGs as they reflect the *precision* or *information* (i.e., inverse of variance) of the corresponding measurements. We address these challenges by proposing a pair of approximation algorithms with provable guarantees and near-optimality certificates.

In this paper, we extend the theoretical results presented in Khosoussi et al. (2014) and Khosoussi et al. (2016a) on the connection between the WST and the D-criterion. Moreover, this work extends the algorithms and theoretical results presented by Khosoussi et al. (2016b) to new formulations, and show how our approach can be used for D-optimal measurement selection in several scenarios. Finally, in this paper we provide an extensive experimental evaluation and validation based on real and synthetic datasets.

1.1. Outline

Section 2 provides a mathematical formulation of SLAM and several related EoG problems. We provide answers to the above-mentioned *analysis* question by linking different measures of graph connectivity to various aspects of estimation reliability in Section 3. Subsequently, in Section 4 we tackle the *synthesis* question and formally define the edge selection problem (ESP) for designing sparse graphs with the maximum weighted tree connectivity. In Section 5, we develop a pair of approximation algorithms for solving the synthesis problem. The proposed framework is extended to more general settings in Sections 6 and 7. In Section 8, we show how our graph-theoretic framework can be applied to various measurement selection problems in SLAM. In Section 9, we evaluate the performance of our algorithms on both synthetic graphs and real benchmark datasets. Section 10 concludes the paper. In addition, for the reader's convenience Appendix A provides a brief review of the terminology and a number of results from estimation theory, linear algebra, and spectral graph theory. Appendix B is where we present the proofs.

1.2. General 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 $\{\mathbf{e}_i\}_{i=1}^{n}$ where n is usually clear from the context. Sets are shown by upper-case letters. $|\mathcal{X}|$ denotes the cardinality of set $\mathcal{X}$. For any finite set $\mathcal{W}$, $\binom{\mathcal{W}}{k}$ is the set of all k-subsets of $\mathcal{W}$. We often use $[n]$ to denote the set $\{i \in \mathbb{N} : i \leqslant n\}$. The eigenvalues of symmetric matrix $\mathbf{M}$ are denoted by $\lambda_{\min}(\mathbf{M}) = \lambda_1(\mathbf{M}) \leqslant \cdots \leqslant \lambda_n(\mathbf{M}) = \lambda_{\max}(\mathbf{M})$. Here $\mathbf{1}$, $\mathbf{I}$ and $\mathbf{0}$ denote the vector of all ones, the identity, and the zero matrices with appropriate sizes, respectively. We use $\mathbf{S}_1 \succ \mathbf{S}_2$ (respectively $\mathbf{S}_1 \succeq \mathbf{S}_2$) to indicate that $\mathbf{S}_1 - \mathbf{S}_2$ is positive definite (respectively positive semidefinite). We use $\|\cdot\|$ to denote the Euclidean norm, and the weighted Euclidean norm of vector $\mathbf{e}$ with respect to matrix $\mathbf{W} \succ \mathbf{0}$ is denoted by $\|\mathbf{e}\|_{\mathbf{W}} \triangleq \sqrt{\mathbf{e}^\top \mathbf{W} \mathbf{e}}$. Here $\mathbb{S}^n_{>0}$ and $\mathbb{S}^n_{\geqslant 0}$ denote the sets of positive definite and positive semidefinite $n \times n$ matrices, respectively. The Kronecker product is denoted by $\otimes$. The block-diagonal matrix whose main diagonal blocks are $\mathbf{W}_1, \ldots, \mathbf{W}_k$ is denoted by $\operatorname{diag}(\mathbf{W}_1, \ldots, \mathbf{W}_k)$.

2. EoG problems

In this section, we mathematically formulate SLAM and two other classes of EoG problems. For each problem, we compute the FIM. It will soon become clear that the graph Laplacian is closely related to the FIM in these problems. The connection between the Fisher information and Laplacian matrices enables us to study the impact of graph topology on estimation in the following sections.

2.1. Synchronization over $\mathbb{R}^d$

Synchronization over $\mathbb{R}^d$ ($\mathbb{R}^d$-Sync) is one of the simplest classes of linear-Gaussian EoG problems. $\mathbb{R}^d$-Sync is a generalization of the time-synchronization problem in sensor networks (see Barooah and Hespanha, 2007). Let $\{\mathbf{x}_i\}_{i=1}^n$ represent the unknown states and $\{\mathbf{z}_i\}_{i=1}^m$ represent relative pairwise noisy measurements where both $\mathbf{x}_i$ and $\mathbf{z}_i$ belong to $\mathbb{R}^d$. In $\mathbb{R}^d$-Sync, the pairwise measurement between $\mathbf{x}_i$ and $\mathbf{x}_j$ is generated according to the following simple model,

$$\mathbf{z}_{ij} = \mathbf{x}_i - \mathbf{x}_j + \boldsymbol{\epsilon}_{ij}, \quad (i,j) \in \mathcal{E} \tag{1}$$

in which $\boldsymbol{\epsilon}_{ij}$ is a zero-mean Gaussian noise.

Assumption 1. Let $\boldsymbol{\epsilon}_i$ be the noise corrupting the ith measurement. We assume that (i) $\mathbb{C}\mathrm{ov}[\boldsymbol{\epsilon}_i, \boldsymbol{\epsilon}_j] = \mathbf{0}_{d\times d}$ for $i \neq j$ and (ii) $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \sigma_i^2\mathbf{I}_d)$.

$\mathbb{R}^d$-Sync with n variables can be naturally represented by a graph $\mathcal{G} = ([n], \mathcal{E})$, in which (i) variable $\mathbf{x}_i$ is represented by the vertex $i \in [n]$, (ii) relative measurement $\mathbf{z}_{ij}$ is represented by the edge $\{i,j\} \in \mathcal{E}$, and (iii) edges are weighted by the precision (inverse of variance) of the corresponding measurements, i.e., $w : \mathcal{E} \to \mathbb{R}_{>0} : e_k \mapsto \sigma_k^{-2}$. Owing to the relative nature of measurements, ML estimation in $\mathbb{R}^d$-Sync is ill-posed unless we anchor a vertex. Let $\mathbf{A}$ be the reduced incidence matrix of $\mathcal{G}$ (see Appendix A.1). Furthermore, let $\mathbf{x}$ and $\mathbf{z}$ denote the stacked vectors of states after anchoring an arbitrary vertex and measurements, respectively. The measurement model can be written as

$$\mathbf{z} = (\mathbf{A} \otimes \mathbf{I}_d)^{\mathrm{T}}\mathbf{x} + \boldsymbol{\epsilon} \tag{2}$$

in which $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ is the stacked vector of measurement noise. It is easy to verify that $\boldsymbol{\Sigma}^{-1} = \mathbf{W} \otimes \mathbf{I}_d$ where

$$\mathbf{W} \triangleq \mathrm{diag}(w(e_1), \ldots, w(e_m)) \tag{3}$$

Proposition 1. (Barooah and Hespanha, 2007). *The FIM in $\mathbb{R}^d$-Sync is given by $\mathbb{I} = \mathbf{L}_w \otimes \mathbf{I}_d$ in which $\mathbf{L}_w$ is the reduced weighted Laplacian matrix of $\mathcal{G}$.*

Remark 1. The ML estimator in $\mathbb{R}^d$-Sync, as a linear-Gaussian estimation problem, is unbiased and *efficient* (i.e., achieves the Cramér–Rao lower bound (CRLB)).

2.2. Compass-SLAM

Compass-SLAM is a simplified SLAM problem in which the robot orientation is assumed to be known, e.g., using a "compass". Duckett et al. (2000, 2002) proposed one of the early SLAM algorithms based on this model. The goal in Compass-SLAM is to estimate the robot and potentially landmarks' positions in $\mathbb{R}^d$ ($d \in \{2,3\}$), $\{\mathbf{p}_i\}_{i=1}^n$ using noisy translational measurements $\{\mathbf{z}_i\}_{i=1}^m$. This simplification reduces the ML estimation in SLAM to a linear-Gaussian estimation problem, whose globally-optimal solution can be computed easily by solving a linear least squares problem.

Remark 2. The ML estimator in Compass-SLAM, as a linear-Gaussian estimation problem, is unbiased and efficient.

The underlying structure of Compass-SLAM can be represented by a graph similar to $\mathbb{R}^d$-Sync. Let $\mathbf{R}$ be a block-diagonal matrix, consisting of $\{\mathbf{R}_i\}_{i=1}^m$ in which $\mathbf{R}_i \in \mathrm{SO}(d)$ is the rotation matrix corresponding to the robot orientation making the ith observation, i.e.,

$$\mathbf{R} \triangleq \mathrm{diag}(\mathbf{R}_1, \ldots, \mathbf{R}_m) \tag{4}$$

Let $\mathbf{p}$ and $\mathbf{z}$ be the stacked vector of robot and potentially landmarks' positions and translational measurements. After anchoring an arbitrary vertex, the measurement model in Compass-SLAM can be expressed as

$$\mathbf{z} = \mathbf{R}^{\mathrm{T}}(\mathbf{A} \otimes \mathbf{I}_d)^{\mathrm{T}}\mathbf{p} + \boldsymbol{\epsilon} \tag{5}$$

in which $\mathbf{A}$ is the reduced incidence matrix of $\mathcal{G}$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ is the measurement noise.

Assumption 2. Let $\boldsymbol{\epsilon}_i$ be the noise affecting the ith measurement. We assume that (i) $\mathbb{C}\mathrm{ov}[\boldsymbol{\epsilon}_i, \boldsymbol{\epsilon}_j] = \mathbf{0}_{d\times d}$ for $i \neq j$ and (ii) $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \sigma_i^2\mathbf{I}_d)$.

Proposition 2. *The FIM in Compass-SLAM is given by $\mathbb{I} = \mathbf{L}_w \otimes \mathbf{I}_d$ in which $\mathbf{L}_w$ is the reduced weighted Laplacian matrix of $\mathcal{G}$.*

2.3. SLAM

This section mainly concerns the 2D pose-graph SLAM problem with *relative-pose* measurements.[1] The state vector is $\mathbf{x} \triangleq \left[\mathbf{p}^\top \theta^\top\right]^\top$, and $\mathbf{z} = [\mathbf{z}_p^\top\ \mathbf{z}_\theta^\top]^\top$ is the stacked vector of translational and rotational measurements. The measurement model can be expressed as

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \boldsymbol{\epsilon} \tag{6}$$

in which $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. The measurement function $\mathbf{h}$, after computing the correct regularization terms for the rotational component of measurements (Carlone, 2013; Carlone et al., 2014; Carlone and Censi, 2014), is given by

$$\mathbf{h}(\mathbf{x}) \triangleq \begin{bmatrix} \mathbf{h}_p(\mathbf{x}) \\ \mathbf{h}_\theta(\theta) \end{bmatrix} = \begin{bmatrix} \mathbf{R}^\top(\mathbf{A} \otimes \mathbf{I}_2)^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^\top \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \theta \end{bmatrix} \tag{7}$$

We make the following assumption regarding the structure of the noise covariance matrix.

Assumption 3. We assume the rotational and translational measurements in SLAM are corrupted by block-isotropic noise, i.e., the noise covariance matrix $\boldsymbol{\Sigma}$ can be written as $\boldsymbol{\Sigma} = \mathrm{diag}(\boldsymbol{\Sigma}_p, \boldsymbol{\Sigma}_\theta)$, where

$$\boldsymbol{\Sigma}_p = \mathrm{diag}(\sigma_{p_1}^2\mathbf{I}_2, \ldots, \sigma_{p_m}^2\mathbf{I}_2) \tag{8}$$

$$\boldsymbol{\Sigma}_\theta = \mathrm{diag}(\sigma_{\theta_1}^2, \ldots, \sigma_{\theta_m}^2) \tag{9}$$

According to (78), for $p(\mathbf{z}; \mathbf{x}) = \mathcal{N}(\mathbf{z}; \mathbf{h}(\mathbf{x}), \boldsymbol{\Sigma})$ the FIM is given by

$$\mathbb{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{J}(\mathbf{x}) \tag{10}$$

where $\mathbf{J}(\mathbf{x}) \triangleq \partial \mathbf{h}(\mathbf{x})/\partial \mathbf{x}$. The Jacobian matrix $\mathbf{J}(\mathbf{x})$ can be easily computed:

$$\mathbf{J}(\mathbf{x}) \triangleq \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{J}_p^p & \mathbf{J}_\theta^p \\ \mathbf{J}_p^\theta & \mathbf{J}_\theta^\theta \end{bmatrix} \tag{11}$$

$$\mathbf{J}_p^p \triangleq \frac{\partial \mathbf{h}_p}{\partial \mathbf{p}} = \mathbf{R}^\top (\mathbf{A} \otimes \mathbf{I}_2)^\top \tag{12}$$

$$\mathbf{J}_\theta^p \triangleq \frac{\partial \mathbf{h}_p}{\partial \theta} = \widetilde{\mathbf{R}} \boldsymbol{\Delta} \tag{13}$$

$$\mathbf{J}_p^\theta \triangleq \frac{\partial \mathbf{h}_\theta}{\partial \mathbf{p}} = \mathbf{0} \tag{14}$$

$$\mathbf{J}_\theta^\theta \triangleq \frac{\partial \mathbf{h}_\theta}{\partial \theta} = \mathbf{A}^\top \tag{15}$$

Here $\widetilde{\mathbf{R}}$ is defined as $\widetilde{\mathbf{R}} \triangleq \boldsymbol{\Gamma} \mathbf{R}^\top$ in which $\boldsymbol{\Gamma}$ is given by

$$\boldsymbol{\Gamma} \triangleq \mathbf{I}_m \otimes \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{16}$$

It is easy to verify that $\boldsymbol{\Gamma}^\top \boldsymbol{\Gamma} = \mathbf{R}^\top \mathbf{R} = \widetilde{\mathbf{R}}^\top \widetilde{\mathbf{R}} = \mathbf{I}$; in fact, $\boldsymbol{\Gamma} \in \mathrm{SO}(2m)$. The other new term, $\boldsymbol{\Delta} \in \mathbb{R}^{2m \times n}$, has the following structure: for each $e_k \in \mathcal{E}$, $(\boldsymbol{\Delta})_{2k-1:2k,\, i_k} = \mathbf{p}_{j_k} - \mathbf{p}_{i_k}$ where i_k is the index of the node observing the j_kth node in the kth measurement. Other entries in $\boldsymbol{\Delta}$ are all zero. As noted by Carlone (2013), $\mathbf{D} \triangleq \boldsymbol{\Delta}^\top \boldsymbol{\Delta}$ is a diagonal matrix with the following structure: $\mathbf{D}_{i,i}$ is equal to the sum of squared distances between the ith robot pose and every node observed by it:

$$\mathbf{D}_{i,i} = \sum_{j \in \mathcal{N}_{\text{out}}(i)} \| \mathbf{p}_i - \mathbf{p}_j \|^2 \tag{17}$$

where $\mathcal{N}_{\text{out}}(v)$ is the set of nodes observed by $v \in \mathcal{V}$. Now we can compute the FIM $\mathbb{I}(\mathbf{x})$:

$$\mathbb{I}(\mathbf{x}) = \begin{bmatrix} \mathbf{L}_{w_p} \otimes \mathbf{I}_2 & (\mathbf{A}_{w_p} \otimes \mathbf{I}_2) \boldsymbol{\Gamma} \boldsymbol{\Delta}_{w_p} \\ *^\top & \mathbf{L}_{w_\theta} + \boldsymbol{\Delta}_{w_p}^\top \boldsymbol{\Delta}_{w_p} \end{bmatrix} \tag{18}$$

in which $*$ denotes the top-right block; $\mathbf{L}_{w_p}$ and $\mathbf{L}_{w_\theta}$ are the reduced weighted Laplacian matrices of $\mathcal{G}$ when edges are weighted according to $w_p : e_k \mapsto \sigma_{p_k}^{-2}$ and $w_\theta : e_k \mapsto \sigma_{\theta_k}^{-2}$, respectively; $\mathbf{W}_p$ is the diagonal matrix of edge weights based on w_p; $\mathbf{A}_{w_p} \triangleq \mathbf{A}\sqrt{\mathbf{W}_p}$ is the reduced weighted incidence matrix of $\mathcal{G}$ when edges are weighted by w_p defined above; $\boldsymbol{\Delta}_{w_p} \triangleq \sqrt{\mathbf{W}_p}\boldsymbol{\Delta}$; and $\mathbf{D}_{w_p} \triangleq \boldsymbol{\Delta}_{w_p}^\top \boldsymbol{\Delta}_{w_p}$ is a diagonal matrix with the following structure:

$$\mathbf{D}_{w_p i,i} = \sum_{j \in \mathcal{N}_{\text{out}}(i)} w_p(i,j) \| \mathbf{p}_i - \mathbf{p}_j \|^2 \tag{19}$$

3. Topology and estimation reliability

3.1. Average degree

We now demonstrate that the ratio between the value of the log-likelihood cost function evaluated at the true value and the MLE can be accurately approximated by a simple function of the underlying graph structure. To simplify our notation, let $\mathbf{x}^\star$ and $\mathbf{x}^\circ$ be the MLE and the true value of $\mathbf{x}$, respectively. Olson and Kaess (2009) investigated the ratio between the minimum of the log-likelihood objective function $f^\star \triangleq f(\mathbf{x}^\star)$, and its value at the true $\mathbf{x}$, $f^\circ \triangleq f(\mathbf{x}^\circ)$. Figure 1 illustrates these values in a toy example. Define $\gamma \triangleq f^\star / f^\circ$. It is easy to see that (i) $0 \leqslant \gamma \leqslant 1$ and (ii) $\gamma \to 1^-$ as $\mathbf{x}^\star \to \mathbf{x}^\circ$. Therefore, $\gamma \approx 1$ indicates that the MLE is close to the ground truth in objective value. Using Monte Carlo simulations, Olson and Kaess (2009) empirically observed that as the average node degree of the graph, i.e., $\bar{d} \triangleq \frac{1}{n}\sum_{i=1}^{n} \deg(i)$ increases, γ on average approaches 1 (see Olson and Kaess, 2009: Figure 5). According to their interpretation, γ is a "coarse measure of *overfitting*" (Olson and Kaess, 2009). We repeated their experiment and observed the same behavior. The blue points in Figure 2 correspond to the average of γ in a series of Monte Carlo simulations over a large number of randomly generated SLAM problems with different average degrees (similar to Figure 5 in Olson and Kaess, 2009). For each random pose-graph, we generated 50 independent and identically distributed realizations of measurement noise. We then computed γ for each realization of noise, and averaged it over the 50 Monte Carlo simulations. In what follows we provide a theoretical explanation for this empirical observation.

Assumption 4 (Additive Gaussian noise). We assume that measurements are corrupted by an additive Gaussian noise, i.e., $\mathbf{z} = \mathbf{h}(\mathbf{x}^\circ) + \boldsymbol{\epsilon}$ in which $\mathbf{z}$ is the measurement, $\mathbf{h}$ is the measurement function and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ is the noise.[2]

The negative log-likelihood cost function for the model specified in Assumption 4 can be written as $f(\mathbf{x}) = \| \mathbf{z} - \mathbf{h}(\mathbf{x}) \|_{\boldsymbol{\Sigma}^{-1}}^2$. To compute γ, we need to compute both $f^\star$ and f°. According to the definition, f° is the value of the negative log-likelihood at $\mathbf{x} = \mathbf{x}^\circ$. Note that $f^\circ \triangleq f(\mathbf{x}^\circ)$ is a random variable as it depends on $\mathbf{z}$, and

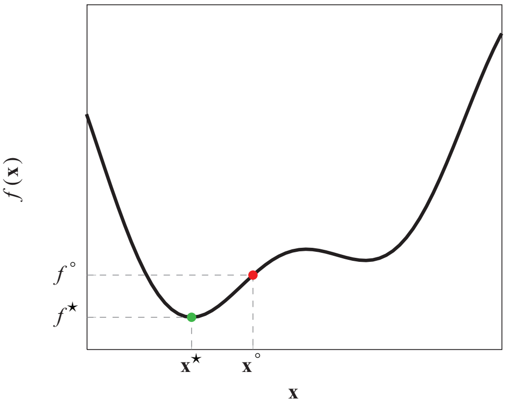


Fig. 2. Here f (drawn in black) is a pictorial representation of the negative log-likelihood objective function. The MLE $\mathbf{x}^\star$ and the ground truth $\mathbf{x}^\circ$, together with their objective values, are specified.

consequently ϵ. The following proposition gives the distribution of f°.

Proposition 3. *Under Assumption 4, $f^\circ \sim \chi^2_{\nu^\circ}$ in which $\nu^\circ = \dim(\mathbf{z})$.*

According to Proposition 3, f° follows a $\chi^2_{\nu^\circ}$ distribution with $\dim(\mathbf{z})$ degrees of freedom. Now let us examine the behavior of $f^\star \triangleq f(\mathbf{x}^\star)$. Note that $\mathbf{x}^\star$ depends on $\mathbf{z}$ and therefore is a random variable. Consequently, $f^\star$, as a function of $\mathbf{x}^\star$ and $\mathbf{z}$, is also a random variable. The following well-known result provides the distribution of $f^\star$ when the measurement function is affine in $\mathbf{x}$.

Proposition 4. *Under Assumption 4 and for linear measurement models we have $f^\star \sim \chi^2_{\nu^\star}$ in which $\nu^\star = \dim(\mathbf{z}) - \dim(\mathbf{x})$.*

Proposition 4 characterizes the distribution of $f^\star$ for affine measurement functions corrupted by additive Gaussian noise (Assumption 4). It is impossible to analytically characterize the distribution of $f^\star$ for general (nonlinear) measurement functions. However, we can always linearize sufficiently smooth nonlinear measurement functions using the first-order Taylor expansion around $\mathbf{x} = \mathbf{x}^\circ$. Given a sufficiently small linearization error, one can use Proposition 4 to approximate the distribution of $f^\star$. According to Proposition 3 and Proposition 4, the distribution of f° and $f^\star$ is determined by the size of $\mathbf{z}$ and $\mathbf{x}$. For example, in 2D pose graphs we have $\nu^\circ = 3m$ and $\nu^\star = 3(m-n)$. $\mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0},\boldsymbol{\Sigma})}[\gamma]$ can be roughly approximated by

$$\mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0},\boldsymbol{\Sigma})}[\gamma] \approx \frac{\mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0},\boldsymbol{\Sigma})}[f^\star]}{\mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0},\boldsymbol{\Sigma})}[f^\circ]} \tag{20}$$

$$\approx \frac{\nu^\star}{\nu^\circ} \tag{21}$$

$$= 1 - \frac{n}{m} \tag{22}$$

The rough approximation made in (20) can be justified by the first-order Taylor expansion of $\gamma = f^\star/f^\circ$ at $f^\star = \mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0},\boldsymbol{\Sigma})}[f^\star] \approx \nu^\star$ and $f^\circ = \mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0},\boldsymbol{\Sigma})}[f^\circ] = \nu^\circ$; this naïve approximation is introduced only to reproduce the aforementioned empirical observations (Olson and Kaess, 2009), as we will see shortly in Figure 3.

Using the handshaking lemma, we can express (22) in terms of the average degree of the graph, i.e.,

$$\mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0},\boldsymbol{\Sigma})}[\gamma] \approx 1 - 2/\bar{d} \tag{23}$$

Equation (23) reveals the relation between the expected value of γ, as a measure of estimation accuracy, and the average node degree $\bar{d}$, as the *simplest* measure of graph connectivity. According to (23), as the average degree increases, γ in expectation approaches 1, which indicates an accurate MLE. The red points in Figure 3 are drawn according to (23). According to this figure, (23) is consistent with our and Olson and Kaess's (2009) empirical observations regarding the average value of γ (blue points in Figure 3).

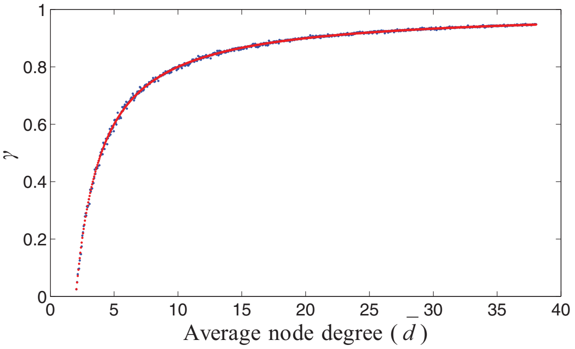


Fig. 3. Average of γ in 50 Monte Carlo simulations for different average node degrees (blue). Estimated value of $\mathbb{E}[\gamma] \approx 1 - \frac{2}{\bar{d}}$ (red).

The average degree is proportional to the ratio between the number of (vector-valued) measurements (e.g., odometry and loop-closure measurements) and the number of (vector-valued) variables (e.g., robot poses). Therefore, for a fixed number of poses, maximizing the average degree is equivalent to merely maximizing the number of measurements. The average degree, as a connectivity measure, is insensitive to topological differences between graphs with the same number of edges (per vertices). Hence, although (23) is consistent with the empirical results, $\bar{d}$ is not sophisticated enough to capture and reflect the differences between graph structures with the same number of measurements. Nonetheless, our analysis provides an insight into how making new observations ultimately leads to a more precise estimate in terms of the value of the log-likelihood function.

3.2. Algebraic Connectivity

Now we present another result on how in $\mathbb{R}^d$-Sync and Compass-SLAM, a "well-connected" graph is necessary for achieving reliable estimates. The following remark sets the stage for our main result.

Remark 3. Let $\mathbb{C}\mathrm{ov}[\mathbf{e}] \succ \mathbf{0}$ be a $n \times n$ estimation error covariance matrix in which $\mathbf{e} \in \mathbb{R}^n$ denotes the estimation error. Let $\lambda_{\max} \triangleq \lambda_n(\mathbb{C}\mathrm{ov}[\mathbf{e}])$ be the largest eigenvalue of $\mathbb{C}\mathrm{ov}[\mathbf{e}]$. The following statements hold regarding $\lambda_{\max}$:

1. $\lambda_{\max}$ specifies the worst-case variance among all unit directions (Joshi and Boyd, 2009);
2. geometrically speaking, if $\mathbb{C}\mathrm{ov}[\mathbf{e}]$ is the covariance matrix of a Gaussian distribution, the "hyperdiameter" of the uncertainty hyperellipsoid is determined by $\sqrt{\lambda_{\max}}$.

In the optimal experimental design literature, the design that minimizes the largest eigenvalue of the estimation error covariance matrix is known as the E-optimal (eigenvalue-optimal) design (Pukelsheim, 1993). For any simple connected undirected graph $\mathcal{G}$, the second-largest eigenvalue of the (weighted) Laplacian matrix $\lambda_2(\mathbf{L}_\circ)>0$ is known as the *algebraic connectivity* (or Fiedler value) of $\mathcal{G}$; see Appendix A.1. A closely related quantity is the smallest eigenvalue of the *reduced Laplacian* $\lambda_1(\mathbf{L})$ which also reflects graph connectivity as shown by the following proposition as shown by the following proposition.

Proposition 5. *Let* $\mathbf{L}$ *be the reduced Laplacian matrix of simple undirected graph* $\mathcal{G}$. *The following statements hold*:

1. $0 \leqslant \lambda_{\min}(\mathbf{L}) \leqslant 1$;
2. $\lambda_{\min}(\mathbf{L}) = 0$ *iff* $\mathcal{G}$ *is disconnected and* $\lambda_{\min}(\mathbf{L}) = 1$ *is maximum iff* $\mathcal{G}$ is complete;
3. $\lambda_{\min}(\mathbf{L})$ *is monotone in the edge set of the graph; i.e., adding a new edge to* $\mathcal{G}$ *cannot decrease* $\lambda_{\min}(\mathbf{L})$;
4. *(Pirani and Sundaram, 2014): let* $w : \mathcal{E}(\mathcal{G}) \to \mathbb{R}_{>0}$ *be a weight function and* $\mathbf{L}_w$ *be the reduced Laplacian of* $\mathcal{G}$ *after anchoring* v_0 *with edge weights given by* w; *it holds that* $\lambda_{\min}(\mathbf{L}_w) \leqslant w^{\ddagger}_{\max}$, where $w^{\ddagger}_{\max} \triangleq \max_{u \sim v_0} w(v_0, u)$.

The following theorem sheds light on the connection between the algebraic connectivity, the smallest eigenvalue of the reduced Laplacian of the graph, and the worst-case estimation error variance in $\mathbb{R}^d$-Sync and Compass-SLAM.

Theorem 1 (Algebraic connectivity and worst-case error variance). *Let* $\mathbb{C}\mathrm{ov}[\mathbf{x}^\star]$ *be the estimation error covariance matrix of the ML estimator in* $\mathbb{R}^d$*-Sync and Compass-SLAM. Let* $\mathbf{L}_{\circ_w}$ *be the corresponding weighted Laplacian matrix. Then we have* $\lambda_{\max}(\mathbb{C}\mathrm{ov}[\mathbf{x}^\star]) = \lambda_1^{-1}(\mathbf{L}_w)$. *Furthermore, it holds*:

$$\lambda_{\max}(\mathbb{C}\mathrm{ov}[\mathbf{x}^\star]) \geqslant 1/\lambda_2(\mathbf{L}_{\circ_w}) \tag{24}$$

$$\lambda_{\max}(\mathbb{C}\mathrm{ov}[\mathbf{x}^\star]) \geqslant 1/w^{\ddagger}_{\max} \tag{25}$$

Theorem 1 states that the worst-case estimation error variance in $\mathbb{R}^d$-Sync and Compass-SLAM can be expressed in terms of graph connectivity as captured by $\lambda_1(\mathbf{L}_w)$. It also shows that reducing the worst-case estimation error variance in these EoG problems requires sufficiently strong algebraic connectivity and a sufficiently large $w^{\ddagger}_{\max}$ (i.e., a high-precision measurement to the anchored node). According to Pirani and Sundaram (2014), (25) provides a tighter bound than (24). Needless to say, scaling the information content (precision) of every edge scales $\lambda_{\max}(\mathbb{C}\mathrm{ov}[\mathbf{x}^\star])$, $\lambda_1(\mathbf{L}_w)$, and $\lambda_2(\mathbf{L}_{\circ_w})$ by the same amount.

3.3. Tree connectivity

In what follows we demonstrate that the weighted number of spanning trees, as a measure of graph connectivity (see Appendix A.1), has a significant impact on the determinant of the estimation error covariance of the ML estimator in several EoG problems, including SLAM. The D-criterion provides a scalar measure of the uncertainty encoded in a covariance matrix. In particular, the square root of the determinant of the covariance matrix is proportional to the hypervolume of the confidence hyperellipsoids in multivariate Gaussian distributions (see Joshi and Boyd, 2009). Furthermore, from an information-theoretic standpoint, the log-determinant of the covariance matrix of a multivariate Gaussian random variable is proportional to its differential entropy up to an additive constant.

Before presenting our main results, we point out several standard numerical tricks that are crucial for handling large problems. First, note that minimizing $\det \mathbb{C}\mathrm{ov}[\mathbf{x}^\star]$ is equivalent to minimizing $\log\det \mathbb{C}\mathrm{ov}[\mathbf{x}^\star]$; however, we directly compute the latter in order to avoid overflow and underflow. Similarly, rather than working with the weighted number of spanning trees $t_w(\mathcal{G}) = \det \mathbf{L}_w$ (Theorem 12), we often use the *weighted tree connectivity* as defined in the following.

Definition 1 (Tree connectivity). The *tree connectivity* of graph $\mathcal{G}$ is formally defined as

$$\tau(\mathcal{G}) \triangleq \begin{cases} \log t(\mathcal{G}) & \text{if } \mathcal{G} \text{ isconnected} \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

Similarly, for graphs weighted by a positive weight function $w : \mathcal{E}(\mathcal{G}) \to \mathbb{R}_{>0}$ (sometimes, for convenience and without loss of generality, $w : \mathcal{E}(\mathcal{G}) \to \mathbb{R}_{\geqslant 1}$), the *weighted tree connectivity* is defined as

$$\tau_w(\mathcal{G}) \triangleq \begin{cases} \log t_w(\mathcal{G}) & \text{if } \mathcal{G} \text{ isconnected} \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

In SLAM and many other real-world EoGs, the Laplacian and the Fisher information matrices are sparse. Algorithm 1 outlines a standard procedure for efficiently computing the log-determinant of sparse positive-definite matrices. Unlike the Fisher information matrix, the covariance matrix in SLAM is generally dense. Therefore, in practice $\log\det \mathbb{C}\mathrm{ov}[\mathbf{x}^\star]$ is computed indirectly via $\log\det \mathbb{I}(\mathbf{x})$:

$$\log\det \mathbb{C}\mathrm{ov}[\mathbf{x}^\star] \approx \log\det \mathbb{I}(\mathbf{x}^\star)^{-1} \tag{28}$$

$$= -\log\det \mathbb{I}(\mathbf{x}^\star) \tag{29}$$

In the worst case of dense matrices, log-determinant can be computed in $O(n^3)$ time where n is the number of poses.

Algorithm 1. $\log\det(\mathbf{S})$ for a sparse $\mathbf{S} \succ \mathbf{0}$

```
function LogDet($\mathbf{S}$)
    // Choose a fill-reducing permutation heuristic $\mathbf{P}$
    $\mathbf{P} \leftarrow$ COLAMD($\mathbf{S}$)    ▷ e.g., column approximate minimum degree
    // Sparse Cholesky factor $\mathbf{C}$ s.t. $\mathbf{S} = \mathbf{C}\mathbf{C}^\top$
    $\mathbf{C} \leftarrow$ SparseCholesky($\mathbf{P}\mathbf{S}\mathbf{P}^\top$)
    return $2\sum_i \log \mathbf{C}_{i,i}$
end function
```

Nonetheless, in many practical sparse scenarios that arise in robotics, Algorithm 1 performs much faster given a sufficiently good fill-reducing permutation. Now we are ready to present our main results.

Theorem 2 (D-optimal $\mathbb{R}^d$-Sync and Compass-SLAM). *In $\mathbb{R}^d$-Sync and d-dimensional Compass-SLAM we have*

$$\log\det(\mathbb{C}\mathrm{ov}[\mathbf{x}^\star]) = -\,d\tau_w(\mathcal{G}) \tag{30}$$

in which the edge weights are given by the precision of the corresponding measurements (see Section 2).

Theorem 2 states that in $\mathbb{R}^d$-Sync and Compass-SLAM, under the specified assumptions, maximizing the WST in the underlying graph is equivalent to minimizing the hypervolume of uncertainty ellipsoids, and maximizing the D-criterion. Graphs with the maximum (weighted) number of spanning trees among a family of graphs are called *t*-optimal. Theorem 2 suggests that D-optimality and *t*-optimality are indeed equivalent under the above-mentioned assumptions. Now we extend Theorem 2 to SLAM.

Theorem 3. *Consider the SLAM problem (Section 2.3) and let n be the number of poses, $\delta \triangleq \| \boldsymbol{\Delta}_{w_p}^{\mathrm{T}}\boldsymbol{\Delta}_{w_p} \|_\infty$, and $\lambda_1 \triangleq \lambda_{\min}(\mathbf{L}_{w_\theta})$. Also define $\ell_\tau(\mathcal{G}) \triangleq 2\tau_{w_p}(\mathcal{G}) + \tau_{w_\theta}(\mathcal{G})$ and $\epsilon(\mathbf{x}) \triangleq \log\det\mathbb{I}(\mathbf{x}) - \ell_\tau(\mathcal{G})$. Then we have*

$$0 \leqslant \epsilon(\mathbf{x}) \leqslant n\log(1 + \delta/\lambda_1) \tag{31}$$

Note that $\ell_\tau(\mathcal{G}) \triangleq 2\tau_{w_p}(\mathcal{G}) + \tau_{w_\theta}(\mathcal{G})$ only depends on the weighted tree connectivity of the underlying graph. The above theorem gives lower and upper bounds for the gap $\epsilon(\mathbf{x})$ between $\ell_\tau(\mathcal{G})$ and the actual D-criterion. More specifically, Theorem 3 states that $\ell_\tau(\mathcal{G})$ is a universal graphical lower bound on the D-criterion for *any* trajectory $\mathbf{x}$ or particular realization of measurements $\mathbf{z}$. Therefore, $\ell_\tau(\mathcal{G})$ can be used as a special graphical surrogate function for maximizing the D-criterion in SLAM whenever, e.g., a reliable estimate of the geometry $\mathbf{x}$ is not available (we discuss a number of examples in the second part of this paper). Theorem 3 also shows that the gap between the D-criterion and its graphical lower bound $\ell_\tau(\mathcal{G})$ is bounded. Now let us focus on this upper bound. From (19) we know that δ is the maximum weighted sum of squared distances between a pose and any other pose observed by it, i.e.,

$$\delta = \max_{i\in[n]} \left\{ \sum_{j\in\mathcal{N}_{\mathrm{out}}(i)} w_p(i,j) \,\|\, \mathbf{p}_j - \mathbf{p}_i \|^2 \right\} \tag{32}$$

Thus, δ depends on a number of factors such as sensor characteristics (e.g., sensing range and field of view), the distance between consecutive poses (odometry), and the precision of translational measurements w_p. The other parameter that appears in Theorem 3 is the smallest eigenvalue of $\mathbf{L}_{w_\theta}$. From Proposition 5 recall that λ_1 is (i) positive iff the graph is connected (which is the case in SLAM), (ii) monotone in the edge set, and (iii) bounded by the maximum precision among all *rotational* measurements connected to the anchored node. In addition, scaling the rotational weights w_θ scales λ_1 by the same factor. Corollary 1 follows directly from Theorem 3.

Corollary 1. *We have*

$$\lim_{\delta/\lambda_1\to 0^+} \log\det\mathbb{I}(\mathbf{x}) = 2\tau_{w_p}(\mathcal{G}) + \tau_{w_\theta}(\mathcal{G}) \tag{33}$$

This corollary implies the *asymptotic tightness* of our graphical lower bound $\ell_\tau(\mathcal{G})$ on the D-criterion, i.e., the approximation gap $\epsilon(\mathbf{x})$ vanishes as δ/λ_1 approaches zero. SLAM problems may *approach* this asymptotic regime as discussed above.

In the rest of this section we explore the implications of Theorem 3 in the special case of isotropic rotational and translational measurement noise.

Corollary 2. *Suppose the covariance matrix for rotational and translational measurements is isotropic, i.e., $\boldsymbol{\Sigma}_p = \sigma_p^2\mathbf{I}$ and $\boldsymbol{\Sigma}_\theta = \sigma_\theta^2\mathbf{I}$. Let $\tilde{\lambda}_1$ be the smallest eigenvalue of the reduced Laplacian and $\tilde{\delta} \triangleq \| \Delta^{\mathrm{T}}\Delta \|_\infty$. Under the following condition, the gap is guaranteed to be $\epsilon(\mathbf{x}) \leqslant \alpha\cdot n$:*

$$\sigma_\theta^2/\sigma_p^2 \leqslant (\exp(\alpha) - 1)\tilde{\lambda}_1/\tilde{\delta} \tag{34}$$

$$\approx \alpha\tilde{\lambda}_1/\tilde{\delta}, \quad \textit{for} \quad \alpha \approx 0 \tag{35}$$

Corollary 3. *Suppose the covariance matrix for rotational and translational measurements is isotropic. Let $\mathbb{I}(\mathbf{x}|\mathcal{T})$ be the FIM associated to an arbitrary spanning tree, e.g., the odometry subgraph. Then,*

$$\lim_{\delta/\lambda_1\to 0^+} \log\det\mathbb{I}(\mathbf{x}) - \log\det\mathbb{I}(\mathbf{x}|\mathcal{T}) = 3\tau(\mathcal{G}) \tag{36}$$

A natural choice for $\mathcal{T}$ is the odometry spanning tree $\mathcal{T}_{\mathrm{odo}}$. In this case, $\Delta_{\mathrm{inf}}(\mathcal{G}) \triangleq \log\det\mathbb{I}(\mathbf{x}) - \log\det\mathbb{I}(\mathbf{x}|\mathcal{T}_{\mathrm{odo}})$ can be interpreted as the information gained by closing loops as compared with the dead-reckoning scenario. According to Corollary 3, $\Delta_{\mathrm{inf}}(\mathcal{G})$ will be proportional to the tree connectivity of $\mathcal{G}$ when δ/λ_1 is sufficiently small. The following corollary directly follows from Corollary 3 and Cayley's formula (see Theorem 11).

Corollary 4. *In the SLAM problem defined in Section 2.3 with isotropic noise we have*

$$\lim_{\delta/\lambda_1\to 0^+} \Delta_{\mathrm{inf}}(\mathcal{K}_n) = 3(n-2)\log(n) \tag{37}$$

Theorems 2 and 3 establish a basis for comparing the graphical structure of different instances of EoG problems based on their number of spanning trees. It is important to note that two graphs are comparable based on tree connectivity only if they have the same number of vertices. For a fair comparison of the tree connectivity of graphs with a different number of vertices, we need to somehow normalize the absolute tree connectivity by the graph size.

Definition 2 (Normalized tree connectivity). Suppose $\mathcal{G}$ is a graph with n vertices and $\mathcal{K}$ is the complete graph over n vertices. We define the *normalized tree connectivity* of $\mathcal{G}$, denoted by $\bar{\tau}(\mathcal{G})$, as $\bar{\tau}(\mathcal{G}) \triangleq \tau(\mathcal{G})/\tau(\mathcal{K})$.

According to this definition, the tree connectivity of each graph is normalized by the tree connectivity of the complete graph with the same number of vertices. In other words, to any simple connected graph $\mathcal{G}$, $\bar{\tau}(\mathcal{G})$ assigns a score that reflects the tree connectivity of $\mathcal{G}$ relative to the tree connectivity of the complete graph with the same number of vertices. The following corollary directly follows from the above definition and Cayley's formula (see Theorem 11).

Corollary 5. *Let $\mathcal{G}$ be a simple undirected graph with n vertices. The following statements hold regarding the normalized tree connectivity of $\mathcal{G}$:*

1. $\bar{\tau}(\mathcal{G}) = \frac{\tau(\mathcal{G})}{(n-2)\log(n)}$;
2. $0 \leqslant \bar{\tau}(\mathcal{G}) \leqslant 1$;
3. $\bar{\tau}(\mathcal{G}) = 0$ *if and only if* $\mathcal{G}$ *is acyclic*;
4. $\bar{\tau}(\mathcal{G}) = 1$ *if and only if* $\mathcal{G}$ *is the complete graph.*

Corollary 6 follows from Corollary 3.

Corollary 6. *In the SLAM problem defined in Section 2.3 with isotropic noise, we have*

$$\lim_{\delta/\lambda_1 \to 0^+} \frac{\Delta_{\text{inf}}(\mathcal{G})}{\Delta_{\text{inf}}(\mathcal{K})} = \bar{\tau}(\mathcal{G}) \tag{38}$$

Hence, under the assumption of isotropic noise, the normalized tree connectivity can be interpreted as the ratio of the information gained relative to dead reckoning, between the realized graph $\mathcal{G}$ and the complete graph $\mathcal{K}$. Finally, we note that a similar normalization scheme can be designed for weighted graphs. Such an extension, however, would require making explicit assumptions regarding the weights of the missing edges.

Remark 4. Typical SLAM problems are naturally sparse. Consequently, normalizing the tree connectivity using the complete graph with the same number of vertices can be misleading since in practice $\mathcal{K}$ is not achievable for a moving robot. As will become clear in Section 5, our graph synthesis frameworks provides us with lower and upper bounds for the maximum tree connectivity of sparse graphs based on a more realistic model. Those bounds can be readily used instead of $\tau(\mathcal{K})$.

3.4. Tree connectivity: experimental results

We conducted a series of experiments designed to:

1. evaluate the tree connectivity of some of the publicly available real and synthetic SLAM benchmarks;
2. empirically validate Theorem 3;
3. assess the sensitivity of the asymptotic result provided by Corollary 1 with respect to the value of δ.

Table 1. A list of publicly available 2D pose-graph datasets, sorted according to $\bar{\tau}(\mathcal{G})$.

Dataset	$\bar{\tau}(\mathcal{G})$	Average degree	RE (%)
M10K	0.22	12.86	0.07
Intel	0.13	3.89	0.06
City10K	0.12	4.13	0.51
Lincoln Lab	0.11	3.90	58.00
Manhattan	0.09	3.11	1.00
RingCity	0.05	2.76	1.08
Freiburg	0.04	2.46	0.04
CSAIL	0.02	2.24	0.12

To validate Theorem 3 and Corollary 1 numerically, we use the relative error (RE) defined as

$$\text{RE} \triangleq \left| \frac{\log\det \mathbb{I}(\mathbf{x}) - \ell_\tau(\mathcal{G})}{\log\det \mathbb{I}(\mathbf{x})} \right| = \frac{\epsilon(\mathbf{x})}{\log\det \mathbb{I}(\mathbf{x})} \tag{39}$$

where $\ell_\tau(\mathcal{G}) \triangleq 2\tau_{w_p}(\mathcal{G}) + \tau_{w_\theta}(\mathcal{G})$. The datasets used in our experiments are all 2D pose-graph SLAM datasets. The non-diagonal noise covariance matrices have been modified to satisfy our assumptions about the noise covariance matrix (e.g., block-isotropic for Theorem 3 and isotropic for Corollary 3). In some cases, we have also removed parallel edges for simplicity.[3] RE, through the FIM, depends on $\mathbf{x}$. Recall that the inverse of the FIM evaluated at the ground truth results in the CRLB. Moreover, the covariance matrix of the ML estimator is usually approximated by computing the inverse of the Fisher information matrix at the MLE. Among the datasets used in this section, Manhattan (Olson, 2008) (a synthetic 2D pose-graph dataset) is the only one for which the ground truth is publicly available. Therefore, RE in other datasets is evaluated at the solution obtained by minimizing the negative log-likelihood cost function using Gauss–Newton initialized by the popular bootstrapping technique proposed in Konolige et al. (2010).

The normalized tree connectivity for several publicly available datasets is shown in Table 1. The entries in Table 1 are sorted (in descending order) based on the normalized tree connectivity $\bar{\tau}(\mathcal{G})$. First, note that the RE is typically small, except in the case of the Lincoln Lab dataset. A small RE indicates that $\log\det \mathbb{I}(\mathbf{x})$ is already close to $\ell_\tau(\mathcal{G})$ as predicted by Theorem 3 and Corollary 1. This empirical observation hence suggests that in these datasets, the actual value of the gap $\epsilon(\mathbf{x})$ is small despite the large value of the upper bound provided by Theorem 3 (e.g., in Manhattan $\delta \approx 1296.91$). In such cases, the log-determinant of the FIM is almost entirely characterized by the tree connectivity of the underlying graph.

It is important to note that evaluating $\log\det \mathbb{I}(\mathbf{x})$ at the solution returned by iterative schemes such as Gauss–Newton is subject to local minima and other convergence failure modes. The large RE in the case of Lincoln Lab dataset (highlighted in red) is partially due to the fact that Gauss–Newton has failed to converge to the true MLE. By

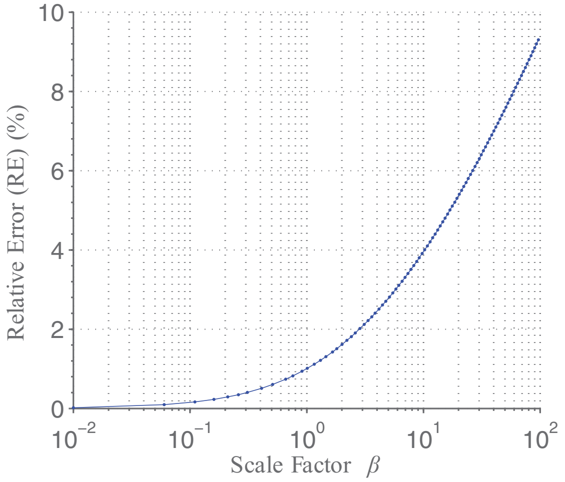


Fig. 4. Evolution of RE evaluated at the ground truth as a function of the scale parameter β for Manhattan. Here $\delta = \beta\delta_{\text{orig}}$ in which $\delta_{\text{orig}} \approx 1296.91$ is the value of δ in Manhattan dataset (see Table 1). This can be done by scaling either σ_p^2 or $\mathbf{p}$. Note the logarithmic scale of the horizontal axis.

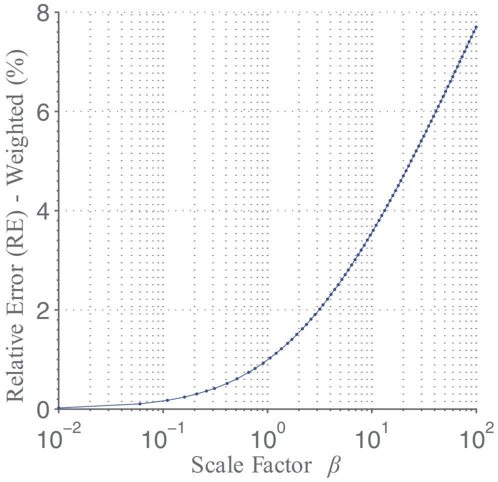


Fig. 5. Evolution of RE evaluated at the ground truth as a function of scale parameter β for Manhattan with block-isotropic noise. Here $\delta = \beta\delta_{\text{orig}}$ in which $\delta_{\text{orig}} \approx 6.21 \times 10^4$. In this experiment, the edges have different noise variances. Note the logarithmic scale of the horizontal axis.

contrast, approximating the D-criterion with the lower bound provided by Theorem 3 is naturally robust to such convergence errors.

Note that in Table 1, the ranking based on the number of spanning trees is not consistent with the ranking based on the average degree of the graph (see the entries highlighted in green). As mentioned earlier, this is because the average degree, as a graph connectivity measure (Olson and Kaess, 2009), is too simple to capture the topological differences between two graphs with the same number of measurements per variables.

Our next experiment is based on the Manhattan dataset. The FIM in this case is computed at the ground truth. In these experiments λ_1 is kept fixed. Figure 4 shows how RE evolves with respect to scaling δ when the noise is isotropic. Scaling δ can be done by scaling either σ_p^2 or $\mathbf{p}$. Figure 4 is obtained by scaling δ according to $\beta\delta_{\text{orig}}$ in which $\delta_{\text{orig}} \approx 1296.91$ is the original value of δ in the Manhattan dataset. As illustrated in Figure 4, the log-determinant of the Fisher information matrix approaches the limit value predicted by Corollary 1 as δ approaches zero (see also the special case in Corollary 3).

We also repeated this experiment for the case of block-isotropic noise. To make the original isotropic noise of Manhattan compatible with the assumption of block-isotropic noise, we added random perturbations to the original noise variances. In Figure 5, we scale δ according to $\beta\delta_{\text{orig}}$ (this time $\delta_{\text{orig}} \approx 6.21 \times 10^4$ owing to random perturbations). Figure 5 shows once again that the approximation gap $\epsilon(\mathbf{x})$ vanishes as δ approaches zero.

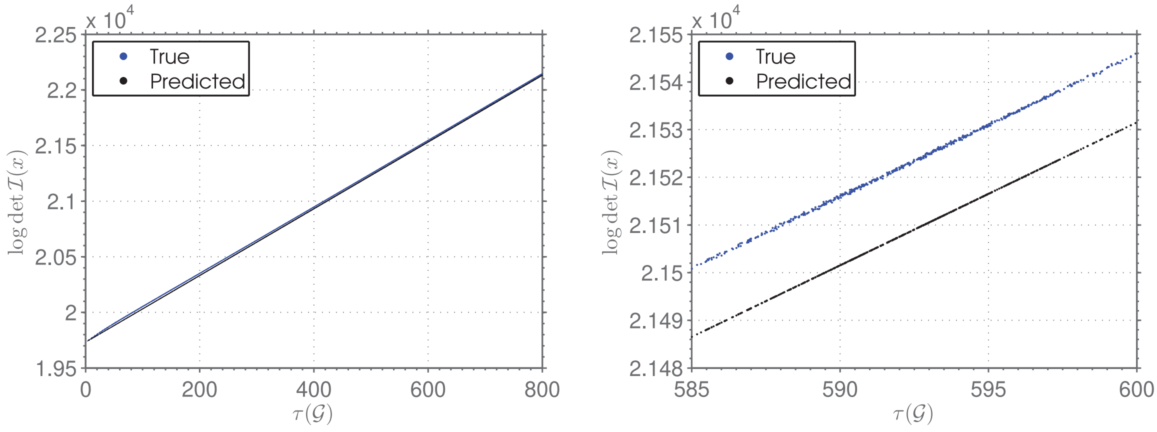
Finally, Figure 6 shows how the D-criterion $\log\det \mathbb{I}(\mathbf{x})$ evolves as a function of $\tau(\mathcal{G})$ for more than 44×10^3 random spanning subgraphs of the Intel Research Lab dataset. Each subgraph contains a random subset of loop-closure edges of the original dataset. For each possible number of loop-closures, we generated 50 random spanning subgraphs. The predicted value is $\ell_\tau(\mathcal{G})$. Note that for a fixed value of tree connectivity $\tau(\mathcal{G})$, variation in the D-criterion (i.e., thickness of the blue line in Figure 6) reflects variation in $\epsilon(\mathbf{x})$; see Theorem 3. Figure 6 suggests that in this experiment, the difference in the D-criterion of SLAM subproblems that have the same tree connectivity is quite small (thickness of blue line relative to the value of D-criterion). This therefore indicates that in this case, tree connectivity almost entirely characterizes the D-criterion.

4. Maximizing tree connectivity

So far we have demonstrated that the WST is closely related to the D-criterion in SLAM and two other EoGs. This observation allows us to reason about the estimation error covariance in such problems based solely on the topology of the underlying graph. Thus, in principle, one can design reliable SLAM problems by synthesizing graph topologies with the maximum WST. Motivated by this insight, in this and the following sections we formulate and tackle the combinatorial optimization problem of designing graphs with the maximum WST under sparsity constraints. We will particularly focus on combinatorial optimization problems that can be used to solve the measurement selection problem in several different settings (see Section 8).

4.1. Characterizing t-optimal graphs in $\mathbb{G}_{n,m}$

We begin by simplifying our notation. Recall that weighted tree connectivity induces a partial ordering on the set of undirected graphs with positive edge weights: two graphs $\mathcal{G}$ and $\mathcal{H}$ are comparable if and only if $|\mathcal{V}(\mathcal{G})| = |\mathcal{V}(\mathcal{H})|$. Let



(a) Overview.

(b) Zoomed region.

Fig. 6. Log-determinant of the FIM $\log\det \mathbb{I}(\mathbf{x})$ as a function of $\tau(\mathcal{G})$ for over 44×10^3 randomly generated spanning subgraphs of the Intel Research Lab dataset. Here $\log\det \mathbb{I}(\mathbf{x})$ is evaluated at the MLE of the original dataset. The prediction is based on the lower bound $\ell_\tau(\mathcal{G})$ provided by Theorem 3, i.e., by using only the graphical structure of the problem.

$\mathbb{G}_n$ be the set of simple undirected graphs with n vertices. Now suppose $\mathbb{G} \subset \mathbb{G}_n$ and a positive weight function $w : \mathcal{E}(\mathcal{K}_n) \to \mathbb{R}_{>0}$ are given. In the most general case, one may seek t-optimal graphs with respect to $\mathbb{G}$ and w, i.e.,

$$\underset{\mathcal{G} \in \mathbb{G}}{\text{maximize}} \quad t_w(\mathcal{G}) \tag{40}$$

Recall that in the special case of uniform edge weights, $t_w(\mathbb{G})$ in (40) can be replaced by the number of spanning trees $t(\mathcal{G})$ without affecting the set of optimal solutions. Let $\mathcal{G}^\star \in \mathbb{G}$ be an optimal design. Note that the number of vertices and the edge weights in (40) are assumed to be given and, thus, the decision variables are graph edges. To emphasize on this point, let $\mathbb{E} \triangleq \{\mathcal{E}(\mathcal{G}) : \mathcal{G} \in \mathbb{G}\}$ be the collection of the edge sets of the graphs in $\mathbb{G}$. With a slight abuse of notation, (40) can be rewritten as

$$\underset{\mathcal{E} \in \mathbb{E}}{\text{maximize}} \quad t_w([n], \mathcal{E}) \tag{41}$$

Lemma 1. *The set of optimal solutions of (40) is invariant under scaling w by any constant* $\alpha>0$.

Define $w_{\min} \triangleq \min w(u, v)$. According to the above lemma, if $w_{\min} < 1$ we can scale every weight by any $\alpha \geqslant w_{\min}^{-1}$ without affecting the t-optimal topologies in (40). Therefore, without losing any generality we can assume the following.

Assumption 5. We assume that $w(u, v) \geqslant 1$ for all adjacent vertices u and v.

For clarity, let us define the WST and weighted tree connectivity as functions of the edge set of the graph. For any $n \geqslant 2$ and $w : \mathcal{E}(\mathcal{K}_n) \to \mathbb{R}_{\geqslant 1}$ define

$$t_{n,w}\colon 2^{\mathcal{E}(\mathcal{K}_n)} \to \mathbb{R}_{\geqslant 0} : \mathcal{E} \mapsto t_w([n], \mathcal{E}) \tag{42}$$

Similarly, we define

$$\tau_{n,w}\colon 2^{\mathcal{E}(\mathcal{K}_n)} \to \mathbb{R}_{\geqslant 0} : \mathcal{E} \mapsto \tau_w([n], \mathcal{E}) \tag{43}$$

Consequently, (41) can be rewritten as

$$\underset{\mathcal{E} \in E}{\text{maximize}} \quad t_{n,w}(\mathcal{E}) \tag{44}$$

or, equivalently, since log is monotone

$$\underset{\mathcal{E} \in E}{\text{maximize}} \quad \tau_{n,w}(\mathcal{E}) \tag{45}$$

It will become clear soon why using $\tau_{n,w}$ is preferred over $t_{n,w}$. Note that (41)–(45) represent the most general case of t-optimal graph synthesis. A natural special case is when $\mathbb{G} = \mathbb{G}_{n,m}$, i.e., the set of all simple undirected graphs with n vertices and m edges. This special case can be expressed as

$$\begin{aligned} &\underset{\mathcal{E} \subset \mathcal{E}(\mathcal{K}_n)}{\text{maximize}} && \tau_{n,w}(\mathcal{E}) \\ &\text{subject to} && |\mathcal{E}| = m \end{aligned} \tag{46}$$

The cardinality constraint enforces a desired sparsity level. In practice, sparsity of the graph is a crucial factor in determining the amount of resources needed for solving the problems that arise over graph structures and networks. In particular, the number of measurements in SLAM influences the computational cost of each iteration in Newton-based iterative solvers.

The problem of characterizing graphs in $\mathbb{G}_{n,m}$ with the maximum number of spanning trees remains open, and is solved only for a number of special cases; see Boesch et al. (2009) for a recent survey. For example, such graphs have been characterized for specific ranges of m (as a function of n) such as when $n - 1 \leqslant m \leqslant n + 3$ (almost-tree graphs) and $\binom{n}{2} - n/2 \leqslant m \leqslant \binom{n}{2}$ (almost-complete graphs). Another major result is due to Cheng (1981) who proved that the family of regular complete multipartite graphs are

t-optimal among all graphs with the same number of vertices and edges. More results can be found in Shier (1974), Cheng (1981), Wang (1994), Kelmans (1996), and Petingi and Rodriguez (2002). Unfortunately the span of these special cases is too narrow for many real applications including EoGs. Furthermore, the $\mathbb{G}_{n,m}$ constraint alone is typically insufficient for characterizing the true set of feasible graphs under spatial and physical constraints. Finally, these results do not cover the case of weighted graphs (or parallel) edges which is essential for our graphical formalism of estimation problems. In the following section, we focus on a class of special cases of (41) that can be used for designing near-D-optimal EoG problems, and particularly SLAM.

4.2. The edge selection problem

Suppose a *connected* base graph and a positive weight function over the set of all possible edges are given. Informally, in the edge selection problem (ESP) we seek t-optimal graphs within the set of graphs whose edge sets are at most k elements different from that of the base graph. This is formally defined as follows.

Problem 1 (k-ESP). The input in k-ESP is as follows: (i) to be set of $n \geqslant 2$ vertices $\mathcal{V} = [n]$; (ii) a planning horizon $k \in \mathbb{N}$; (iii) a connected base graph $\mathcal{G}_{\text{init}} = (\mathcal{V}, \mathcal{E}_{\text{init}})$; (iv) a weight function $w : \mathcal{E}(\mathcal{K}_n) \to \mathbb{R}_{\geqslant 1}$; (v) and a set of $c \geqslant k$ candidate edges $\mathcal{C}$. We seek to solve the following combinatorial optimization problem:

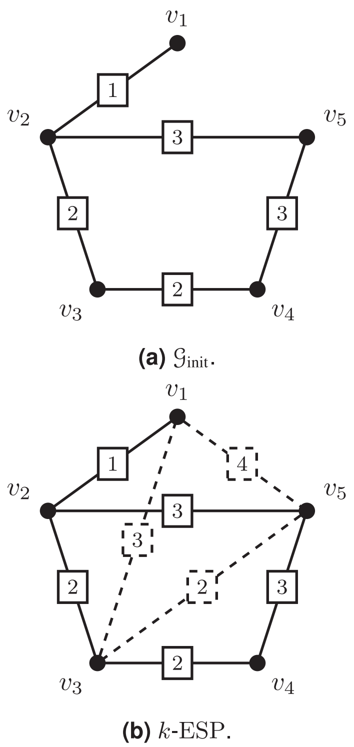
$$\begin{array}{ll} \underset{\mathcal{E} \subset \mathcal{C}}{\text{maximize}} & \tau_{n,w}(\mathcal{E}_{\text{init}} \cup \mathcal{E}) \\ \text{subject to} & |\mathcal{E}| = k \end{array} \tag{47}$$

Figure 7 illustrates an instance of k-ESP.

Remark 5. At first glance, the $\mathbb{G}_{n,m}$ constraint may seem to be a special case of k-ESP with $\mathcal{E}_{\text{init}} = \varnothing$ and $\mathcal{C} = \binom{[n]}{2}$. However, it is worth emphasizing that this is not the case as $\mathcal{E}_{\text{init}} = \varnothing$ violates the assumption of connectedness of the base graph. In some applications, $\mathcal{G}_{\text{init}}$ is naturally connected and therefore we do not lose any practical generality by making this assumption. For example, in pose-graph SLAM the odometry subgraph (i.e., the path graph with n vertices and weighted edges) is a natural choice for $\mathcal{G}_{\text{init}}$.

Remark 6. It is trivial to see that the problem of pruning a k'-subset of a candidate set of edges in the base graph can also be posed as an instance of k-ESP if the base graph remains connected after removing the candidate set.

Solving the general case of k-ESP by exhaustive search requires examining $\binom{c}{k} = \Theta(c^k)$ graphs which is impractical even in rather small problems. For example, for $c = 30$ and $k = 10$ exhaustive search has to compute more than 3×10^7 Cholesky decompositions. To the best of the authors' knowledge, there is no known faster algorithm for finding t-optimal graphs or solving the k-ESP problem in general. In the following sections, we provide efficient near-optimal approximation algorithms for this problem.



(a) $\mathcal{G}_{\text{init}}$.

(b) k-ESP.

Fig. 7. An example of k-ESP. The candidate edges are drawn with dashed lines. The number written on each edge is the weight assigned to that edge.

4.4. 1-ESP

Let us first consider 1-ESP as the simplest instance of k-ESP. As will become clear shortly, the solution of 1-ESP can be used as a building block for finding near-optimal solutions for the general k-ESP problem. An optimal edge in 1-ESP can be found by examining every candidate edge $e \in \mathcal{C}$, and choosing the one that maximizes the WST of the resulting graph $([n], \mathcal{E}_{\text{init}} \cup \{e\})$. Finding an optimal design using this brute force strategy requires computing the tree connectivity of c graphs (one for each candidate edge). If the base graph is dense (worst case), computing tree connectivity can be done in $O(n^3)$ operations using the Cholesky decomposition of the reduced weighted Laplacian matrix (similar to Algorithm 1). The total time complexity in this case is thus $O(cn^3)$. We can improve this computational complexity by leveraging the matrix determinant lemma. According to Lemma 5, the solution of 1-ESP is given by

$$e^\star \in \underset{e \in \mathcal{C}}{\operatorname{argmax}}\, w(e)\Delta_e \tag{48}$$

Algorithm 2. 1-ESP

```
function OneESP($\mathcal{C}, \mathbf{C}$)    ▷ $\mathcal{C}$: Candidate Set, $\mathbf{C}$: Cholesky factor of $\mathbf{L}_{\text{init}}$
    $m \leftarrow 0$    ▷ Maximum value
    for all $e \in \mathcal{C}$ do    ▷ Parallelizable loop
        $w_e \leftarrow w(e)$
        $\Delta_e \leftarrow \text{Reff}(e, \mathbf{C})$
        if $w_e \Delta_e > m$ then
            $e^\star \leftarrow e$
            $m \leftarrow w_e \Delta_e$
        end if
    end for
    return $e^\star$
end function
```

Algorithm 3. Effective resistance

```
function Reff($e_{uv}, \mathbf{C}$)    ▷ Effective Resistance
    // column of the reduced incidence matrix
    $\mathbf{a}_{uv} \leftarrow \text{reduce}(\mathbf{e}_u - \mathbf{e}_v)$
    // solve $\mathbf{C}\mathbf{x}_{uv} = \mathbf{a}_{uv}$
    $\mathbf{x}_{uv} \leftarrow \text{ForwardSolver}(\mathbf{C}, \mathbf{a}_{uv})$    ▷ Lower triangular
    $\Delta_{uv} \leftarrow \| \mathbf{x}_{uv} \|^2$
    return $\Delta_{uv}$
end function
```

Algorithm 4. Greedy Edge Selection

```
function GreedyESP($\mathbf{L}_{\text{init}}, \mathcal{C}, k$)
    $\mathcal{E} \leftarrow \varnothing$
    $\mathbf{L} \leftarrow \mathbf{L}_{\text{init}}$
    $\mathbf{C} \leftarrow$Cholesky($\mathbf{L}$)
    while $|\mathcal{E}| < k$ do
        $e^\star_{uv} \leftarrow \text{OneESP}(\mathcal{C} \backslash \mathcal{E}, \mathbf{C})$
        $\mathcal{E} \leftarrow \mathcal{E} \cup \{e^\star_{uv}\}$
        // column of the reduced incidence matrix
        $\mathbf{a}_{uv} \leftarrow \text{reduce}(\mathbf{e}_u - \mathbf{e}_v)$
        $\mathbf{L} \leftarrow \mathbf{L} + w(e^\star_{uv})\mathbf{a}_{uv}\mathbf{a}_{uv}^\top$
        $\mathbf{C} \leftarrow \text{CholeskyUpdate}(\mathbf{C}, \sqrt{w(e^\star_{uv})}\mathbf{a}_{uv})$    ▷ Rank-one update
    end while
    return $\mathcal{E}$
end function
```

where $\Delta_e \triangleq \mathbf{a}_e^\mathrm{T} \mathbf{L}_{\text{init}}^{-1} \mathbf{a}_e$. Given the Cholesky factor of $\mathbf{L}_{\text{init}}$, one can compute $w(e)\Delta_e$ for all candidate edges in $O(cn^2)$ time. Therefore, in this way, the Cholesky factor of $\mathbf{L}_{\text{init}}$ needs to be computed only once, which takes $O(n^3)$ operations. This results in $O(n^3 + cn^2)$. Solving 1-ESP using this procedure will therefore be much faster than the naïve brute force algorithm outlined above. This process is summarized in Algorithm 2.

Remark 7 (Effective resistance). It is worth noting that Δ_e is the so-called *effective resistance* between the two endpoints of e in the base graph $\mathcal{G}_{\text{init}}$. The effective resistance induces a metric on graphs. Therefore, one can interpret Δ_e as the *distance* between the two endpoints of e in the base graph (*resistance distance*). This metric arises also in a broad range of applications; see Ghosh et al. (2008) for a survey. This insight provides an intuitive interpretation for (48). In the special case of unit edge weights, the optimal candidate edge for maximizing the tree connectivity of the graph in 1-ESP is the one that connects the vertices that are furthest (as measured by the resistance distance) from each other in the base graph.

5. Approximation algorithms for *k*-ESP

5.1. Greedy algorithm

The greedy algorithm finds an approximate solution for *k*-ESP by solving *k* instances of 1-ESP using the procedure described above. After solving each subproblem, an optimal edge is moved from the candidate set $\mathcal{C}$ to the base graph. The next instance of 1-ESP is then defined using the updated candidate set and base graph. In the worst case (dense graphs), a naïve implementation of the greedy algorithm runs in $O(kcn^3)$ time. As mentioned above, this runtime can be reduced to $O(kn^3 + kcn^2)$ by leveraging the matrix determinant lemma (Lemma 5). However, in this case we need to recompute the Cholesky factor of the updated base graph after solving each 1-ESP. Note that the Cholesky factor of the updated base graph can be computed by performing a rank-one update on that of the base graph in the previous round. This operation can be done in $O(n^2)$ time. Therefore, an improved implementation of the greedy algorithm runs in $O(n^3 + kcn^2)$ time. This procedure is described in Algorithm 4. It must be noted that when the base graph and set of candidate edges are sparse, the computational complexity of the greedy algorithm scales much better with n and depends on the sparsity pattern of the corresponding Laplacian matrices.

5.2. Performance guarantees

Now we analyze the performance of the greedy algorithm described in Algorithm 5.

Definition 3 (Tree-connectivity gain). Given an instance of *k*-ESP, the tree connectivity gain is defined as

$$\Phi_w : \mathcal{E} \mapsto \tau_{n,w}(\mathcal{E}_{\text{init}} \cup \mathcal{E}) - \tau_{n,w}(\mathcal{E}_{\text{init}}) \tag{49}$$

The domain of Φ_w is restricted to $2^{\mathcal{C}}$.

Here Φ_w is a set function that takes as input a subset of the candidate edges $\mathcal{E} \subset \mathcal{C}$, and returns the marginal increase in weighted tree connectivity after adding the edges in $\mathcal{E}$ to a given base graph $\mathcal{G}_{\text{init}}$. Now *k*-ESP (47) can be expressed as

$$\begin{aligned} \underset{\mathcal{E} \subset \mathcal{C}}{\text{maximize}} \quad & \Phi_w(\mathcal{E}) \\ \text{subject to} \quad & |\mathcal{E}| = k \end{aligned} \tag{50}$$

Algorithm 5. Greedy Dual Edge Selection

```
function GreedyDualESP($\mathbf{L}_{\text{init}}, \mathcal{C}, \Delta$)
    $\mathcal{E} \leftarrow \varnothing$
    $\mathbf{L} \leftarrow \mathbf{L}_{\text{init}}$
    $\mathbf{C} \leftarrow$ Cholesky($\mathbf{L}$)
    while ($\Phi_w(\mathcal{E}) < \Delta$) & ($\mathcal{E} \neq C$) do
        $e^{\star}_{uv} \leftarrow$ OneESP($\mathcal{C} \backslash \mathcal{E}, \mathbf{C}$)
        $\mathcal{E} \leftarrow \mathcal{E} \cup \{e^{\star}_{uv}\}$
        // column of the reduced incidence matrix
        $\mathbf{a}_{uv} \leftarrow$ reduce($\mathbf{e}_u - \mathbf{e}_v$)
        $\mathbf{L} \leftarrow \mathbf{L} + w(e^{\star}_{uv})\mathbf{a}_{uv}\mathbf{a}_{uv}^{\top}$
        $\mathbf{C} \leftarrow$ CholeskyUpdate($\mathbf{C}, \sqrt{w(e^{\star}_{uv})}\mathbf{a}_{uv}$)    ▷ Rank-one update
    end while
    return $\mathcal{E}$
end function
```

Theorem 4. *The set function Φ_w is normalized, monotone, and submodular for any $n \geqslant 2$, positive weight function w, and connected base graph.*

Maximizing an arbitrary monotone submodular function subject to a cardinality constraint generalizes the maximum coverage problem (Hochbaum, 1996) and, hence, is NP-hard in general. A classical result is due to Nemhauser et al. (1978) who have shown that the greedy algorithm is an η-approximation algorithm for maximizing any normalized monotone submodular function subject to a cardinality constraint where $\eta \triangleq 1 - 1/e \approx 0.63$. Corollary 7 follows directly from Theorems 4 and 14 (Nemhauser et al., 1978).

Corollary 7. *Let* OPT *be the optimal value of (47), τ_{greedy} be the weighted tree connectivity of the graph synthesized by the greedy algorithm, and τ_{init} be that of the base graph. It holds*

$$\tau_{\text{greedy}} \geqslant \eta \cdot \text{OPT} + (1 - \eta) \cdot \tau_{\text{init}} \tag{51}$$

Remark 8. Theorem 4 also allows us to reduce the number of lower triangular linear systems that need to be solved in each round of the greedy algorithm; this is known as the lazy (or accelerated) greedy algorithm (Krause and Golovin, 2012; Minoux, 1978). We leverage this idea to speed up the greedy algorithm in Section 9.4.

5.3. Convex relaxation

In this section, we propose a second approximation algorithm for k-ESP through a convex relaxation inspired by that of Joshi and Boyd (2009). Let us begin by assigning an indicator variable $\pi_i \in \{0, 1\}$ to each candidate edge $e_i \in \mathcal{C}$. We can then express the problem as an integer program where finding the optimal set of candidate edges is equivalent to finding optimal π_i's. Let $\pi \triangleq [\pi_1 \; \pi_2 \cdots \pi_c]^{\mathrm{T}}$ be the stacked vector of indicator variables. Let $\mathbf{L}_{\text{init}}$ denote the reduced weighted Laplacian matrix of $\mathcal{G}_{\text{init}}$. Define,

$$\mathbf{L}_w(\pi) \triangleq \mathbf{L}_{\text{init}} + \sum_{i=1}^{c} \pi_i w(e_i) \mathbf{L}_{e_i} = \mathbf{A}\mathbf{W}^{\pi}\mathbf{A}^{\top}, \tag{52}$$

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

$$w^{\pi} : e \mapsto \begin{cases} \pi_{i_e} w(e) & e \in \mathcal{C} \\ w(e) & \text{otherwise} \end{cases} \tag{53}$$

in which $\pi_{i_e} \in \{\pi_i\}_{i=1}^{c}$ is the indicator variable associated to the candidate edge e.

Lemma 2. *If $\mathcal{G}_{\text{init}}$ is connected, $\mathbf{L}_w(\pi) \succ \mathbf{0}$ for any $\pi \in [0, 1]^c$.*

As before, for convenience we assume $\mathcal{G}_{\text{init}}$ is connected. Now consider the following integer program.

$$\begin{array}{ll} \underset{\pi}{\text{maximize}} & \log\det \mathbf{L}_w(\pi) \\ \text{subject to} & \sum\limits_{i=1}^{c} \pi_i = k \\ & \pi \in \{0, 1\}^c \end{array} \tag{P$_1$}$$

It is easy to verify that the above optimization problem is equivalent to k-ESP in (47). First, recall that according to the generalized matrix-tree theorem (Theorem 12) the objective is equal to the weighted tree connectivity of $\mathcal{G}_{\text{all}} = ([n], \mathcal{E}_{\text{init}} \cup \mathcal{C})$ when edges are weighted by w^{π}. In this new narrative, the ith candidate edge is selected iff $\pi_i = 1$. The combinatorial hardness of k-ESP is embodied in the $\pi \in \{0, 1\}^c$ constraint. A natural choice for relaxing (P_1) is to replace each $\pi_i \in \{0, 1\}$ with $p_i \in [0, 1]$:

$$\begin{array}{ll} \underset{\mathbf{p}}{\text{maximize}} & \log\det \mathbf{L}_w(\mathbf{p}) \\ \text{subject to} & \sum\limits_{i=1}^{c} p_i = k \\ & \mathbf{p} \in [0, 1]^c \end{array} \tag{P$_2$}$$

where, after a minor abuse of notation, $\mathbf{p}$ is the stacked vector of p_i's.[4] It should be immediately clear that the feasible set of (P_2) contains that of (P_1), and therefore the optimal value of (P_2) is an upper bound for the optimal value of (P_1). Note that (P_2) is a convex optimization problem since $\log\det \mathbf{L}_w(\mathbf{p})$ is concave on $\{\mathbf{p} : \mathbf{L}_w(\mathbf{p}) \succ \mathbf{0}\}$, which contains $[0, 1]^c$ when $\mathcal{G}_{\text{init}}$ is connected (Lemma 2), and the feasible set is convex. An ℓ_1-regularized variant of (P_2) is an instance of the MAXDET problem (Vandenberghe et al., 1998),

$$\begin{array}{ll} \underset{\mathbf{p}}{\text{maximize}} & \log\det \mathbf{L}_w(\mathbf{p}) - \lambda \sum\limits_{i=1}^{c} p_i \\ \text{subject to} & \mathbf{p} \in [0, 1]^c \end{array} \tag{P$_3$}$$

(P_3) is also closely related to the graphical lasso (Friedman et al., 2008). The new term in the objective of (P_3) promotes sparsity, while the log-determinant term rewards stronger tree connectivity. The penalty coefficient λ is a parameter that controls the desired degree of sparsity, i.e., a larger λ yields a sparser graph.

5.4. Rounding

Problems (P_2) (and (P_3)) can be solved in polynomial time using interior-point methods (Boyd and Vandenberghe, 2004; Joshi and Boyd, 2009). After finding a globally optimal solution $\mathbf{p}^\star$ for the relaxed problem (P_2), we need to map it into a feasible π for (P_1) and pick k candidate edges accordingly. The following lemma follows trivially from the fact that (P_2) is a relaxation of (P_1).

Lemma 3. *The solution* $\mathbf{p}^\star$ *is an optimal solution for* k-ESP *iff* $\mathbf{p}^\star \in \{0,1\}^c$.

In the more likely case of $\mathbf{p}^\star$ containing fractional values, we need a *rounding procedure* to set k auxiliary variables to one and others to zero. The most intuitive heuristic choice is to pick the k edges with the largest $p_i^\star$ (Joshi and Boyd, 2009). We call this strategy the deterministic rounding by sorting. The relaxation approach described so far can be seen as a graphical specialization of the algorithm proposed by Joshi and Boyd (2009). In the rest of this section, we extend their results and shed more light on the connection between (P_1) and (P_2).

Consider a simple randomized rounding scheme in which we independently sample $\pi_i \sim \mathrm{Bern}(p_i^\star)$ for $i \in [c]$ and pick corresponding candidate edges accordingly.[5] Random graphs generated in this way are called *anisotropic random graphs* (ARGs) (Cohen, 1986). This model generalizes the well-studied Gilbert–Erös–Rényi random graph model (Gilbert, 1959) by allowing anisotropic edge probabilities. The naïve procedure for computing the expected WST in ARGs involves a summation over exponentially many terms. Nonetheless, Cohen (1986) has shown that the expected number of spanning in ARGs is equal to the WST when edges are weighted by their corresponding probabilities, and thus can be computed efficiently using the matrix-tree theorem. In what follows, we generalize this result to the case of edge-weighted ARGs and leverage it to analyze the performance of the rounding scheme introduced above. Later on we generalize this result even further to the sensor selection problem studied by Joshi and Boyd (2009).

Theorem 5. *Let* k^* *and* t_w^* *denote the number of selected candidate edges and the WST attained by the randomized rounding algorithm outlined above with edge (occurrence) probabilities* $\mathbf{p} = [p_1\ p_2 \cdots p_c]^\top$. *It holds*:

1. $\mathbb{E}[k^*] = \sum_{i=1}^{c} p_i$;
2. $\mathbb{E}[t_w^*] = \det \mathbf{L}_w(\mathbf{p})$.

Theorem 5 states that the randomized rounding described above, on average, selects $\sum_{i=1}^{c} p_i$ candidate edges and, in expectation, attains $\det \mathbf{L}_w(\mathbf{p})$ WST. Now note that these expressions appear in the constraints and the objective of the relaxed problem (P_2), respectively. Consequently, this implies that the relaxed problem can be interpreted as the problem of finding the optimal edge sampling probabilities $\mathbf{p}$ for the randomized algorithm described above.

Corollary 8. *The objective in problem* (P_2) *is to find the optimal edge sampling probabilities* $\mathbf{p}^\star$ *such that the WST is maximized in expectation, while the expected number of newly selected edges is equal to k.*

Put differently, problem (P_2) can be seen as a convex relaxation of k-ESP at the expense of maximizing the objective and satisfying the constraint, both *in expectation*. Similarly, problem (P_3) is equivalent to

$$\begin{array}{ll} \underset{\mathbf{p}}{\text{maximize}} & \log \mathbb{E}[\det \mathbf{L}_w(\pi)] - \lambda \mathbb{E}\left[\sum_{i=1}^{c} \pi_i\right] \\ \text{subject to} & \mathbf{p} \in [0,1]^c \end{array} \qquad (\mathrm{P}_3')$$

where $\pi_i \sim \mathrm{Bern}(p_i)$ and $\pi_i \perp \pi_j$ for all $i,j \in [c]$ $(i \neq j)$.

This new interpretation can potentially be used to design randomized rounding procedures based on the randomized scheme described above. Using $\mathbf{p}^\star$ in the above-mentioned randomized scheme, Theorem 5 ensures that, on average, we attain $\det \mathbf{L}_w(\mathbf{p}^\star)$ by picking k new edges in expectation. Needless to say, merely sampling candidate edges with the probabilities in $\mathbf{p}^\star$ is not sufficient to guarantee attaining a feasible solution with high probability.

Now recall the deterministic rounding by sorting scheme and let $\mathcal{S}_{\mathrm{sort}}$ be the edges selected by this algorithm. For any $\mathcal{S} \subset \mathcal{C}$, let $p_k(\mathcal{S})$ denote the conditional probability of the event in which the randomized scheme selects $\mathcal{S}$, given that $k^* = k$ candidates have been selected. We have

$$p_k(\mathcal{S}) \propto \begin{cases} \prod_{e_i \in \mathcal{S}} p_i^\star \prod_{e_j \in \mathcal{C} \setminus \mathcal{S}} (1 - p_j^\star) & |\mathcal{S}| = k \\ 0 & \text{otherwise} \end{cases} \qquad (54)$$

Theorem 6. $\mathcal{S}_{\mathrm{sort}} \in \arg\max_{\mathcal{S} \subset \mathcal{C}} p_k(\mathcal{S})$.

According to Theorem 6, the deterministic rounding via sorting is equivalent to selecting the most probable feasible subset of candidates, when candidates are selected with the probabilities in $\mathbf{p}^\star$. Theorem 5, Corollary 8, and Theorem 6 can be extended to the more general case of D-optimal sensor selection studied in Joshi and Boyd (2009). The only non-trivial piece is provided below.

Theorem 7 (Determinant of random sum of rank-one matrices). *Given m pairs of real n-vectors* $\{\mathbf{u}_i\}_{i=1}^m$ *and* $\{\mathbf{v}_i\}_{i=1}^m$, *and random variables* $\{\pi_i\}_{i=1}^m$ *such that*

$$\pi_i \sim \mathrm{Bern}(p_i) \qquad \mathrm{i} \in [m] \qquad (55)$$

$$\pi_i \perp \pi_j \qquad i,j \in [m],\ i \neq j \qquad (56)$$

we have

$$\mathbb{E}\left[\det\left(\sum_{i=1}^{m}\pi_i\mathbf{u}_i\mathbf{v}_i^{\top}\right)\right]=\det\left(\sum_{i=1}^{m}p_i\mathbf{u}_i\mathbf{v}_i^{\top}\right) \tag{57}$$

Note that the second statement in Theorem 5 follows from Theorem 7 as a special case in which $\{\mathbf{u}_i\}$ and $\{\mathbf{v}_i\}$ are the columns of the reduced (weighted) incidence matrix.

5.5. Certifying near-optimality

It is intractable to compute OPT in real-world instances of k-ESP for (even offline) empirical evaluation of a proposed approximate solution. As a proxy, we can use the approximation algorithms presented in this section to find lower and upper bounds for OPT. Let $\tau^{\star}_{\text{cvx}}$ be the optimal value of (P_2), and τ_{cvx} be the suboptimal value obtained after rounding the fractional solution of (P_2) (e.g., picking the k largest $p_i^{\star}$'s). Corollary 9 follows from the analysis presented for the proposed approximation algorithms.

Corollary 9. *Let* $\mathcal{U}_{\text{greedy}} \triangleq \zeta\tau_{\text{greedy}} + (1-\zeta)\tau_{\text{init}}$ *in which* $\zeta \triangleq \eta^{-1} \approx 1.58$. *It holds that*

$$\max(\tau_{\text{greedy}}, \tau_{\text{cvx}}) \leqslant \text{OPT} \leqslant \min(\mathcal{U}_{\text{greedy}}, \tau^{\star}_{\text{cvx}}) \tag{58}$$

The lower bounds are the suboptimal values attained by the approximation algorithms. More interestingly, $\mathcal{U}_{\text{greedy}}$ follows from Corollary 7, and $\tau^{\star}_{\text{cvx}}$ is an upper bound for OPT since the feasible set of (P_2) contains that of (P_1) (i.e., k-ESP). These bounds can be computed by running the greedy and convex relaxation algorithms. In the instances of k-ESP where OPT is beyond our reach, these bounds can be used to assess and *certify* the quality of any other solution. Let $\mathcal{E}'$ be an arbitrary k-subset of $\mathcal{C}$ and $\tau' \triangleq \tau_{n,w}(\mathcal{E}' \cup \mathcal{E}_{\text{init}})$. $\mathcal{E}'$ can be, e.g., (i) the solution of the greedy algorithm, (ii) the solution of (P_2) after rounding, (iii) an existing design (e.g., an existing pose-graph problem), or (iv) a heuristic suboptimal solution proposed by, for example, an expert. Let $\mathcal{U}$ denote the upper bound in (58). From Corollary 9 we have, $\text{OPT} - \tau' \leqslant \mathcal{U} - \tau'$ and $\text{OPT}/\tau' \leqslant \mathcal{U}/\tau'$ for the approximation gap and ratio, respectively.

6. A dual approach: Δ-ESP*

Our ultimate goal is to design sparse graphs with strong tree connectivity. We pursued this objective in k-ESP by searching for a graph with the maximum weighted tree connectivity among all graphs with certain number of edges constrained by a base graph and a candidate edge set. Alternatively, one may seek the sparsest graph that attains a desired weighted tree connectivity. This dual problem is formally defined below.

Problem 2 (Δ-ESP*). *Problem* Δ-ESP* *aims to select as few edges as possible from a given set of candidate edges* $\mathcal{C}$, *such that adding those edges to a given connected base graph* $\mathcal{G}_{\text{init}} = ([n], \mathcal{E}_{\text{init}})$ *results in a tree connectivity gain of at least* $0 \leqslant \Delta \leqslant \Phi_w(\mathcal{C})$, *i.e.,*

$$\begin{array}{ll}\underset{\mathcal{E}\subset\mathcal{C}}{\text{minimize}} & |\mathcal{E}| \\ \text{subject to} & \Phi_w(\mathcal{E}) \geqslant \Delta\end{array} \tag{59}$$

Note that in k-ESP, one seeks to adapt the "performance" to a given budget constraint on the number of edges (measurements in EoGs). Alternatively, in Δ-ESP* we aim to find the most resources-efficient Δ-valuable subset of edges (measurements). In the rest of this section, we explain how our approximation algorithms designed for k-ESP and their analyses can be adapted to the dual problem Δ-ESP*.

6.1. Greedy algorithm

The greedy algorithm proposed for k-ESP in Algorithm 5.2 solves k instances of 1-ESP. The same algorithm can be applied to Δ-ESP* after a minor modification of the stopping criterion: instead of solving exactly k instances of 1-ESP, we continue solving a sequence of 1-ESP's until the tree connectivity gain is at least Δ (or, alternatively, until we run out of edges in $\mathcal{C}$ which indicates an empty feasible set). The greedy algorithm for approximating the solution of (59) is outlined in Algorithm 5.

6.2. Performance guarantees

Recall that the performance guarantee presented for the greedy algorithm for k-ESP was made possible by Theorem 4 and the seminal work of Nemhauser et al. (1978). Wolsey (1982, Problem (Q)) has analyzed the performance of the greedy algorithm for solving the submodular set covering problem:

$$\begin{array}{ll}\underset{\mathcal{A}\subset\mathcal{W}}{\text{minimize}} & \sum_{a_i\in\mathcal{A}} f_i \\ \text{subject to} & z(\mathcal{A}) = z(\mathcal{W})\end{array} \tag{60}$$

in which $f_i > 0$ for all $a_i \in \mathcal{W}$ and $z\colon 2^{\mathcal{W}} \to \mathbb{R}$ is any monotone submodular function. Now note that Δ-ESP* (59), is a special case of (60), in which $\mathcal{W} = \mathcal{C}$, $f_i = 1$ for all $a_i \in \mathcal{A}$, and finally z is $\mathcal{A} \mapsto \min\{\Delta, \Phi_w(\mathcal{A})\}$. This function is monotone submodular since Φ_w is monotone submodular (Theorem 4) and constant truncation preserves both properties (Theorem 15) as noted by Wolsey (1982).

Let f_{greedy} and OPT be the value of the greedy algorithm and the optimal value of (60). For the general case of (60), where z does not need to be an integer-valued function, Wolsey (1982: Theorem 1) has established three *a posteriori* performance guarantees of the form

$$f_{\text{greedy}} \leqslant (1 + \log\gamma)\text{OPT} \tag{61}$$

for several different values of γ. These bounds are *a posteriori* in the sense that the value of γ can be determined only *after* running the greedy heuristic. The following corollary states one of these bounds for the performance of the greedy algorithm (Algorithm 6.2) in Δ-ESP*.

Corollary 10. *Let k_{opt} and k_{greedy} be the optimal value of (59) and the value achieved by Algorithm 5, respectively. Also, let $\widetilde{\Phi}_{\text{greedy}}$ be the tree connectivity gain achieved by the greedy algorithm one step before termination. It holds*:

$$k_{\text{greedy}} \leqslant (1 + \log \gamma) k_{\text{opt}} \tag{62}$$

where $\gamma \triangleq \Delta / (\Delta - \widetilde{\Phi}_{\text{greedy}})$.

6.3. Convex relaxation

We now demonstrate how the dual problem Δ-ESP* can be formulated as an integer program and relaxed similar to (P_1). As before, let $\tau_{\text{init}} \triangleq \log\det \mathbf{L}_w(\mathbf{0})$. The dual problem can be expressed as

$$\begin{array}{ll} \underset{\pi}{\text{minimize}} & \sum_{i=1}^{c} \pi_i \\ \text{subject to} & \log\det \mathbf{L}_w(\pi) \geqslant \Delta + \tau_{\text{init}} \\ & \pi \in \{0,1\}^c \end{array} \tag{D$_1$}$$

Relaxing the integral constraints on π yields the following convex optimization problem,

$$\begin{array}{ll} \underset{\mathbf{p}}{\text{minimize}} & \sum_{i=1}^{c} p_i \\ \text{subjectto} & \log\det \mathbf{L}_w(\mathbf{p}) \geqslant \Delta + \tau_{\text{init}} \\ & \mathbf{p} \in [0,1]^c \end{array} \tag{D$_2$}$$

As we saw earlier for k-ESP, (D_2) is a convex optimization problem when $\mathcal{G}_{\text{init}}$ is connected and can be solved efficiently using interior-point methods. Let $\mathbf{p}^\star$ be the minimizer of (D_2). It immediately follows that $\lceil \sum_{i=1}^{c} p_i^\star \rceil \leqslant k_{\text{opt}}$.

6.4. Rounding

Lemma 4. $\mathbf{p}^\star$ *is an optimal solution for* Δ-ESP* *iff* $\mathbf{p}^\star \in \{0,1\}^c$.

In general, $\mathbf{p}^\star$ may contain fractional values, and thus a rounding scheme is necessary to map $\mathbf{p}^\star$ into a feasible (suboptimal) solution for (D_1). A natural deterministic rounding scheme is outlined in Algorithm 6. An efficient implementation of this algorithm using binary search runs in $O(c \log c)$ time.

As we saw before, the fractional values $\mathbf{p}$ can be interpreted as the probability of independently selecting candidate edges. Hence, similar to Corollary 8, the following corollary readily follows from Theorem 5.

Algorithm 6. Deterministic rounding for δ-ESP*

```
function RoundDualESP($\mathbf{p}^\star$)
    $\pi \leftarrow \mathbf{0}$
    $i \leftarrow 0$
    // returns the indices of the sorted $\mathbf{p}^\star$
    $\mathbf{s}$ ←SortDescending($\mathbf{p}^\star$)        ▷ Such that $p^\star_{s_1} \geqslant p^\star_{s_2} \geqslant \ldots \geqslant p^\star_{s_c}$
    // implemented as binary search
    while ( $\log\det \mathbf{L}_w(\pi) < \Delta + \tau_{\text{init}}$) & ($\pi \neq \mathbf{1}$) do
        $\pi_{s_i} \leftarrow 1$
        $i \leftarrow i + 1$
    end while
    return $\pi_{\text{sort}} = \pi$
end function
```

Corollary 11. *The objective in* (D_2) *is to find the optimal probabilities* $\mathbf{p}^\star$ *for sampling edges from* $\mathcal{C}$ *such that the expected value of the number of selected edges is minimized while the expected value of the WST is at least* $\exp(\tau_{\text{init}} + \Delta)$*, i.e.*,

$$\begin{array}{ll} \underset{\mathbf{p}}{\text{minimize}} & \mathbb{E}[\sum_{i=1}^{c} \pi_i] \\ \text{subject to} & \log \mathbb{E}[\det \mathbf{L}_w(\pi)] \geqslant \Delta + \tau_{\text{init}} \\ & \mathbf{p} \in [0,1]^c \end{array} \tag{D$'_2$}$$

where $\pi_i \sim \text{Bern}(p_i)$ and $\pi_i \perp \pi_j$ for all $i, j \in [c]$ $(i \neq j)$.

Thus, sampling edges independently with probabilities in $\mathbf{p}^\star$ will result in selecting

$$\mathbb{E}\left[\sum_{i=1}^{c} \pi_i\right] = \sum_{i=1}^{c} p_i^\star$$

edges on average, while also satisfying the constraint in expectation. Now, as we saw earlier for k-ESP, this narrative can provide a new interpretation for the deterministic rounding procedure in Algorithm 6. Recall that $p_k(\mathcal{S})$ was defined to be the conditional probability of sampling exactly the set $\mathcal{S} \subset \mathcal{C}$, given that exactly k candidates have been selected. Let $\mathcal{S}_k \in \arg\max_{\mathcal{S} \subset \mathcal{C}} p_k(\mathcal{S})$. Now from Theorem 6 it readily follows that Algorithm 6 selects $\mathcal{S}_{k_{\min}}$ where $k_{\min}$ is the smallest $k \in [c]$ such that $\Phi_w(\mathcal{S}_k) \geqslant \Delta$.

6.5. Certifying near-optimality

Finding an optimal solution of Δ-ESP* by brute force is impractical in real-world instances of the dual problem, and thus k_{opt} is generally beyond our reach in practice even for offline empirical evaluation. Fortunately, as we saw in the case of k-ESP, our approximation algorithms can provide lower and upper bounds for k_{opt} that can be used as proxies.

Corollary 12. *Define $\zeta^* \triangleq 1/(1 + \log \gamma)$ where γ is the parameter defined in Corollary 10. Let k_{cvx} be the number of new edges selected by the deterministic rounding procedure in Algorithm 6. Then,*

$$\max\left(\lceil \zeta^* k_{\text{greedy}} \rceil, \left\lceil \sum_{i=1}^{c} p_i^\star \right\rceil\right) \leqslant k_{\text{opt}} \leqslant \min\left(k_{\text{greedy}}, k_{\text{cvx}}\right) \tag{63}$$

These bounds can be computed by running the corresponding approximation algorithms. They can then be used to bound the gap between k_{opt} and any suboptimal design $\mathcal{E}'$ with a value of k'. Let $k_{\mathcal{L}}$ denote the lower bound in Corollary 12. Then we have $k' - k_{\text{opt}} \leqslant k' - k_{\mathcal{L}}$ and $k'/k_{\text{opt}} \leqslant k'/k_{\mathcal{L}}$.

7. Random edge selection

In some applications, candidate edges represent *potential* choices whose existence (or "occurrence") is revealed only *after* solving the ESP. This may be due to the fact that determining the occurrence of a candidate edge requires performing a costly operation. For example, to determine whether a potential loop-closure edge exists between two particular poses in pose-graph SLAM, a robot first has to match the corresponding pair of images or laser scans (and, in multi-robot scenarios, exchange observations (Giamou et al., 2018)), which incurs a cost in terms of mission-critical resources. This type of uncertainty can be incorporated into k-ESP by utilizing the mathematical machinery developed earlier in Theorem 5 (and Theorem 7) for analyzing randomized rounding. For simplicity, let us assume that the edges of the base graph are deterministic. Our results, however, can be trivially extended to the case where the base graph also contains random edges, as long as it remains connected with a positive probability. ARGs can capture the uncertainty over the occurrence of candidate edges (see the discussion that led to Theorem 5). In particular, we consider the case where the ith candidate edge is "operational" with probability p_i and "fails" with probability $1 - p_i$, independent of other candidates.

Now given a k-ESP with occurrence probabilities of the candidate edges $\{p_i\}_{i=1}^{c}$, one may instead seek to maximize the *expected* WST by choosing k candidates. We call this the random edge selection problem (RESP) or k-RESP:

$$\begin{array}{ll} \underset{\mathcal{E} \subset \mathcal{C}}{\text{maximize}} & \mathbb{E}[t_{n,w}(\mathcal{E}_{\text{init}} \cup \mathcal{E})] \\ \text{subject to} & |\mathcal{E}| = k \end{array} \tag{64}$$

From Theorem 5 recall that the objective is equal to the WST after scaling candidate edge weights by the corresponding occurrence probabilities. Thus, any k-RESP can be cast and solved (approximately) as a k-ESP after scaling the edge weights.

In some applications, one may wish to choose candidates such that k operational edges are selected in expectation. Let $o(\mathcal{E})$ denote the number of operational edges after selecting $\mathcal{E}$. This problem can be expressed as

$$\begin{array}{ll} \underset{\mathcal{E} \subset \mathcal{C}}{\text{maximize}} & \mathbb{E}[t_{n,w}(\mathcal{E}_{\text{init}} \cup \mathcal{E})] \\ \text{subject to} & \mathbb{E}[o(\mathcal{E})] \leqslant k \end{array} \tag{65}$$

Note that $\mathbb{E}[o(\mathcal{E})] = \sum_{i \in \mathcal{E}} p_i$ and, thus, this constraint can be easily incorporated into our convex relaxation scheme. Furthermore, after a small modification, a greedy algorithm provides a $1/2 \cdot (1 - 1/e)$ approximation factor for solving (65) (Leskovec et al., 2007).

8. Applications

The theoretical results presented in Section 3 (particularly, Theorem 3) reveal that designing D-optimal SLAM problems is closely related to designing graphs with the maximum WST. This led to the study of a class of constrained t-optimal graph synthesis problems. In this section, we show how our near-optimal graph synthesis framework can be used for measurement selection and pose-graph (edge) pruning in SLAM.

8.1. Measurement selection

Mobile robots are constrained by weight, size, and power budgets, limiting the capability of onboard hardware. This may leave a robot unable to process all the raw data its sensors provide to it in real time. Therefore, SLAM systems must be able to adapt to an allocated budget of mission-critical resources (e.g., battery, bandwidth, and CPU time). This is a critical prerequisite for scalable and long-term autonomy. Measurement selection is one of several mechanisms through which SLAM systems can achieve resource adaptation. In measurement selection, a robot seeks to select an "information-rich" budget-feasible subset of existing, newly acquired, and/or potential measurements. In this work, a budget-feasible subset refers to a subset of size at most k measurements for a given budget k. Measurement selection in SLAM can bring substantial resource savings in three different ways: (i) by reducing the computational cost per iteration of sparse linear back-end SLAM solvers; (ii) by reducing the computational cost of SLAM front-end loop-closure detection; and (iii) by reducing data transmission during inter-robot loop-closure discovery in collaborative SLAM. We briefly discuss each of these cases in the following.

1. **Back-end computational cost.** The computational cost per iteration of Newton-based solvers is monotone in the measurement set. In particular, the number of non-zero entries (NNZ) in the Cholesky factor

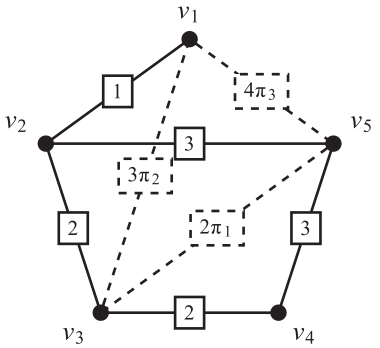
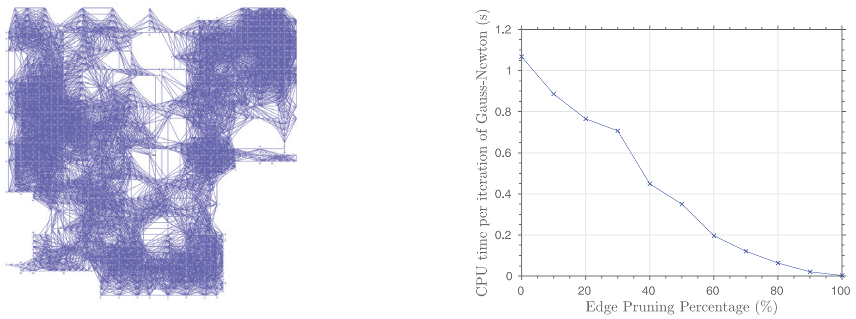


Fig. 8. Expressing the k-ESP depicted in Figure 7 as an integer program with indicator variables π_1, π_2, π_3.



(a) Synthetic dataset (b) Average CPU time per Gauss–Newton iteration

Fig. 9. A synthetic Manhattan-like pose graph generated using g2o's simulator (Kümmerle et al., 2011). Sensor range and field of view are 5 meters and π radians, respectively (default parameters). In addition, loop closures can be established only if the headings of the corresponding two poses are at most π radians apart. The robot follows a random walk (each step: 1 meter forward motion or $\pi/2$ rotation) in a 45 m × 45 m environment. There are 3000 poses and about $81,000$ loop closures. Gauss–Newton can solve this (low-noise regime) problem in few iterations (<5). For example, the total runtime for solving the original dataset on an Intel Core i7-6820HQ CPU operating at 2.70 GHz is about 4.52 seconds.

(excluding numerical cancelation) is determined by the NNZ in the information matrix and the resulting fill-in. The NNZ in the coefficient matrix is $\Theta(m+n)$ where m and n denote the number of loop closures and poses, respectively, while fill-in depends on the sparsity pattern of the information matrix and the heuristic ordering used (see, e.g., Dellaert and Kaess, 2006). Although measurement selection techniques should ideally consider both m and the fill-in, finding the fill-in minimizing ordering is NP-hard (see, e.g., Frey et al., 2018). Despite the natural sparsity of SLAM, the number of loop closures grows rapidly if a robot revisits an area multiple times, or in collaborative SLAM, where multiple robots may cross one another's trajectories repeatedly. These scenarios are especially important in the context of lifelong SLAM where, e.g., a service robot operates in an office building or campus (Huang et al., 2013). Pruning highly redundant pose graphs becomes inevitable because of the limited mission-critical resources available onboard. Figure 9 illustrates this type of scenario using a realistic synthetic dataset generated by g2o's simulator (Kümmerle et al., 2011). This figure shows the impact of pruning loop closures on the runtime of a Gauss–Newton solver. Note that the computational cost of online incremental solvers such as iSAM (Kaess, 2008) that require periodic batch updates has a similar dependence on m; see also (Kaess and Dellaert, 2009)

2. **Front-end computational cost.** Loop-closure detection and establishing relative pose measurements require matching two observations ("sensor registration" hereafter). Standard SLAM front-ends first select a *potential* subset of all pairs of robot poses for sensor registration. This is typically done by taking into account various criteria such as geometric information (i.e., sensor characteristics, pose estimates and their uncertainties) and appearance-based similarity metrics (Gálvez-López and Tardos, 2012; Giamou et al., 2018; Mur-Artal et al., 2015). After this initial pruning, we may still be left with a considerable number of potential loop-closure candidates. This situation arises when pose estimates are highly uncertain, or when appearance-based schemes are weakened by perceptual aliasing. Resource-constrained platforms may benefit from *front-end* measurement selection by restricting the search for loop closures to the k "best" candidate matches. Selecting potential measurements at this stage (i.e., *before* sensor registration) necessitates taking into account the stochastic nature of *potential* loop closures. Note that by constraining the number of sensor registrations, one also controls the growth of NNZ in the information matrix (and hence the cost per iteration of the SLAM back-end).
3. **Communication cost in cooperative SLAM.** Inter-robot loop closure detection in cooperative SLAM incurs an additional communication cost as robots have to exchange their corresponding observations with each other (e.g., laser scans, visual features, etc.; see, e.g., Giamou et al., 2018). In general, the minimum number of exchanges needed to verify a set of potential inter-robot loop closures between two robots is determined by the vertex cover number of a bipartite *exchange* graph whose edges represent potential loop closures and whose vertices correspond to robot poses (Giamou et al., 2018). Consequently, by limiting the number of inter-robot sensor registrations to k or fewer, we also guarantee that at most k exchanges will be needed for loop-closure verification.[6]

8.2. Related works

Joshi and Boyd (2009) and Shamaiah et al. (2010) proposed approximation algorithms for D-optimal sensor selection under linear measurement models with additive Gaussian noise. The approach of Joshi and Boyd (2009) is based on convex relaxation, while Shamaiah et al. (2010) leverage the submodular property of the log-determinant and provide near-optimality guarantees. Although these works are originally proposed for linear-Gaussian models, they can be applied to nonlinear measurement models in SLAM using a *fixed* linearization point.[7] Our approximation algorithms for k-ESP exploit the same structures to maximize the WST. However, backed by our results in Section 3, we apply a novel graphical approach to D-optimal measurement selection in SLAM. In hindsight, our approach can be intuitively interpreted as a special *graphical linearization* specifically designed for SLAM as an EoG problem.

Controlling the growth of computational complexity of SLAM systems is essential for scalability and has been a central research direction: (see Bailey and Durrant-Whyte, 2006; Cadena et al., 2016: and references therein). Pruning state variables (i.e., nodes in the graphical representation of SLAM) is one of the most common approaches. Nodes can be pruned by marginalizing out the corresponding variables from the belief. However, this process creates fill-in. Consequently, the reduced belief is often approximated by pruning the fill-in, i.e., pruning edges from the underlying Gaussian Markov random field (Thrun et al., 2004). Many variations of node pruning and belief sparsification have been proposed (see, e.g., Carlevaris-Bianco et al., 2014; Choudhary et al., 2015; Eade et al., 2010; Huang et al., 2013; Kretzschmar and Stachniss, 2012; Mazuran et al., 2016; Paull et al., 2016, 2015; Vallve et al., 2018; Vial et al., 2011; Wang et al., 2013). The key difference between our approach and these works is that we prune edges (i.e., loop closures in pose graphs) directly from the pose graph, while the above-mentioned works prune nodes (variables) and the resulting fill-in (edges in the underlying probabilistic graphical model). It is worth noting that in principle, belief sparsification schemes may also be used for pruning measurements instead of fill-in (Huang et al., 2013). However, by directly pruning measurements we neither commit to a particular linearization point, nor do we introduce unnecessary artifacts in the belief. In addition, the belief created by measurement selection is “conservative” and “consistent” (Vial et al., 2011) by design.[8] We also note that by pruning a loop closure between poses i and j, one also avoids the fill-in that would have been created after marginalizing out either one of these poses in the first place.

Davison (2005) pioneered information-theoretic measurement selection by proposing an active search strategy for visual SLAM that greedily matches features based on mutual information. This work was later extended by Chli and Davison (2009) who adopted a Gaussian mixture model. Kaess and Dellaert (2009) presented a computationally efficient measurement selection scheme similar to Davison (2005) for discarding redundant and uninformative measurements. Ila et al. (2010) study pose-graph SLAM and propose a strategy for evaluating the information gain of potential loop closures *before* allocating resources to sensor registration. Kretzschmar and Stachniss (2012) presented a scheme for discarding laser scans based on approximating the mutual information between laser scans and grid maps. More recently, Carlone and Karaman (2017) took advantage of submodularity of the log-determinant and approximate submodularity of the smallest eigenvalue of the information matrix (E-optimality criterion) for selecting the k most informative visual measurements in visual inertial navigation for resource-constrained micro aerial vehicles. The objective function used for measurement selection in Carlone and Karaman (2017) also takes into account the probability of successfully tracking features. Our work is similar to the above-mentioned works in that we also select an informative subset of measurements, albeit based on a graphical surrogate for the D-criterion; see Section 8.5 for a discussion of the advantages of our approach. Inspired by Ila et al. (2010), we also present a front-end (potential) loop-closure selection scheme based on tree connectivity to perform sensor registration only for information-rich candidate loop closures (detected based on visual similarity or geometry). Our framework takes the probability of obtaining positive matches from potential loop closures into account by exploiting the closed-form expression for computing the expected WST in random graphs (see Theorem 5).

8.3. D-optimal measurement selection

The measurement selection problem is captured by our k-ESP t-optimal graph synthesis formulation. There is, however, a subtle difference between the two formulations. In the synthesis problems studied so far, each edge is weighted by a single weight function. However, in SLAM each relative pose measurement is composed of two components (i.e., translational and rotational), each of which has its own precision; see w_p and w_θ in Theorem 3. Thus, we need to revisit the synthesis problem in a more general setting, where multiple weight functions assign weights, simultaneously, to a single edge.[9] Fortunately, our near-t-optimal graph synthesis framework and its analysis can be easily generalized to handle the expression that appears in Theorem 3 with multiple weight functions.

1. **Greedy algorithm**: For the greedy algorithm, we just need to replace Φ_w with

$$\Psi : \mathcal{E} \mapsto 2\Phi_{w_p}(\mathcal{E}) + \Phi_{w_\theta}(\mathcal{E}) \tag{66}$$

which appears in $\ell_\tau(\mathcal{G})$; see Theorem 3. Note that Ψ 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 approach can also be generalized by replacing the concave objective function $\log\det \mathbf{L}_w(\pi)$ with the following concave function,

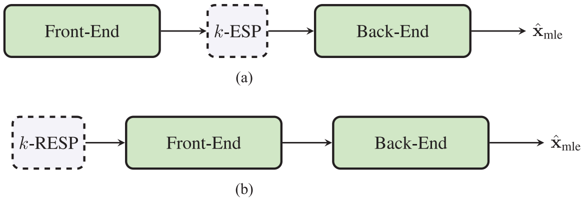

Front-End
k-ESP
Back-End
$\hat{\mathbf{x}}_{\text{mle}}$
(a)
k-RESP
Front-End
Back-End
$\hat{\mathbf{x}}_{\text{mle}}$
(b)

Fig. 10. An overview of SLAM pipeline after incorporating the proposed measurement selection frameworks. In (a), measurement selection is performed after the existence of edges have been confirmed with sensor registration. In (b) we have to choose random edges with occurrence probabilities to maximize the expected weighted tree connectivity.

$$2 \cdot \log\det \mathbf{L}_{w_p}(\pi) + \log\det \mathbf{L}_{w_\theta}(\pi) \tag{67}$$

which appears as $\ell_\tau(\mathcal{G})$ in Theorem 3.

Therefore, we can easily use the approximation algorithms developed for *k*-ESP to find a near-optimal *k*-subset of the measurements.

8.4. D-optimality-aware SLAM front-end

In this section, we motivate and develop a SLAM front-end that is capable of allocating mission-critical resources to search for potential loop closures that significantly improve D-optimality through improving the tree connectivity of the underlying graph. It is worth emphasizing that this scenario is different from the measurement selection problem presented above (i.e., back-end measurement selection), where candidate edges correspond to (true) loop closures that have already been extracted from raw observations (i.e., after sensor registration); see Figure 10(a).

We propose a D-optimality-aware SLAM front-end based on *k*-RESP (Section 7) for selecting a subset of potential candidates *before* performing sensor registration (Figure 10(b)). In our model, we assume a prior occurrence probability can be computed for any potential loop closure, and that loop closures occur independently. The occurrence probabilities can be *estimated* based on sensor specifications (e.g., sensing range) and estimated robot trajectory, or through the use of similarity scores provided by a place recognition system such as DBoW2 (Gálvez-López and Tardos, 2012; see also Tian et al., 2018). The independence assumption provides tractability, and allows us to use the provable guarantees presented for *k*-RESP. We evaluate this model in Section 9 with real and synthetic SLAM datasets. Now suppose that due to resource constraints, a robot is only capable of performing *k* sensor registrations within a fixed interval of *h* poses. Consequently, in each interval, the robot has to gather an initial set of candidates $\mathcal{P}$ consisting of potential loop closures between either (i) a pose in the next *h* poses and one of the previous poses, or (ii) two of the *h* new poses. We then estimate the occurrence probabilities for each *potential* candidate in $\mathcal{P}$. At this point, we need to select *k* promising potential edges from $\mathcal{P}$, perform scan registration (for full verification and computing the relative transformation between the corresponding poses), and add the resulting loop-closure measurements to the pose graph.

This problem is closely related to *k*-RESP, where the goal is to select *k* random edges such that the expected WST in the resulting graph is maximized. Using a similar idea, we solve the following problem:

$$\begin{array}{ll} \underset{\mathcal{E} \subset \mathcal{P}}{\text{maximize}} & 2\log \mathbb{E}[t_{n,w_p}(\mathcal{E})] + \log \mathbb{E}[t_{n,w_\theta}(\mathcal{E})] \\ \text{subject to} & |\mathcal{E}| = k \end{array} \tag{68}$$

Like *k*-RESP, the proposed near-*t*-optimal approximation algorithms and their provable guarantees readily generalize to this case. In practice, it is necessary to *prefilter* $\mathcal{P}$ by removing highly improbable candidates before solving the above problem since selecting a false potential loop closure with a low occurrence probability may still seem desirable for maximizing the *expected* WST. Unlike Ila et al. (2010), this work explicitly takes into account both the occurrence probability and information gain (albeit using our graphical approximation) by incorporating expected values in our objective function. Furthermore, while Ila et al. (2010) mainly considered the case where $h = 1$, our approach shows more flexibility in dealing with longer horizons as we only need to reason about the resulting graph topologies and not relying on open-loop estimates. Indeed, estimating the information gain by evaluating the Jacobian at the open-loop estimate may lead to erroneous selections.

The D-optimality-aware front-end based on *k*-RESP as proposed above can also be used for *inter-robot* loop-closure detection in collaborative SLAM. Robots need to share their observations with their peers in order to establish inter-robot loop closures (see, e.g., Cieslewski and Scaramuzza, 2017; Dong et al., 2015; Giamou et al., 2018). As such, in addition to computational costs of sensor registration, in collaborative SLAM robots are also subject to communication costs that prevent them from sharing the entirety of their beliefs and observations for inter-robot loop closure detection (Giamou et al., 2018; Tian et al., 2018). Our D-optimality-aware front-end can ensure that the limited allocated resources are used to extract highly informative (in expectation) inter-robot loop closures. Potential inter-robot loop closures and their probabilities can be obtained by exchanging compact *clues* such as bag-of-words vectors (see Cieslewski and Scaramuzza, 2017; Giamou et al., 2018; Tian et al., 2018). By selecting *k* informative potential inter-robot loop closures, one can bound both the number of observation exchanges (i.e., communication cost) and sensor registrations (i.e., computational cost) for inter-robot loop closure detection by *k*. It is worth noting that while it is true that in the worst case verifying *k* potential inter-robot loop closures requires exchanging *k* observations, in practice this can often be done by fewer exchanges as shown in Giamou et al. (2018) (see Tian et al. (2018) for inter-robot loop-closure detection under budgeted communication using a precise model of communication cost).

Since the expected WST only depends on the structure of the graph and not on the underlying geometry, it is uniquely suited to multi-robot settings where inter-robot geometry has not yet been reliably established. This is in contrast to the D-criterion which requires an accurate (fixed) linearization point for robots' trajectories. Consequently, in Section 9 we present experimental results for collaborative SLAM scenarios to motivate measurement selection.

8.5. Observed D-optimality versus tree connectivity

Recall that in general nonlinear estimation problems such as SLAM, evaluating the D-criterion requires a linearization point. In principle, one should seek the *asymptotic* covariance matrix of the ML estimator, or equivalently, the CRLB, which is given by the inverse of the FIM $\mathbb{I}(\mathbf{x})$ evaluated a the *true value* $\mathbf{x} = \mathbf{x}_{\text{true}}$. Since $\mathbf{x}_{\text{true}}$ is not available, the D-criterion is usually approximated by evaluating $\log\det\mathbb{I}(\mathbf{x})$ at the MLE (hereafter, "observed D-criterion").

In Section 9 we empirically demonstrate that measurement selection based on our graphical surrogate objective provides a comparable performance to measurement selection based on the observed D-criterion, while being significantly faster. Here we briefly discuss three advantages of using tree connectivity over the observed D-criterion for measurement selection in SLAM.

Computational cost. From a computational standpoint, measurement selection based on the observed D-criterion would be more expensive than the proposed approach. Here we justify this using a worst-case computational complexity analysis; see Section 9 for experimental results. Recall that in each round of the greedy algorithm, we need to assess the remaining candidate loop closures. Rather than evaluating the actual objective function once for each candidate, we can compute the Cholesky factorization only once in each round, and then use the matrix determinant lemma for comparing the candidates (see Section 4.2 and Algorithm 2). Using the tree connectivity as the objective function, in the rth round of the greedy algorithm we need to compute two Cholesky decompositions of $n \times n$ matrices (for $\mathbf{L}_{w_p}$ and $\mathbf{L}_{w_\theta}$), followed by solving $2 \cdot c_r$ (for $\mathbf{L}_{w_p}$ and $\mathbf{L}_{w_\theta}$) $n \times n$ triangular linear systems where $c_r \triangleq c - r + 1$ is the number of remaining candidates. The total floating-point operations (flop) count for the most expensive steps in each round of the greedy algorithm is therefore $\sim 2 \cdot n^3 + 2 \cdot c_r \cdot n^2$. Now in the case of D-criterion, we need to compute the Cholesky factor of the $3n \times 3n$ information matrix, followed by solving $3 \cdot c_r$ (3 is the dimension of each measurement in pose-graph SLAM) triangular linear systems. The flop count for the D-criterion objective is $\sim (3n)^3 + 3 \cdot c_r \cdot (3n)^2 = 27 \cdot n^3 + 27 \cdot c_r \cdot n^2$ (i.e., roughly 13.5 times the flop count for tree connectivity).[10]

Linearization error and convergence failure. In practice, the solution obtained by iterative methods such as Gauss–Newton (applied to the negative log-likelihood) is treated as the MLE. These techniques are subject to local minima and, therefore, may fail to convergence to the correct solution. Measurement selection under such erroneous evaluations of the D-criterion objective is clearly unreliable. Formally, from Theorem 3 we know that $\log\det\mathbb{I}(\mathbf{x}) = 2\tau_{w_p}(\mathcal{G}) + \tau_{w_\theta}(\mathcal{G}) + \epsilon(\mathbf{x})$. The first two terms are independent of $\mathbf{x}$ and thus the error between the D-criterion evaluated at the ground truth and, e.g., a local minimum $\tilde{\mathbf{x}}$ can be written as

$$|\log\det\mathbb{I}(\tilde{\mathbf{x}}) - \log\det\mathbb{I}(\mathbf{x}_{\text{true}})| = |\epsilon(\tilde{\mathbf{x}}) - \epsilon(\mathbf{x}_{\text{true}})| \quad (69)$$

In general, this error term is unbounded and can grow arbitrarily large depending on the error in $\tilde{\mathbf{x}}$. Similarly, even in cases where the true MLE is available, significant estimation errors may still cause the observed D-criterion to deviate significantly from the true D-criterion (i.e., evaluated at the true value), which can potentially affect the quality of selected measurements. By contrast, the proposed tree connectivity surrogate objective is inherently immune to such convergence errors since it does not depend on $\mathbf{x}$; see the case of the Lincoln Lab dataset in Table 1 for an example.

9. Experiments

In this section, we present experimental results based on randomly generated graphs, realistic simulations, and real SLAM benchmark datasets to validate our theoretical results and evaluate the performance of the proposed approximation algorithms. The experiments are specifically designed to demonstrate and evaluate the use of our near-t-optimal graph synthesis framework for measurement selection in SLAM. We implemented our algorithms in MATLAB. Problem (P_2) (convex relaxation) is modeled using YALMIP (Löfberg, 2004) and solved using SDPT3 (Tütüncü et al., 2003). The fractional solution of the convex program is rounded using the deterministic (sorting) rounding scheme. We used an Intel Core i7-6820HQ CPU operating at 2.70 GHz to run our experiments.

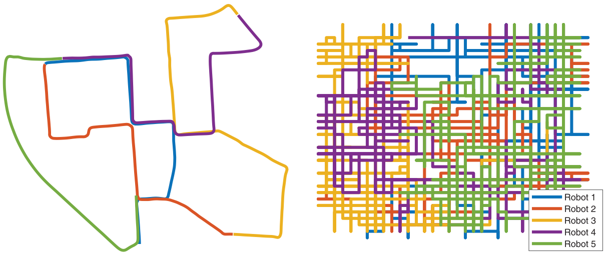
We introduce in Figure 11(a) a dataset based on KITTI odometry sequence 0 (Geiger et al., 2013). This sequence was chosen due to the existence of a significant number of loop closures created by overlaps in the trajectory. In the experiments of Section 9.3, the occurrence probabilities for candidate edges in k-RESP are generated according to the normalized DBoW2-based score α described in Gálvez-López and Tardos (2012). While this quantity is not an exact empirical probability estimate of edge likelihood, our experiments demonstrate that they are suitable proxies. In order to simulate loop-closure candidates that would arise

Table 2. Dataset details and algorithm runtimes for edge cardinality budget $k = 150$. All runtimes reported in seconds. Our D-optimality-aware front-end is able to quickly select a tiny fraction of all available candidate edges for loop-closure verification. This leads to a sparse pose graph that can be optimized efficiently via solvers such as g2o. "Valid Edges" refers to actual loop closures, while "Total Edges" refers to all *potential* loop-closure candidates.

Dataset	Poses	Valid Edges	Total Edges	% Edges Checked	WST Runtime (s)	D-Crit. Runtime (s)	WST Batch Runtime (s)	D-Crit. Batch Runtime (s)	Full g2o Iter. Time (s)	WST g2o Iter. Time (s)
KITTI 00	2049	300	514	29	0.89	9.36	0.38	2.20	0.0023	0.0025
Atlas1K	1000	2459	6094	2.5	11.32	28.18	0.98	4.12	0.0079	0.0013
Atlas2K	2000	10,517	26,013	0.6	120.48	230.13	4.84	17.26	0.44	0.0027
Atlas3K	3000	14,596	36,476	0.4	211.96	412.29	6.25	20.78	0.93	0.0041
Atlas4K	4000	21,730	53,935	0.3	474.91	859.69	8.89	27.76	7.40	0.0057
Atlas5K	5000	27,006	66,371	0.2	712.80	1317.2	10.67	32.65	7.66	0.0069

in a multi-agent scenario, the dataset was partitioned into fifths, with each fifth representing a distinct robot's trajectory and measurements. The candidate edge set is restricted to those that are incident on poses associated with two distinct robots. The 2D pose constraints for odometry and loop-closure candidates were generated using ORB-SLAM (Mur-Artal et al., 2015). Stereo image pairs from the KITTI dataset were used to generate 3D odometry and loop closures which were projected into a 2D plane. The loop-closure candidate poses and edges were discovered and assigned probabilities using the version of DBoW2 included in ORB-SLAM. Candidates with normalized score less than a threshold of 0.2 were omitted in order to prefilter outliers and unlikely matches.

To evaluate the performance of our proposed methods on problem instances with a large number of poses and candidate edges, we created a series of synthetic, Manhattan-like grid world datasets, which we will refer to as Atlas datasets, using the 2D simulator included in g2o (Kümmerle et al., 2011). The odometry and loop closures generated by the simulator are corrupted with additive zero-mean Gaussian noise with $\sigma_\theta^{-2} = 5000$ and $\sigma_p^{-2} = 500$. The simulator includes loop closures between poses within a range of 5 meters and angular field of view of 180°. The overlapping fraction of poses' fields of view were used to generate (exact) probabilities for candidate edges in the k-RESP problem, which were then assigned as valid or invalid by random sampling. Only the edges that were randomly sampled as valid are available for solving the resulting pose graph if they are selected; the remainder are treated as false positives. Thus, the measurement noise and probabilities used in our SLAM front-end experiments are exact for the Atlas dataset, providing an idealized scenario with ground truth in which to test our proposed algorithm on large-scale problems in Section 9.3. In order to keep the number of candidate edges manageable, the maximum degree for any vertex was limited to 40 via an initial random pruning of edges. The dense graphs studied here could arise in scenarios where one or more robots frequently revisit certain areas where they are able to establish loop closures. Limiting this density by selecting an informative subset of candidate edges is a means of maintaining high localization and mapping accuracy while controlling resource consumption; see Section 8.



(a) KITTI 00 (b) Atlas Simulation

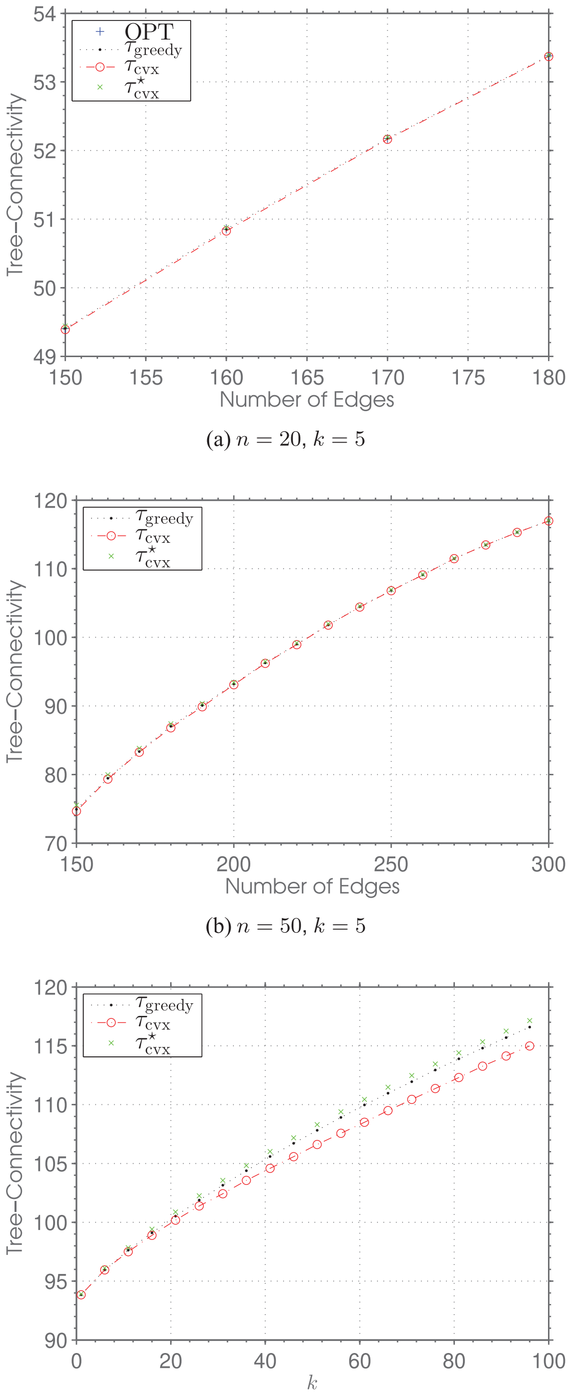
Fig. 11. Left: KITTI odometry sequence 00. Right: 2D Atlas simulation (5000 poses). Each trajectory is partitioned into five sub-trajectories representing different robots.

Figure 11(b) displays the largest Atlas dataset created, which contains 5000 poses. The same strategy of partitioning the datasets into fifths used on the KITTI dataset was used, providing a large number of inter-robot loop closures. Table 2 summarizes the number of poses and loop closures in the KITTI and Atlas datasets.

9.1. Near-t-optimal graph synthesis: randomly generated graphs

We conducted a series of experiments on randomly generated graphs to evaluate the performance of the proposed approximation algorithms. In each experiment, a random base graph was generated with $n \in \{20, 50\}$ vertices. In these experiments, the set of candidate edges is $\mathcal{C} = \mathcal{E}(\mathcal{K}_n) \backslash \mathcal{E}_{\text{init}}$. Figure 18 illustrates the performance of our approximate solutions for k-ESP.

In the first experiment, we kept $k = 5$ fixed and gradually increased the number of edges in the base graph $|\mathcal{E}_{\text{init}}|$. Note that this is equivalent to gradually reducing the



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

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

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

Fig. 12. Problem k-ESP on randomly generated graphs.

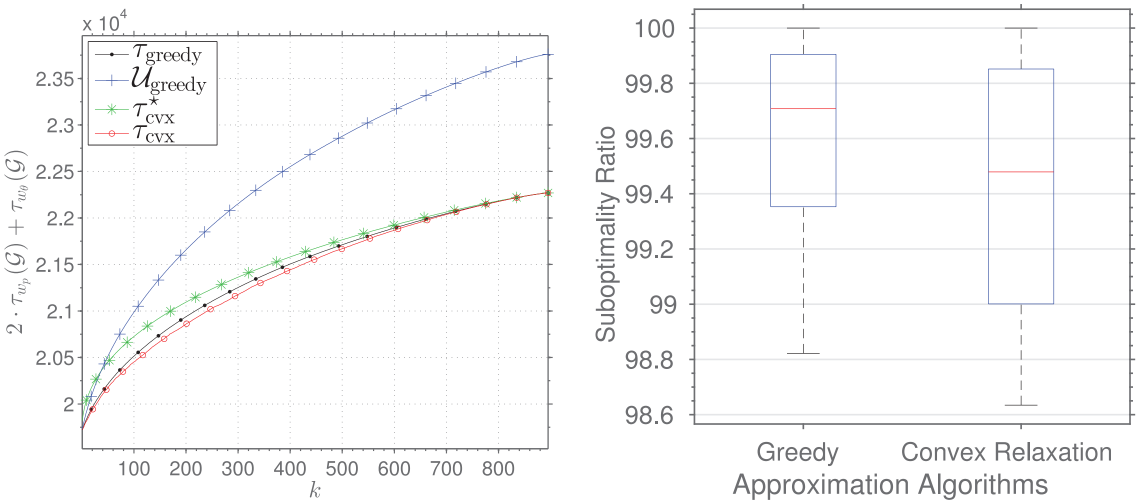
cardinality of the candidate set $\mathcal{C}$. We then ran the proposed greedy and convex approximation algorithms, and recorded the tree connectivity of the resulting graphs. For $n = 20$, we also found OPT via exhaustive search. In practice, finding OPT is only feasible in small graphs. The results are depicted in Figures 12(a) and (b). In these figures, τ_{greedy} and τ_{cvx} denote the tree connectivity achieved by the greedy and convex relaxation followed by deterministic rounding, respectively. We also report the optimal value of the convex program (before rounding) $\tau^\star_{\text{cvx}}$, which is an upper bound on OPT; see Corollary 9. Hence, although OPT is not available in the case of $n = 50$, we can still assess the proposed approximate algorithms using $\tau^\star_{\text{cvx}}$. As we can see from these figures, both algorithms are able to design graphs with a value close to OPT.

In the second experiment, we kept the base graph fixed with $n = 50$ vertices and $|\mathcal{E}_{\text{init}}| = 200$ edges. The number of candidate edges is thus $c = 1025$. We then gradually increased the value of k from 1 to 100. Figure 12(c) shows the results. Unlike the previous figures, here we can see that the optimality gap for the convex relaxation algorithm becomes noticeable as k increases. According to this figure, the greedy algorithm begins to significantly outperform the convex relaxation once $k \geqslant 20$ and attains near-t-optimal designs as certified by $\tau^\star_{\text{cvx}}$.

9.2. Near-D-optimal measurement selection and graph pruning in SLAM

Now we evaluate the graphical measurement selection scheme proposed in Section 8.3 using the KITTI, Atlas, and Intel Research Lab datasets. The original Intel Research Lab dataset was collected by Hähnel (2003). We extracted the graph from the preprocessed version provided as part of g2o (Kümmerle et al., 2011). In our experiments, the base graph consists of the $n = 943$ poses and $|\mathcal{E}_{\text{init}}| = 942$ odometry edges. The candidate set $\mathcal{C}$ is the set of loop closures with $c = 895$ edges extracted from the original dataset. For the edge weights, we use the information matrices provided in the g2o 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. Our goal is to pick k loop closures for varying values of k such that $\ell_\tau(\mathcal{G}) = 2\tau_{w_p}(\mathcal{G}) + \tau_{w_\theta}(\mathcal{G})$ is maximized (i.e., our graphical surrogate for the D-criterion). Computing the true OPT through exhaustive search is not practical in this dataset.

Figure 13 shows the results for different values of $1 \leqslant k \leqslant c$. In Figure 13(a) we see the values obtained by running the greedy algorithm (τ_{greedy}), convex relaxation before and after rounding ($\tau^\star_{\text{cvx}}$ and τ_{cvx}, respectively), and the upper bound on OPT based on the approximation factor proved for the greedy algorithm ($\mathcal{U}_{\text{greedy}}$); see Corollary 9. While both approximation algorithms generally perform well, once again we can see that the greedy algorithm outperforms convex relaxation with deterministic rounding, especially for $80 \leqslant k \leqslant 560$. Based on Corollary 9, OPT is bounded from above by $\mathcal{U} \triangleq \min\{\mathcal{U}_{\text{greedy}}, \tau^\star_{\text{cvx}}\}$. For small values of k, the tightest upper bound on OPT is given by $\mathcal{U} = \mathcal{U}_{\text{greedy}}$ (blue curve). However, for $k \geqslant 60$, the convex relaxation provides a significantly tighter upper bound on OPT (green curve). Figures 12 and 13 suggest that the greedy algorithm can find provably near-optimal designs, while the fractional solution of the relaxed program can provide even stronger (albeit *a posteriori*) certificates for



(a) (b)

Fig. 13. Problem k-ESP for pose-graph SLAM in the Intel Research Lab dataset. (a) The performance of the proposed approximation algorithms, as well as two upper bounds on OPT for varying k. (b) A summary of the distribution of a lower bound on the suboptimality ratio (%) computed using the tightest upper bound $\mathcal{U} = \min\{\tau^{\star}_{\text{cvx}}, \mathcal{U}_{\text{greedy}}\}$ on OPT for each approximation algorithm.

the near-optimality of the greedy design. Figure 13(b) displays the distribution of a conservative estimate of the suboptimality ratios (%) of the proposed approximation algorithms defined as $\tau_{\text{cvx}}/\mathcal{U} \times 100$ and $\tau_{\text{greedy}}/\mathcal{U} \times 100$.

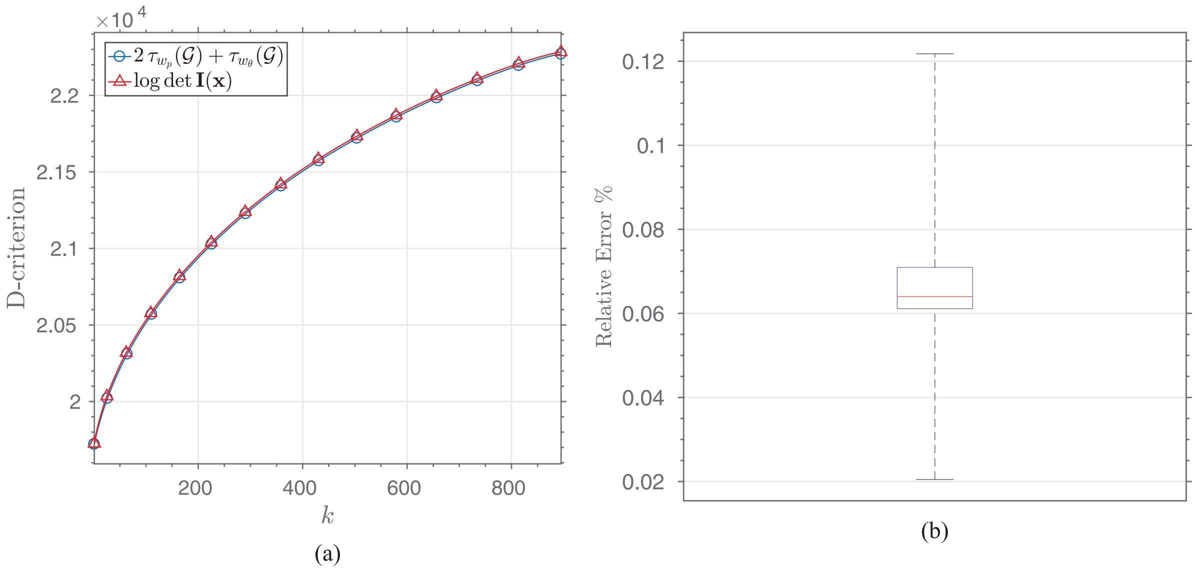
Recall that in Section 3.4 experimental results were presented to validate our graphical approximation of the D-criterion. Now in Figure 14 we assess the accuracy of this approximation in the context of measurement selection. Figure 14(a) shows the D-criterion for the loop closures selected by the greedy algorithm in Figure 13. We have used g2o (Kümmerle et al., 2011) to solve each problem (for $1 \leqslant k \leqslant 895$). As expected, the true value is consistently close to $\ell_\tau(\mathcal{G}) = 2\tau_{w_p}(\mathcal{G}) + \tau_{w_\theta}(\mathcal{G})$. The box plot in Figure 14(b) shows a summary of the distribution of RE (%) defined in (39).

Figure 15 illustrates the pose graphs synthesized by the greedy algorithm for six different values of k in the Intel Research Lab dataset. We have used the MLE based on the original dataset to visualize the trajectory of the robot. The objective value attained by the greedy algorithm and an upper bound for OPT are provided above each figure. Figure 15(f) shows the original dataset with 895 loop closures drawn in blue.

Given a fixed linearization point, the D-criterion (after normalization) can be expressed as a function of selected loop closures,

$$f_{\text{D}} : \mathcal{E} \mapsto \log\det(\mathbb{I}_{\text{init}} + \sum_{e \in \mathcal{E}} \mathbb{I}_e) - \log\det \mathbb{I}_{\text{init}} \tag{70}$$

where $\mathbb{I}_{\text{init}} \succ \mathbf{0}$ and $\mathbb{I}_e$ are the information matrices associated with the base graph and loop closure e, respectively. Here f_{D} is normalized, monotone, and submodular (Shamaiah et al., 2010). Therefore, a near-optimal solution for the problem of maximizing f_{D} subject to a cardinality constraint on $\mathcal{E}$ can be found by the greedy algorithm (Nemhauser et al., 1978; Shamaiah et al., 2010). We ran the greedy algorithm once for our graphical surrogate function based on the WST, and once for f_{D}. The linearization point used for evaluating f_{D} is the MLE based on the edges in the base (odometry) graph. We then took the measurements selected by each algorithm, and evaluated their resulting D-criterion using the best available linearization point (ground truth for synthetic data, and the MLE using all loop closures for KITTI). Figures 16 and 17 display the D-criterion achieved by each algorithm (namely, greedy WST and greedy *observed* D-optimality) and their runtimes in the KITTI and Atlas datasets. The lazy evaluation method (Krause and Golovin, 2012; Minoux, 1978) was used to speed up the computation time of both greedy algorithms. In order to make the runtime of both algorithms tractable for datasets containing tens of thousands of candidate loop closures, the full problem solution (i.e., selecting k edges out of *all* c candidate edges) was compared with an approach that randomly partitioned the candidate edges into multiple smaller batches. For the KITTI dataset, from each batch of 25 candidates, 10 edges were selected. Similarly, for the Atlas5K dataset, 10 edges were selected for each random batch of 1300 candidates. These batch sizes and sampling rate of 10 edges per batch were chosen so that the maximum edge selection budget k used in our experiments (see the x-axis of Figures 16 and 17) was spent after all batches were examined, enabling a comparison with the full problem solution. This batch approach additionally serves the purpose of simulating a scenario where



(a)

(b)

Fig. 14. Greedy *k*-ESP on the Intel Research Lab Dataset. In (a), the horizontal axis shows the number of loop-closure edges selected by the greedy algorithm. The base graph here is the odometry-only subgraph. The red curve (marked by triangles) shows the value of the log-determinant of the information matrix at the MLE. The blue curve (marked by circles) shows the approximated D-criterion computed only using the graphical structure of the problem. The box plot in (b) gives a summary of the relative error (%) defined in (39) between the 895 data points shown in (a).

intermittent availability of communication and/or the sequential acquisition of measurements produces smaller problem instances. We also include results for a baseline edge selection strategy that simply chooses a completely random subset of k edges.

According to Figure 16(a), greedy edge selection based on the proposed objective function (WST) performs nearly as well as picking edges greedily based on the observed D-criterion, once again indicating that WST provides a suitable proxy for D-optimal measurement selection. For both objective functions, the batch approach incurs a slight performance penalty, but is still superior to the random baseline. Figure 16(b) demonstrates that WST is cheaper to compute and, thus, runs significantly faster than the D-criterion for both full and batch algorithm variants. Even on a relatively small dataset such as KITTI, the batch approach provides significant runtime savings for both objective functions.

Since the Atlas datasets are much larger and denser than the KITTI dataset, the runtime savings in Figure 17(b) associated with the batch approach are much more significant. Computing the WST objective function is also significantly faster than computing the D-criterion for both full and batch approaches. Figure 17(a) demonstrates that the batch approach does not reduce solution quality significantly, and that the WST is once again a meaningful surrogate for D-criterion maximization.

9.3. D-optimality-aware SLAM front-end

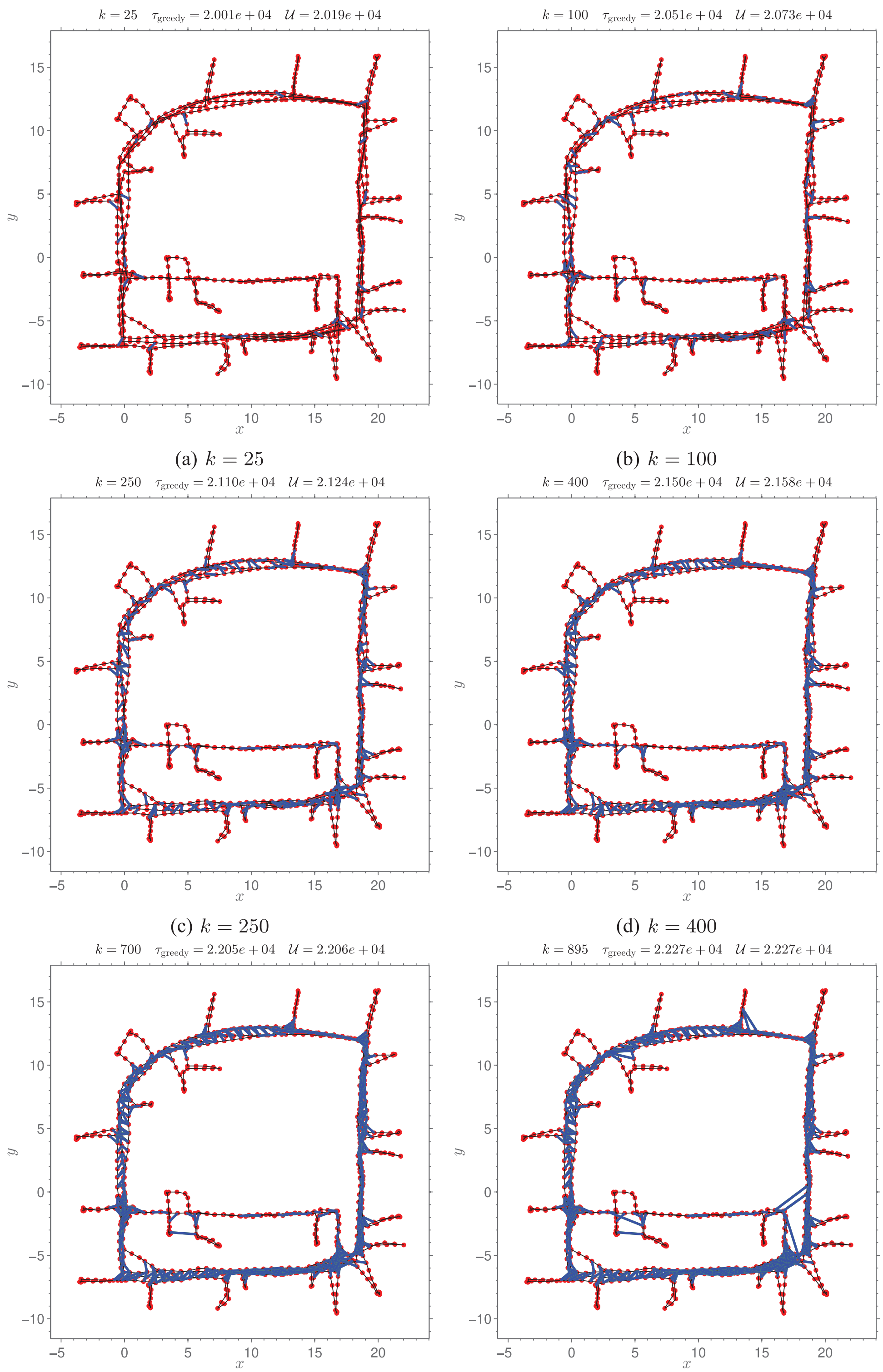
In this section, we evaluate the proposed D-optimality-aware SLAM front-end of Section 8.4 using both the KITTI and Atlas datasets. Owing to the random nature of *potential* loop closures in this setting, we compare the performance of the greedy algorithm based on expected WST (68) with the greedy algorithm applied to an objective function proposed by Carlone and Karaman (2017):

$$\tilde{f}_{\mathrm{D}} : \mathcal{E} \mapsto \log\det\big(\mathbb{I}_{\mathrm{init}} + \sum_{e\in\mathcal{E}} p_e \mathbb{I}_e\big) - \log\det \mathbb{I}_{\mathrm{init}} \qquad (71)$$

in which p_e is the occurrence probability of potential loop closure e. We took the *valid* (i.e., true) loop closures selected by each algorithm, and evaluated their resulting D-criterion using the best available linearization point (ground truth for synthetic data, and the MLE using all loop closures for KITTI). This was also done for a random potential measurement selection baseline.

Figure 18(a) displays the results for the KITTI dataset. In spite of the fact that the D-criterion greedy algorithm is aiming to explicitly maximize the D-criterion, our WST objective function provides comparable results. Furthermore, the timing results in Figure 18(b) and Table 2 indicate that the WST function is nearly an order of magnitude faster on this dataset. The batch algorithms also retain performance similar to the full greedy solutions while requiring less runtime. In this dataset, batches of 25 candidate edges were formed, and 10 edges were greedily selected from each batch.

Note that the D-criterion improvement from using a D-optimality-aware front-end strategy over the random baseline in Figure 18(a) is greater than the improvement in the back-end case of Figure 16(a). This is because the back-end task has no invalid loop closures (i.e., potential candidates that did not correspond to true loop closures), whereas



(a) $k = 25$

(b) $k = 100$

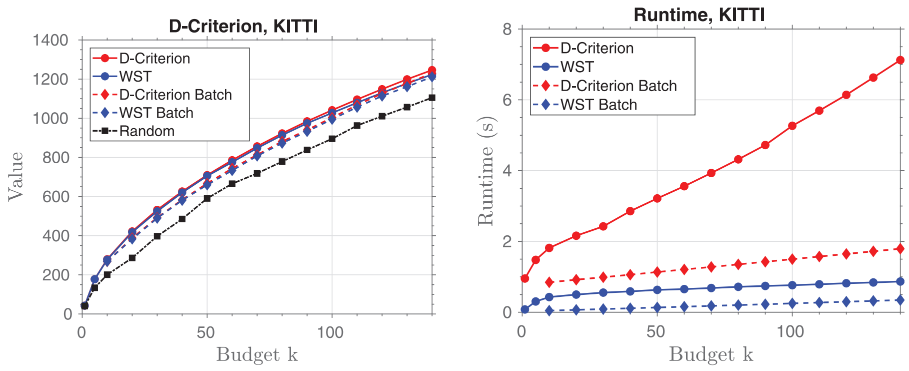
(c) $k = 250$

(d) $k = 400$

(e) $k = 700$

(f) $k = 895$

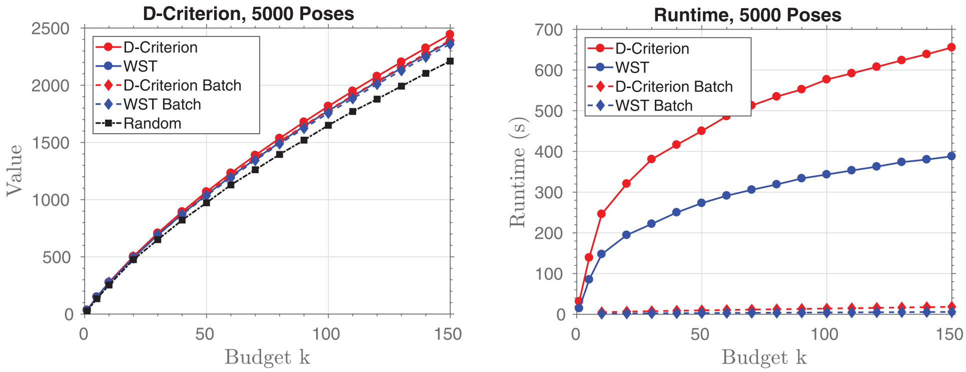
Fig. 15. Greedy design for k loop closures (out of 895). Loop-closure edges are shown in blue, and odometry measurements are shown in black. Red circles are the robot poses. See https://youtu.be/5JZF2QiRbDE for the complete evolution of pose graphs designed by the greedy algorithm.



(a) KITTI D-Criterion, ESP

(b) KITTI Runtime, ESP

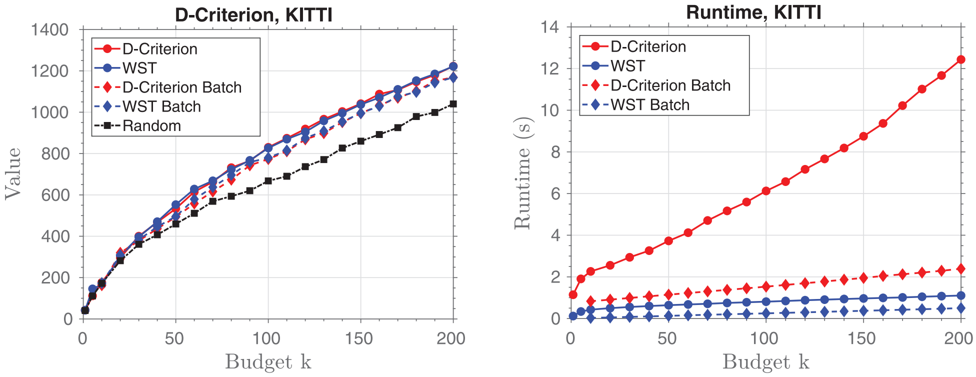
Fig. 16. Comparison of D-criterion (a) and runtime (b) for the full and batch variants on the back-end *k*-ESP scenario (only valid edges as candidates) with the KITTI dataset.



(a) Atlas5K D-Criterion, *k*-ESP

(b) Atlas5K Runtime, *k*-ESP

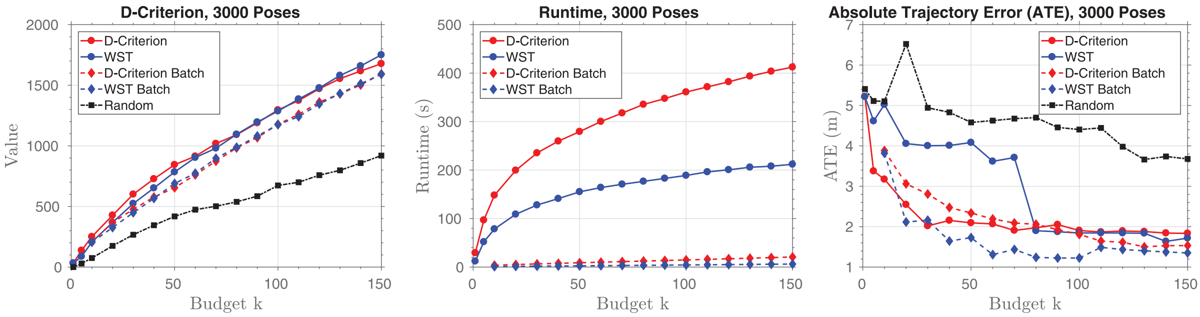
Fig. 17. Comparison of D-criterion (a) and runtime (b) for the full and batch variants on the back-end *k*-ESP scenario (only valid edges as candidates) on the *Atlas5K* dataset.



(a) KITTI D-Criterion, *k*-RESP.

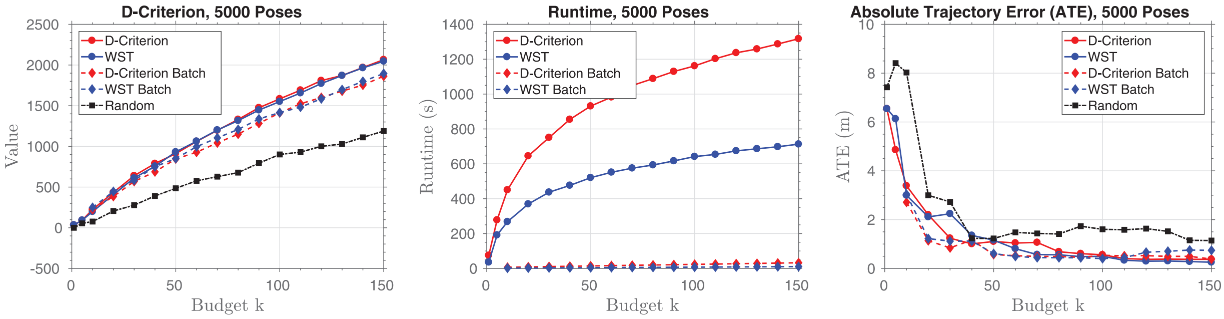
(b) KITTI Runtime, *k*-RESP.

Fig. 18. Comparison of D-criterion (a) and runtime (b) for the full and batch variants of WST and D-criterion greedy algorithms for the front-end SLAM scenario on the KITTI dataset.



(a) D-criterion, RESP, Atlas3K (b) Runtime, RESP, Atlas3K (c) ATE, RESP, Atlas3K

Fig. 19. Comparison of full and batch greedy strategies on the Atlas3K simulated dataset for the D-optimality-aware front-end task. In the batch algorithms, 10 edges were selected from random batches of 1300.



(a) D-criterion, RESP, Atlas5K (b) Runtime, RESP, Atlas5K (c) ATE, RESP, Atlas5K

Fig. 20. Comparison of full and batch greedy strategies on the *Atlas5K* simulated dataset for the D-optimality-aware SLAM front-end task. In the batch algorithms, 10 edges were selected from random batches of 1300.

in the front-end task the greedy algorithms take the edge probability into account when selecting edges, while the random baseline selects a number of invalid potential loop closures.

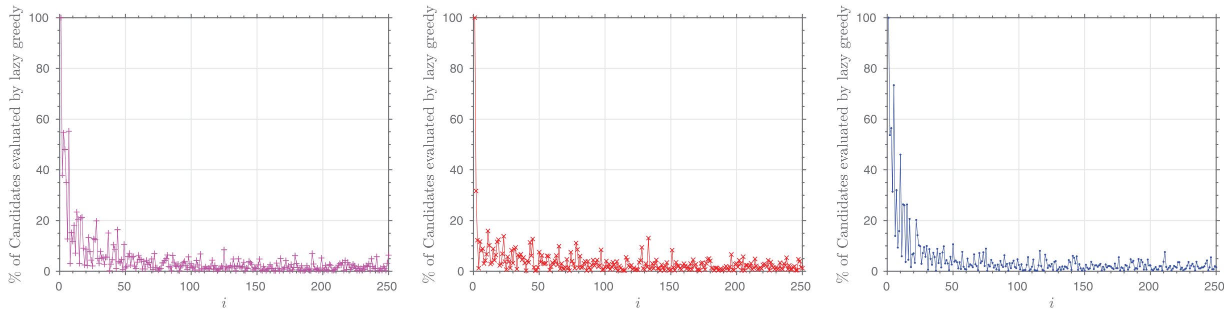
Figures 19 and 20 display the performance of the various algorithms on two of the Atlas datasets. Since ground truth is available, the absolute trajectory error (ATE) is presented in addition to the D-criterion with respect to the ground-truth trajectory. In Figures 19(a) and 20(a), the WST objective function performs similarly to the observed D-criterion in terms of the (true; i.e., evaluated at the ground truth) D-criterion, while only requiring a fraction of the runtime (see Section 9.4 for further details). The same is true in terms of the ATE in Figures 19(c) and 20(c). In the batch variant, a mere 10 edges were selected for each random batch of 1300 candidates. The slight performance decrease caused by partitioning the data into batches is acceptable, especially considering the greatly reduced runtimes reported in Figures 20(b) and 19(b).

Similar to the KITTI data, the performance improvement in Figures 19 and 20 stemming from use of the greedy algorithms over the random baseline is larger in the k-RESP-based probabilistic SLAM front-end case than the back-end experiments in Figure 17(a). Visual inspection indicates that the improvement for the Atlas datasets is even greater, owing to the fact that the synthetic probabilities are exact, as opposed to the normalized visual similarity scores used for KITTI. This allows greater accuracy in computing the expected gain for candidate edges when running the greedy algorithms.

9.4. Runtime

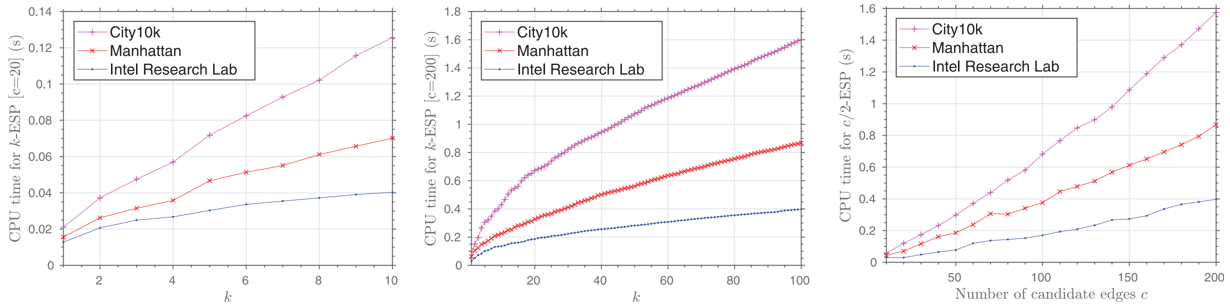
Solving the convex relaxation of k-ESP using YALMIP and SDPT3 scales poorly as the number of poses/vertices n increases. In the Intel Research Lab dataset ($n = 943$), the runtime is between 10 and 30 seconds for different values of c. For $n = 3500$ (Manhattan dataset) and $n = 10,000$ (City10K), the runtime is $\gg$ 10 minutes.[11] To improve the scalability of this scheme, one needs to leverage the structures that are not being exploited by general-purpose tools such as YALMIP and SDPT3. In practice, the convex relaxation approach is only useful for certifying the quality of other designs (e.g., greedy), which is usually done offline.

Figure 22 shows the runtimes for three benchmark pose-graph datasets, namely Intel Research Lab, Manhattan, and City10K with, respectively, 943, 3500, and 10,000 poses. The greedy algorithm is implemented in MATLAB, does not exploit rank-one updates of the Cholesky factor (Algorithm 4), and uses lazy evaluations (Krause and Golovin, 2012; Minoux, 1978) to speed up the algorithm. The impact of this improvement is illustrated in Figure 21.



(a) City10K (b) Manhattan (c) Intel Research Lab

Fig. 21. The naïve greedy algorithm needs to evaluate $c - i + 1$ edges in its ith round. Lazy evaluations (Krause and Golovin, 2012; Minoux, 1978) can reduce this number significantly. This figure shows the percentage of remaining edges in each round evaluated by the lazy greedy algorithm for $c = 500$ and $k = 250$.



(a) (b) (c)

Fig. 22. CPU time (s) for (a) $c = 20$, (b) $c = 200$, and (c) $c/2$-ESP using the proposed greedy algorithm.

According to this figure, after the first few rounds, the lazy greedy algorithm can select the best candidate by just evaluating less than 10% of the remaining candidates.

The impact of the number of poses n on the runtime of the greedy algorithm can be seen in Figure 22. As expected, the runtime of the greedy algorithm scales much better with n than that of the proposed convex relaxation scheme. Figure 22(a) and (b) show the runtime of greedy for $c = 20$ and 200, respectively. Figure 22(c) shows the runtime of greedy in $c/2$-ESP as a function of c. Note that in online applications, one can control the runtime of greedy by controlling the number of candidate edges c (i.e., batch size in the "batch" experiments presented earlier). Therefore, by periodically pruning measurements one can control the runtime of greedy measurement selection. Needless to say, more informative edges will be selected if we expand the candidate set.

Table 2 summarizes the runtime of the k-RESP algorithms used in the D-optimality-aware front-end experiments. The table displays full and batch greedy runtimes for both D-criterion and WST objective functions after each has selected 150 edges. The percentage of edges checked is also displayed, as this influences the fraction of candidate sensor registrations that would be attempted, further affecting the runtime. WST is consistently far faster than the D-criterion in both the full and batch algorithm variants. The reduction in g2o iteration time incurred by the full WST selection is also given, demonstrating further computation savings provided by maintaining sparsity through subselection of candidate edges.

In addition to optimizing our current implementation, we can speed up the greedy algorithm by taking advantage of two key structures.

1. In our current implementation, we factorize the reduced Laplacian from scratch in each round of the greedy algorithm. As proposed earlier (Algorithm 4), it would be more efficient if we instead update the Cholesky factor recursively via rank-one updates. Unfortunately, cholupdate in MATLAB does not support sparse matrices. This functionality, however, is provided by CHOLMOD (Chen et al., 2008) and CSparse (Davis, 2006) and can be utilized in a C/C++; implementation to further speed up the greedy algorithm.
2. In its ith round, the greedy algorithm solves c_i linear systems, with a single triangular coefficient matrix and c_i different right-hand sides where $c_i = c - i + 1$.

These triangular linear systems can be solved in parallel on multiple CPU/GPU cores.[12] Exploiting this *embarrassingly parallel* structure of the greedy algorithm would dramatically reduce the runtime.

10. Conclusion and future work

In this paper, we have explored and exploited the overlooked interplay between the graphical and estimation-theoretic facets of SLAM as an EoG problem. First, a theoretical explanation was presented for the empirical observations made by Olson and Kaess (2009) regarding the impact of average degree on overfitting. We then highlighted the intrinsic connection between the FIM and the Laplacian matrix in several EoG problems, including SLAM. This observation paved the way to reveal intimate connections between different notions of graph connectivity and estimation reliability. Among them, we focused particularly on the relation between the WST and the D-criterion. Our theoretical analysis was empirically validated through extensive experimental evaluation based on publicly available real and synthetic SLAM datasets. This key insight makes it possible to reason about the D-criterion in SLAM based *only* on the weighted tree connectivity of the graph underneath. Consequently, we studied the combinatorial optimization problem of designing sparse t-optimal graphs. Characterizing t-optimal graphs, even in simpler settings, remains an open problem and, to the best of the authors' knowledge, no efficient algorithm for synthesizing such graphs in the general case is known. We instead presented a complementary pair of efficient approximation algorithms with provable guarantees and near-optimality certificates. This was achieved by establishing a number of new theoretical results, including log-submodularity of the WST as a function of graph edges. To the best the authors' knowledge, this result is new in graph theory. We then leveraged this structure to show that the greedy algorithm is a constant-factor approximation algorithm based on the seminal work of Nemhauser et al. (1978). Our second approximation algorithm is based on a convex relaxation approach for sensor selection proposed by Joshi and Boyd (2009). We provided new intuitive explanations for this relaxation and a natural rounding scheme. Our approximation algorithms were extended to two additional settings, namely ARGs and a dual narrative. We then applied our algorithmic framework to measurement selection problems in SLAM. The performance of the proposed framework was extensively evaluated using randomly generated graphs, realistic simulations, and real benchmark datasets. It was demonstrated that the proposed graphical approach exhibits comparable performance to D-optimal designs, while being significantly faster and robust to convergence/linearization errors. Our empirical observations indicate that in almost all cases, the greedy algorithm can find better approximate solutions and is faster than the proposed convex relaxation scheme. However, the latter still plays a crucial role in our framework by providing better near-optimality certificates which are used to certify the quality of greedy solutions.

The near-t-optimal graph synthesis framework presented in this work can be readily used in many other domains where maximizing the number of spanning trees is desired (often as a measure of network robustness). For instance, a well-known result in network reliability theory indicates that, in the so-called all-terminal reliability model, if a uniformly most reliable network exits, it must have the maximum tree connectivity among all graphs with the same size (Bauer et al., 1987; Boesch et al., 2009; Myrvold, 1996). Moreover, Cheng (1981) noted that D-optimal incomplete block designs in statistics are associated with t-optimal concurrence graphs (see Bailey and Cameron, 2009: for a survey). See also Gutman et al. (1983), Brown et al. (1996), and Kim et al. (2013) for applications in chemistry and RNA modeling.

We plan to investigate an intriguing overlap between the parameters that emerged from Theorem 3 and those that emerged from a convergence analysis due to Carlone (2013) as part of our future work. Preliminary results suggest that this theorem can be extended to 3D SLAM. This extension will be examined in future work. Improving the computational complexity of the greedy algorithm is another avenue for future work. Our approach has been recently adopted by Li et al. (2018) who proposed a faster greedy algorithm with an approximation ratio of $(1 - 1/e - \epsilon)$ for any positive ϵ. Finally, we are ultimately interested in a seamless incorporation of graph topology into planning and decision-making pipelines beyond measurement selection.

Acknowledgments

The authors would like to thank Kümmerle et al. (2011) for g2o, as well as the Robotics Data Set Repository (Radish) (Howard and Roy, 2003), C. Stachniss, D. Hähnel, E. Olson, M. Kaess, Y. Latif, and N. Sünderhauf for original and preprocessed datasets that were used in this work. The authors also gratefully acknowledge Yulun Tian for assistance with experiments.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was supported in part by the Australian Government through the Australian Research Council's Discovery Projects funding scheme (project DP120102786), by the NASA Convergent Aeronautics Solutions project Design Environment for Novel Vertical Lift Vehicles (DELIVER), and by ONR under MURI program award N000141110688 and BRC award N000141712072.

Notes

1. Extending this model and our following results to the 2D feature-based SLAM problem is straightforward.
2. Note that here we do not make any other assumptions regarding the structure of the noise covariance matrix $\mathbf{\Sigma}$.
3. Note that parallel edges can be replaced by a single edge weighted by the sum of the weights.

4. Note that **p** should not be confused with the notation we used to refer to the position of a robot throughout its trajectory. We use **p** instead of π to distinguish between the integer and fractional values.
5. For now, we ignore the fact that the rounded solution may violate the cardinality constraint.
6. In most cases, we can verify all k potential loop closures using fewer than k data exchanges. The problem of inter-robot loop-closure detection under communication constraints is studied in (Tian et al., 2018).
7. Linearization may cause other problems, see Section 8.5.
8. In the SLAM literature, these terms mean that the pruned covariance matrix is "greater" than the original covariance matrix (Loewner order), i.e., we do not underestimate belief uncertainty.
9. It is important to note that this case (the expression in Theorem 3) is different from the case in which there are multiple (parallel) edges (with different weights) between two vertices. In the latter case, we can simply replace parallel edges with a single edge whose weight is equal to the sum of parallel edge weights.
10. The presented approximate flop counts are derived for the worst case of dense matrices; counting flops for sparse pose graphs is too complex as it depends on the sparsity pattern, heuristic reordering, etc.
11. We stopped SDPT3 after this time.
12. This idea can be used alongside lazy evaluation (Minoux, 1978) by guessing the number of remaining candidates that are needed to be evaluated based on empirical observations, and solving the corresponding triangular linear systems in parallel.

References

Bailey RA and Cameron PJ (2009) Combinatorics of optimal designs. *Surveys in Combinatorics* 365: 19–73.

Bailey T and Durrant-Whyte H (2006) Simultaneous localization and mapping (slam): Part II. *IEEE Robotics and Automation Magazine* 13(3): 108–117.

Barooah P and Hespanha JP (2007) Estimation on graphs from relative measurements. *IEEE Control Systems* 27(4): 57–74.

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

Boesch FT, Satyanarayana A and Suffel CL (2009) A survey of some network reliability analysis and synthesis results. *Networks* 54(2): 99–107.

Boyd S and Vandenberghe L (2004) *Convex optimization*. Cambridge: Cambridge University Press.

Brown TJ, Mallion RB, Pollak P and Roth A (1996) Some methods for counting the spanning trees in labelled molecular graphs, examined in relation to certain fullerenes. *Discrete Applied Mathematics* 67(1): 51–66.

Cadena C, Carlone L, Carrillo H, et al. (2016) Simultaneous localization and mapping: Present, future, and the robust-perception age. *Preprint arXiv:1606.05830.*

Carlevaris-Bianco N, Kaess M and Eustice RM (2014) Generic node removal for factor-graph SLAM. *IEEE Transactions on Robotics* 30(6): 1371–1385.

Carlone L (2013) Convergence analysis of pose graph optimization via gauss-newton methods. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).*

Carlone L, Aragues R, Castellanos JA and Bona B (2014) A fast and accurate approximation for planar pose graph optimization. *The International Journal of Robotics Research* 33: 965–987.

Carlone L and Censi A (2014) From angular manifolds to the integer lattice: Guaranteed orientation estimation with application to pose graph optimization. *IEEE Transactions on Robotics* 30(2): 475–492.

Carlone L and Karaman S (2017) Attention and anticipation in fast visual-inertial navigation. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 3886–3893.

Chen Y, Davis TA, Hager WW and Rajamanickam S (2008) Algorithm 887: Cholmod, supernodal sparse Cholesky factorization and update/downdate. *ACM Transactions on Mathematical Software* 35(3): 22.

Cheng CS (1981) 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): 240–248.

Chli M and Davison AJ (2009) Active matching for visual tracking. *Robotics and Autonomous Systems* 57(12): 1173–1187.

Choudhary S, Indelman V, Christensen HI and Dellaert F (2015) Information-based reduced landmark SLAM. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 4620–4627.

Cieslewski T and Scaramuzza D (2017) Efficient decentralized visual place recognition using a distributed inverted index. *IEEE Robotics and Automation Letters* 2(2): 640–647.

Cohen JE (1986) Connectivity of finite anisotropic random graphs and directed graphs. *Mathematical Proceedings of the Cambridge Philosophical Society* 99: 315–330.

Davis TA (2006) *Direct methods for sparse linear systems*, volume 2. Philadelphia, PA: SIAM.

Davison AJ (2005) Active search for real-time vision. In: *Tenth IEEE International Conference on Computer Vision, 2005 (ICCV 2005)*, volume 1. IEEE, pp. 66–73.

Dellaert F and Kaess M (2006) Square root SAM: Simultaneous localization and mapping via square root information smoothing. *The International Journal of Robotics Research* 25(12): 1181–1203.

Dong J, Nelson E, Indelman V, Michael N and Dellaert F (2015) Distributed real-time cooperative localization and mapping using an uncertainty-aware expectation maximization approach. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 5807–5814.

Duckett T, Marsland S and Shapiro J (2000) Learning globally consistent maps by relaxation. In: *Proceedings ICRA'00: IEEE International Conference on Robotics and Automation, 2000*, volume 4. IEEE, pp. 3841–3846.

Duckett T, Marsland S and Shapiro J (2002) Fast, on-line learning of globally consistent maps. *Autonomous Robots* 12(3): 287–300.

Eade E, Fong P and Munich ME (2010) Monocular graph SLAM with complexity reduction. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 3017–3024.

Frey KM, Steiner TJ and How JP (2018) Complexity analysis and efficient measurement selection primitives for high-rate graph

SLAM. In: *IEEE International Conference on Robotics and Automation (ICRA)*. https://arxiv.org/abs/1709.06821.

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

Gálvez-López D and Tardos JD (2012) Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics* 28(5): 1188–1197.

Geiger A, Lenz P, Stiller C and Urtasun R (2013) Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research* 32(11): 1231–1237.

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

Giamou M, Khosoussi K and How JP (2018) Talk resource-efficiently to me: Optimal communication planning for distributed loop closure detection. In: *IEEE International Conference on Robotics and Automation (ICRA)*. https://arxiv.org/abs/1709.06675 .

Gilbert EN (1959) Random graphs. *The Annals of Mathematical Statistics* 30(4): 1141–1144.

Godsil C and Royle G (2001) *Algebraic graph theory* (*Graduate Texts in Mathematics Series*). London: Springer.

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

Hähnel D (2003) Intel Research Lab dataset. https://svn.openslam.org/data/svn/g2o/trunk/data/2d/intel/intel.g2o. (accessed 10 May 2017).

Hochbaum DS (1996) *Approximation algorithms for NP-hard problems*. PWS Publishing Co.

Horn RA and Johnson CR (1990) *Matrix analysis*. Cambridge: Cambridge University Press.

Howard A and Roy N (2003) The robotics data set repository (radish). http://radish.sourceforge.net/ .

Huang G, Kaess M and Leonard JJ (2013) Consistent sparsification for graph optimization. In: *2013 European Conference on Mobile Robots (ECMR)*. IEEE, pp. 150–157.

Ila V, Porta JM and Andrade-Cetto J (2010) Information-based compact pose SLAM. *IEEE Transactions on Robotics* 26(1): 78–93.

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

Kaess M (2008) *Incremental Smoothing and Mapping*. Ph.D. Thesis, Georgia Institute of Technology.

Kaess M and Dellaert F (2009) Covariance recovery from a square root information matrix for data association. *Robotics and Autonomous Systems* 57(12): 1198–1210.

Kaess M, Ranganathan A and Dellaert F (2008) iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics* 24(6): 1365–1378.

Kelmans AK (1996) On graphs with the maximum number of spanning trees. *Random Structures and Algorithms* 9(1-2): 177–192.

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

Khosoussi K, Huang S and Dissanayake G (2016a) Tree-connectivity: Evaluating the graphical structure of SLAM. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1316–1322.

Khosoussi K, Sukhatme GS, Huang S and Dissanayake G (2016b) Designing sparse reliable pose-graph SLAM: A graph-theoretic approach. In: *2016 International Workshop on the Algorithmic Foundations of Robotics (WAFR)*.

Kim N, Petingi L and Schlick T (2013) Network theory tools for RNA modeling. *WSEAS Transactions on Mathematics* 9(12): 941.

Konolige K, Grisetti G, Kummerle R, Burgard W, Limketkai B and Vincent R (2010) Efficient sparse pose adjustment for 2D mapping. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 22–29.

Krause A and Golovin D (2012) Submodular function maximization. *Tractability: Practical Approaches to Hard Problems* 3: 19.

Kretzschmar H and Stachniss C (2012) Information-theoretic compression of pose graphs for laser-based SLAM. *The International Journal of Robotics Research* 31: 1219–1230.

Kümmerle R, Grisetti G, Strasdat H, Konolige K and Burgard W (2011) g2o: A general framework for graph optimization. In: *2011 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 3607–3613.

Leskovec J, Krause A, Guestrin C, Faloutsos C, VanBriesen J and Glance N (2007) Cost-effective outbreak detection in networks. In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, pp. 420–429.

Li H, Patterson S, Yi Y and Zhang Z (2018) Maximizing the number of spanning trees in a connected graph. Preprint arXiv:1804.02785.

Löfberg J (2004) Yalmip: A toolbox for modeling and optimization in MATLAB. In: *Proceedings of the CACSD Conference*. Taipei, Taiwan. http://users.isy.liu.se/johanl/yalmip .

Mazuran M, Burgard W and Tipaldi GD (2016) Nonlinear factor recovery for long-term SLAM. *The International Journal of Robotics Research* 35(1–3): 50–72.

Meyer CD (2000) *Matrix analysis and applied linear algebra*. Philadelphia, PA: SIAM.

Minoux M (1978) Accelerated greedy algorithms for maximizing submodular set functions. In: *Optimization Techniques*. New York: Springer, pp. 234–243.

Mur-Artal R, Montiel JMM and Tardos JD (2015) ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics* 31(5): 1147–1163.

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

Nemhauser GL, Wolsey LA and Fisher ML (1978) An analysis of approximations for maximizing submodular set functions - I. *Mathematical Programming* 14(1): 265–294.

Olson E (2008) *Robust and Efficient Robotic Mapping*. PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

Olson E and Kaess M (2009) Evaluating the performance of map optimization algorithms. In: *RSS Workshop on Good Experimental Methodology in Robotics*, p. 40.

Paull L, Huang G and Leonard JJ (2016) A unified resource-constrained framework for graph SLAM. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1346–1353.

Paull L, Huang G, Seto M and Leonard JJ (2015) Communication-constrained multi-AUV cooperative SLAM. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 509–516.

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

Pirani M and Sundaram S (2014) Spectral properties of the grounded Laplacian matrix with applications to consensus in the presence of stubborn agents. In: *American Control Conference (ACC), 2014*. IEEE, pp. 2160–2165.

Pukelsheim F (1993) *Optimal design of experiments*, volume 50. Philadelphia, PA: SIAM.

Shamaiah M, Banerjee S and Vikalo H (2010) Greedy sensor selection: Leveraging submodularity. In: *49th IEEE Conference on Decision and Control (CDC)*. IEEE, pp. 2572–2577.

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

Sorenson H (1980) Control and systems theory. In: *Parameter estimation: principles and problems*. New York: Dekker.

Thrun S, Liu Y, Koller D, Ng AY, Ghahramani Z and Durrant-Whyte H (2004) Simultaneous localization and mapping with sparse extended information filters. *The International Journal of Robotics Research* 23(7-8): 693.

Tian Y, Khosoussi K, Giamou M, How JP and Kelly J (2018) Near-optimal budgeted data exchange for distributed loop closure detection. In: *Proceedings of Robotics: Science and Systems*. Pittsburgh, PA, accepted.

Tütüncü RH, Toh KC and Todd MJ (2003) Solving semidefinite-quadratic-linear programs using sdpt3. *Mathematical programming* 95(2): 189–217.

Vallve J, Sola J and Andrade-Cetto J (2018) Graph SLAM sparsification with populated topologies using factor descent optimization. *IEEE Robotics and Automation Letters*, in press.

Vandenberghe L, Boyd S and Wu SP (1998) Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications* 19(2): 499–533.

Vial J, Durrant-Whyte H and Bailey T (2011) Conservative sparsification for efficient and consistent approximate estimation. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 886–893.

Wang G (1994) A proof of Boesch's conjecture. *Networks* 24(5): 277–284.

Wang Y, Xiong R, Li Q and Huang S (2013) Kullback–Leibler divergence based graph pruning in robotic feature mapping. In: *2013 European Conference on Mobile Robots (ECMR)*. IEEE, pp. 32–37.

Wolsey LA (1982) An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica* 2(4): 385–393.

Appendix A. Preliminaries

For the reader's convenience, here we briefly review a number of important results, terminologies, and preliminaries from several areas that are used in this work.

A.1. Graph theory

Consider a *simple* undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Here $\mathcal{V}$ is the set of vertices and $\mathcal{E} \subset \binom{\mathcal{V}}{2}$ is the edge set of $\mathcal{G}$. $\mathcal{V}(\mathcal{G})$ and $\mathcal{E}(\mathcal{G})$ refer to, respectively, the vertex set and edge set of graph $\mathcal{G}$. Moreover, $n(\mathcal{G}) \triangleq |\mathcal{V}(\mathcal{G})|$ and $m(\mathcal{G}) \triangleq |\mathcal{E}(\mathcal{G})|$. Vertex u is *adjacent* to vertex v ($u \sim v$) if and only if there is an edge connecting u to v. For any vertex $v \in \mathcal{V}$, $\mathcal{N}(v) \subset \mathcal{V}$ denotes the set of all vertices adjacent to v. We generally assume the vertices are labeled according to $\mathcal{V} = [n]$ for an integer n. Let $w : \mathcal{E} \to \mathbb{R}$ be a *weight function* that assigns a real weight to any edge of $\mathcal{G}$. Then $(\mathcal{V}, \mathcal{E}, w)$ is an *edge-weighted graph* (or, simply, a weighted graph). In *directed graphs*, each edge has an *orientation* and thus is represented by an ordered pair, i.e., $\vec{\mathcal{G}} = (\mathcal{V}, \vec{\mathcal{E}})$ where $\vec{\mathcal{E}} \subset \mathcal{V} \times \mathcal{V}$. An undirected graph is *connected* if and only if there is a path between any two vertices. A directed graph is *strongly connected* if and only if there is a *directed path* between any two vertices. Similarly, a directed graph is *weakly connected* if there is a path between any two vertices after ignoring the edge orientations. $\mathcal{H} = (\mathcal{S}, \mathcal{F})$ is a *subgraph* of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ if and only if $\mathcal{S} \subset \mathcal{V}$ and $\mathcal{F} \subset \mathcal{E}$. $\mathcal{H}$ is a *spanning subgraph* of $\mathcal{G}$ if and only if and $\mathcal{S} = \mathcal{V}$. Moreover, $\mathcal{H}$ is an *induced subgraph* of $\mathcal{G}$ if and only if

$$\mathcal{F} = \{\{u, v\} : u, v \in \mathcal{S}, \{u, v\} \in \mathcal{E}\} \tag{72}$$

A *connected component* of an undirected graph $\mathcal{G}$ is a connected induced subgraph of $\mathcal{G}$ whose set of vertices is not adjacent to the other vertices of $\mathcal{G}$. The number of connected components of $\mathcal{G}$ is shown by $n_c(\mathcal{G})$. The *degree* of vertex $v \in \mathcal{V}$, $\deg(v)$, is defined as the number of edges incident to v, i.e., $\deg(v) \triangleq |\mathcal{N}(v)|$. From the handshaking lemma we know that $\sum_{v \in \mathcal{V}} \deg(v) = 2|\mathcal{E}|$. In weighted graphs, the weighted degree of vertex $v \in \mathcal{V}$ is defined as $\deg_w(v) \triangleq \sum_{u \in \mathcal{N}(v)} w(u, v)$. An undirected *tree* is a graph that is (i) connected and (ii) does not have a cycle. Removing any edge from a tree results in a disconnected graph; thus, trees are *minimally connected* graphs. We generally use $\mathcal{T}$ for referring to trees. The complete graph $\mathcal{K}$ (or $\mathcal{K}_n$ when $|\mathcal{V}| = n$) is the graph whose any two vertices are connected with an edge, i.e., $\mathcal{E}(\mathcal{K}) = \binom{\mathcal{V}}{2}$. A *spanning tree* of $\mathcal{G}$ is a spanning subgraph of $\mathcal{G}$ that is a tree. To simplify our notation, let us assume $n(\mathcal{G}) = n$ and $m(\mathcal{G}) = m$. The degree matrix $\mathbf{D} \in \mathbb{R}^{n \times n}$ is defined as $\mathbf{D} \triangleq \operatorname{diag}(\deg(v_1), \ldots, \deg(v_n))$ where $\mathcal{V}(\mathcal{G}) = \{v_1, \ldots, v_n\}$. The adjacency matrix $\mathbf{N} \in \{0, 1\}^{n \times n}$ is defined as follows:

$$\mathbf{N}_{u,v} = \begin{cases} 1 & \text{if } u \sim v \\ 0 & \text{otherwise} \end{cases}$$

The incidence matrix $\mathbf{A}_\circ \in \{-1, 0, 1\}^{n \times m}$ is defined for directed graphs. Let $\mathcal{E} = \{e_1, \ldots, e_m\}$ where $e_i \triangleq (u_i, v_i)$. Then we have

$$\mathbf{A}_{\circ\,u,i} = \begin{cases} +1 & \text{if } \exists v \in \mathcal{V}{:}e_{\mathrm{i}} = (v, u) \in \mathcal{E} \\ -1 & \text{if } \exists v \in \mathcal{V}{:}e_{\mathrm{i}} = (u, v) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$

Sometimes we refer to the incidence matrix of *undirected* graphs. In such cases, we have implicitly assumed an arbitrary orientation is assigned to the edges of graph. The graph Laplacian (or Kirchhoff) matrix $\mathbf{L}_\circ \in \mathbb{R}^{n\times n}$ is defined as $\mathbf{L}_\circ \triangleq \mathbf{A}_\circ\mathbf{A}_\circ^\top = \mathbf{D} - \mathbf{N}$. Deleting a row/column from the incidence and Laplacian matrices results in the following matrices. The *reduced* incidence matrix $\mathbf{A}$ is the matrix obtained by deleting the row associated to a vertex from the incidence matrix. The vertex associated to the deleted row is usually referred to as the *anchored* or *grounded* vertex. In general, it is possible to have multiple anchors. The *reduced* Laplacian (also known as the Dirichlet or grounded Laplacian) matrix $\mathbf{L}$ is the matrix obtained by deleting the column and row associated to a vertex from the Laplacian matrix. Similar to what we saw earlier regarding the relation between $\mathbf{A}_\circ$ and $\mathbf{L}_\circ$, $\mathbf{L} = \mathbf{A}\mathbf{A}^\top$ for the same anchor(s).

Theorem 8. *Let $\lambda_1(\mathbf{L}_\circ) \leqslant \lambda_2(\mathbf{L}_\circ) \leqslant \cdots \leqslant \lambda_n(\mathbf{L}_\circ)$ be the spectrum of the Laplacian matrix $\mathbf{L}_\circ$ of an arbitrary graph. The following statements hold.*

1. *The Laplacian matrix is positive semidefinite, i.e., $\lambda_i(\mathbf{L}_\circ) \geqslant 0$ for all $i \in [n]$.*
2. *The Laplacian matrix has at least one zero eigenvalue associated to the $\mathbf{1}_n$ eigenvector, i.e., $\mathbf{L}_\circ\mathbf{1}_n = \mathbf{0}$.*
3. *The multiplicity of the zero eigenvalue is equal to the number of connected components of graph $n_c(\mathcal{G})$, i.e., $\lambda_i(\mathbf{L}_\circ) = 0$ for all $i \in [n_c(\mathcal{G})]$ and $\lambda_{n_{c(\mathcal{G})}+1}(\mathbf{L}_\circ) > 0$.*

Theorem 9. *The reduced incidence matrix $\mathbf{A}$ is full row rank if and only if the corresponding graph is weakly connected.*

The following corollary directly follows Theorem 9 and the fact that $\mathbf{L} = \mathbf{A}\mathbf{A}^\top$.

Corollary 13. *The reduced Laplacian matrix $\mathbf{L}$ is positive definite if and only if the corresponding graph is connected.*

Theorem 10 (Kirchhoff's matrix-tree theorem). *Let $\mathbf{L}$ and $\mathbf{L}_\circ$ be, respectively, the reduced Laplacian and the Laplacian matrix of any simple undirected graph $\mathcal{G}$ after anchoring an arbitrary vertex out of its n vertices. The following statements are true:*

1. $t(\mathcal{G}) = \det(\mathbf{L})$;
2. $t(\mathcal{G}) = \frac{1}{n}\prod_{i=2}^{n}\lambda_i(\mathbf{L}_\circ)$.

Theorem 11. (Cayley's formula) *We have*

$$t(\mathcal{K}_n) = n^{n-2} \tag{73}$$

Definition 4. (Tree value function). Suppose $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ is a weighted graph with a non-negative weight function. The value of each spanning tree of $\mathcal{G}$ is measured by the following function,

$$\mathbb{V}_w : T_{\mathcal{G}} \to \mathbb{R}_{\geqslant 0} \tag{74}$$

$$\mathcal{T} \mapsto \prod_{e\in\mathcal{E}(\mathcal{T})} w(e) \tag{75}$$

Furthermore, we define the weighted number of trees as $t_w(\mathcal{G}) \triangleq \sum_{\mathcal{T}\in T_{\mathcal{G}}} \mathbb{V}_w(\mathcal{T})$.

Theorem 12 (Weighted matrix-tree theorem). *For $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ where $w : \mathcal{E} \to \mathbb{R}_{>0}$ we have $t_w(\mathcal{G}) = \det \mathbf{A}\mathbf{W}\mathbf{A}^\top$ in which $\mathbf{W} \triangleq \operatorname{diag}(w(e_1), \ldots, w(e_m))$.*

For edge-weighted graphs, $\mathbf{L} = \mathbf{A}\mathbf{W}\mathbf{A}^{\mathsf{T}}$ is the reduced Laplacian matrix. Thus, unweighted graphs can be thought of the case where $w(e) = 1$ for all $e \in \mathcal{E}$.

Lemma 5. *Let $\mathcal{G}_{uv}$ be the graph obtained by adding $\{u, v\} \notin \mathcal{E}$ with weight w_{uv} to $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$. Let $\mathbf{L}$ be the reduced Laplacian matrix and $\mathbf{a}_{uv}$ be the corresponding column of the reduced incidence matrix of $\mathcal{G}$ after anchoring an arbitrary vertex. If $\mathcal{G}$ is connected,*

$$t_w(\mathcal{G}_{uv}) = t_w(\mathcal{G}) \cdot (1 + w_{uv}\Delta_{uv}^{\mathcal{G}}) \tag{76}$$

where $\Delta_{uv}^{\mathcal{G}} \triangleq \mathbf{a}_{uv}^{\mathsf{T}}\mathbf{L}^{-1}\mathbf{a}_{uv}$ is the effective resistance between u and v (see, e.g., Ghosh et al., 2008).

Proof of Lemma 5. The reduced Laplacian matrix of $\mathcal{G}_{uv}$ can be written as $\mathbf{L}_{uv} = \mathbf{L} + w_{uv}\mathbf{a}_{uv}\mathbf{a}_{uv}{}^{\mathsf{T}}$. Taking the determinant and applying the matrix determinant lemma concludes the proof. □

Lemma 6. *Let $\mathcal{G}_1$ be a spanning subgraph of $\mathcal{G}_2$. For any $w : \mathcal{E}(K) \to \mathbb{R}_{\geqslant 0}$, $\mathbf{L}_{\mathcal{G}_2}^{w} \succeq \mathbf{L}_{\mathcal{G}_1}^{w}$ in which $\mathbf{L}_{\mathcal{G}}^{w}$ is the reduced weighted Laplacian matrix of $\mathcal{G}$ when its edges are weighted by w.*

Proof of Lemma 6. From the definition of the reduced weighted Laplacian matrix we have

$$\mathbf{L}_{\mathcal{G}_2}^{w} - \mathbf{L}_{\mathcal{G}_1}^{w} = \sum_{\{u,v\}\in\mathcal{E}(\mathcal{G}_2)\setminus\mathcal{E}(\mathcal{G}_1)} w_{uv}\mathbf{a}_{uv}\mathbf{a}_{uv}^\top \succeq \mathbf{0} \tag{77}$$

□

A.2. Estimation theory

An estimator $\hat{\mathbf{x}}$ of parameter $\mathbf{x}$ is called an *unbiased* estimator iff $\mathbb{E}[\hat{\mathbf{x}}] = \mathbf{x}$.

Theorem 13. (CRLB) *Under some regularity conditions (Sorenson, 1980), the covariance matrix of any unbiased estimator of $\mathbf{x}$, such as $\hat{\mathbf{x}}$, satisfies $\mathbb{C}\mathrm{ov}[\hat{\mathbf{x}}] \succeq \mathbb{I}^{-1}(\mathbf{x})$, where $\mathbb{I}(\mathbf{x})$ is the FIM,*

$$\mathbb{I}(\mathbf{x}) \triangleq \mathbb{E}_{\mathbf{z}}\left[\frac{\partial}{\partial\mathbf{x}}\log p(\mathbf{z};\mathbf{x})\frac{\partial^{\mathsf{T}}}{\partial\mathbf{x}}\log p(\mathbf{z};\mathbf{x})\right] \tag{78}$$

Here the expectation is over $\mathbf{z}$ and with respect to $p(\mathbf{z};\mathbf{x})$. Note that the FIM depends only on the true value of $\mathbf{x}$ and $p(\mathbf{z};\mathbf{x})$, and does not depend on any particular realization of $\mathbf{z}$.

An unbiased estimator that achieves CRLB is called an efficient estimator.

Corollary 14. *The following statements are true.*

1. *The diagonal elements of CRLB are lower bounds for the variance of any unbiased estimator for each parameter.*
2. *The determinant of CRLB is a lower bound for the determinant of the covariance matrix of any unbiased estimator.*

A.3. Submodular maximization

For a comprehensive survey of key results in submodular set function optimization see Krause and Golovin (2012). Suppose $\mathcal{W}$ is a finite *ground set*. Let $\xi\colon 2^{\mathcal{W}} \to \mathbb{R}$ be a real set function defined over the power set of $\mathcal{W}$. Finally, let $\mathcal{F} \subset 2^{\mathcal{W}}$ be the set of feasible subsets of $\mathcal{W}$. Then our goal is to solve the following optimization problem:

$$\begin{array}{ll} \underset{\mathcal{A}\subset\mathcal{W}}{\text{maximize}} & \xi(\mathcal{A}) \\ \text{subjectto} & \mathcal{A} \in \mathcal{F} \end{array} \tag{79}$$

Definition 5. Suppose $\mathcal{W}$ is a finite ground set. For any set function $\xi\colon 2^{\mathcal{W}} \to \mathbb{R}$:

1. ξ is called *normalized* if and only if $\xi(\varnothing) = 0$;
2. ξ is called *monotone* if $\xi(\mathcal{B}) \geqslant \xi(\mathcal{A})$ for every $\mathcal{A}$ and $\mathcal{B}$ s.t. $\mathcal{A} \subset \mathcal{B} \subset \mathcal{W}$;
3. ξ is called *submodular* if and only if for every $\mathcal{A}$ and $\mathcal{B}$ s.t. $\mathcal{A} \subset \mathcal{B} \subset \mathcal{W}$ and $\forall s \in \mathcal{W}\backslash\mathcal{B}$ we have,

$$\xi(\mathcal{A} \cup \{s\}) - \xi(\mathcal{A}) \geqslant \xi(\mathcal{B} \cup \{s\}) - \xi(\mathcal{B}) \tag{80}$$

4. ξ is called log-*submodular* if and only if ξ is positive and $\log \xi$ is submodular.

Theorem 14 (Nemhauser et al., 1978). *Let* $f\colon 2^{\mathcal{W}} \to \mathbb{R}$ *be a normalized monotone submodular. Let* OPT *be the optimal value of the following problem,*

$$\begin{array}{ll} \underset{\mathcal{A}\subset\mathcal{W}}{\text{maximize}} & f(\mathcal{A}) \\ \text{subject to} & |\mathcal{A}| \leqslant k \in \mathbb{N} \end{array} \tag{81}$$

Then, $f_{\text{greedy}} \geqslant (1 - 1/e)\text{OPT}$ *where* f_{greedy} *is the value achieved by the natural greedy algorithm.*

Theorem 15 introduces several operations under which both monotonicity and submodularity are preserved.

Theorem 15. *Both monotonicity and submodularity are preserved under the following operations.*

- *For any constant* $c \in \mathbb{R}$, $g\colon 2^{\mathcal{W}} \to \mathbb{R} : \mathcal{A} \mapsto f(\mathcal{A}) + c$ *is monotone submodular if and only if* $f\colon 2^{\mathcal{W}} \to \mathbb{R}$ *is monotone submodular.*
- *For any constant* $c \in \mathbb{R}$, $g\colon 2^{\mathcal{W}} \to \mathbb{R} : \mathcal{A} \mapsto \min\{f(\mathcal{A}), c\}$ *is monotone submodular if and only if* $f\colon 2^{\mathcal{W}} \to \mathbb{R}$ *is monotone submodular.*
- *If* $f_i\colon 2^{\mathcal{W}} \to \mathbb{R}$ *for all* $i \in [n]$ *are monotone submodular and* w_i *for all* $i \in [n]$ *are non-negative, then* $g : \mathcal{A} \mapsto \sum_{i=1}^{n} w_i f_i(\mathcal{A})$ *is monotone submodular.*
- *If* $f\colon 2^{\mathcal{W}} \to \mathbb{R}$ *is monotone submodular, then for any* $\mathcal{M} \subset \mathcal{W}$, $g : \mathcal{A} \mapsto f(\mathcal{A} \cup \mathcal{M})$ *is monotone submodular.*

A.4. Linear algebra

Lemma 7. (Schur's determinant formula (Meyer, 2000)). *If* $\mathbf{A}^{-1}$ *exists,*

$$\det\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \det \mathbf{A} \cdot \det(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}) \tag{82}$$

Lemma 8. *For any two* $\mathbf{N}, \mathbf{M} \in \mathbb{S}^n_{\geqslant 0}$ *we have*

$$\det(\mathbf{M} + \mathbf{N}) \geqslant \det(\mathbf{M}) \tag{83}$$

Proof. It is trivial to verify the lemma when $\mathbf{M}$ is singular. For $\mathbf{M} \succ \mathbf{0}$, we can decompose $\mathbf{M}$ as $\mathbf{M} = \mathbf{M}^{\frac{1}{2}}\mathbf{M}^{\frac{1}{2}}$ in which $\mathbf{M}^{\frac{1}{2}} \in \mathbb{S}^n_{>0}$. Then we have

$$\det(\mathbf{M} + \mathbf{N}) = \det(\mathbf{M}) \det(\mathbf{I} + \mathbf{M}^{-\frac{1}{2}}\mathbf{N}\mathbf{M}^{-\frac{1}{2}})) \tag{84}$$

$$= \det(\mathbf{M}) \prod_{i=1}^{n} (1 + \underbrace{\lambda_i(\mathbf{M}^{-\frac{1}{2}}\mathbf{N}\mathbf{M}^{-\frac{1}{2}})}_{\geqslant 0}) \tag{85}$$

$$\geqslant \det(\mathbf{M}) \tag{86}$$

Lemma 9. *For any* $\mathbf{M} \in \mathbb{S}^n_{>0}$ *and* $\mathbf{N} \in \mathbb{S}^n_{>0}$, $\mathbf{M} \succeq \mathbf{N}$ *iff* $\mathbf{N}^{-1} \succeq \mathbf{M}^{-1}$.

Proof. Owing to symmetry it suffices to prove that $\mathbf{M} \succeq \mathbf{N} \Rightarrow \mathbf{N}^{-1} \succeq \mathbf{M}^{-1}$. Multiplying both sides of $\mathbf{M} \succeq \mathbf{N}$ by $\mathbf{N}^{-\frac{1}{2}}$ from left and right results in $\mathbf{N}^{-\frac{1}{2}}\mathbf{M}\mathbf{N}^{-\frac{1}{2}} - \mathbf{I} \succeq \mathbf{0}$. Therefore, the eigenvalues of $\mathbf{N}^{-\frac{1}{2}}\mathbf{M}\mathbf{N}^{-\frac{1}{2}}$, which are the same as the eigenvalues of $\mathbf{M}^{\frac{1}{2}}\mathbf{N}^{-1}\mathbf{M}^{\frac{1}{2}}$, are at least 1. Therefore, $\mathbf{M}^{\frac{1}{2}}\mathbf{N}^{-1}\mathbf{M}^{\frac{1}{2}} - \mathbf{I} \succeq \mathbf{0}$. Multiplying both sides by $\mathbf{M}^{-\frac{1}{2}}$ from left and right proves the lemma.

Appendix B. Proofs

Proof of Proposition 1. Let $\mathbf{H} \triangleq (\mathbf{A} \otimes \mathbf{I}_d)^{\top}$ denote the measurement function in $\mathbb{R}^d$-Sync. Plugging $p(\mathbf{z}; \mathbf{x}) = \mathcal{N}(\mathbf{z}; \mathbf{H}\mathbf{x}, \mathbf{\Sigma})$ into the definition of the FIM (78) results in

$$\mathbb{I} = \mathbf{H}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\mathbf{H} \tag{87}$$

$$= (\mathbf{A} \otimes \mathbf{I}_d)(\mathbf{W} \otimes \mathbf{I}_d)(\mathbf{A} \otimes \mathbf{I}_d)^{\top} \tag{88}$$

$$= (\mathbf{A}\mathbf{W}\mathbf{A}^{\top}) \otimes \mathbf{I}_d \tag{89}$$

$$= \mathbf{L}_w \otimes \mathbf{I}_d \tag{90}$$ □

Proof of Proposition 2. The proof is similar to that of Proposition 1. Define $\mathbf{H} \triangleq \mathbf{R}^\top (\mathbf{A} \otimes \mathbf{I}_d)^\top$. Plugging $p(\mathbf{z}; \mathbf{x}) = \mathcal{N}(\mathbf{z}; \mathbf{H}\mathbf{x}, \boldsymbol{\Sigma})$ into the definition of the FIM (78) results in

$$\mathbb{I} = \mathbf{H}^\mathrm{T} \mathbb{S}^{-1} \mathbf{H} \tag{91}$$

$$= (\mathbf{A} \otimes \mathbf{I}_d) \mathbf{R} \mathbb{S}^{-1} \mathbf{R}^\mathrm{T} (\mathbf{A} \otimes \mathbf{I}_d)^\mathrm{T} \tag{92}$$

$$= (\mathbf{A} \otimes \mathbf{I}_d)(\mathbf{W} \otimes \mathbf{I}_d)(\mathbf{A} \otimes \mathbf{I}_d)^\mathrm{T} \tag{93}$$

$$= (\mathbf{A}\mathbf{W}\mathbf{A}^\mathrm{T}) \otimes \mathbf{I}_d \tag{94}$$

$$= \mathbf{L}_w \otimes \mathbf{I}_d \tag{95}$$

Note that the ith $d \times d$ diagonal block of $\mathbf{R}$ (i.e., $\mathbf{R}_i$ for which we have $\mathbf{R}_i^\mathrm{T} \mathbf{R}_i = \mathbf{I}_d$) commutes with that of $\mathbb{S}^{-1}$, i.e., $\sigma_i^{-2} \mathbf{I}_d$. □

Proof of Proposition 3. Note that $f^\circ \triangleq f(\mathbf{x}^\circ) = \| \boldsymbol{\epsilon} \|^2_{\mathbb{S}^{-1}} = \| \bar{\boldsymbol{\epsilon}} \|^2$ in which $\bar{\boldsymbol{\epsilon}} \triangleq \mathbb{S}^{-1/2} \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. This concludes the proof as $\| \bar{\boldsymbol{\epsilon}} \|^2$, by definition, is distributed according to $\chi^2_{\dim(\mathbf{z})}$. □

Proof of Proposition 4. Define $\tilde{\mathbf{H}} \triangleq \mathbb{S}^{-\frac{1}{2}} \mathbf{H}$, $\tilde{\mathbf{z}} \triangleq \mathbb{S}^{-\frac{1}{2}} \mathbf{z}$, $\tilde{\mathbf{c}} \triangleq \mathbb{S}^{-\frac{1}{2}} \mathbf{c}$ and $\tilde{\boldsymbol{\epsilon}} \triangleq \mathbb{S}^{-\frac{1}{2}} \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The MLE is given by

$$\mathbf{x}^\star = \underbrace{(\tilde{\mathbf{H}}^\top \tilde{\mathbf{H}})^{-1} \tilde{\mathbf{H}}^\top}_{\tilde{\mathbf{H}}^\dagger} (\tilde{\mathbf{z}} - \tilde{\mathbf{c}}) \tag{96}$$

in which $\tilde{\mathbf{H}}^\dagger \triangleq (\tilde{\mathbf{H}}^\top \tilde{\mathbf{H}})^{-1} \tilde{\mathbf{H}}^\top$ is the Moore–Penrose pseudoinverse of $\tilde{\mathbf{H}}$. Now we evaluate $f(\mathbf{x})$ at $\mathbf{x} = \mathbf{x}^\star$,

$$f(\mathbf{x}^\star) = \| \mathbf{z} - \mathbf{H}\mathbf{x}^\star - \mathbf{c} \|^2_{\mathbb{S}^{-1}} \tag{97}$$

$$= \| \tilde{\mathbf{z}} - \tilde{\mathbf{H}}\mathbf{x}^\star - \tilde{\mathbf{c}} \|^2 \tag{98}$$

$$= \| \tilde{\mathbf{z}} - \tilde{\mathbf{c}} - \tilde{\mathbf{H}}\tilde{\mathbf{H}}^\dagger (\tilde{\mathbf{z}} - \tilde{\mathbf{c}}) \|^2 \tag{99}$$

$$= \| (\mathbf{I} - \tilde{\mathbf{H}}\tilde{\mathbf{H}}^\dagger)(\tilde{\mathbf{z}} - \tilde{\mathbf{c}}) \|^2 \tag{100}$$

$$= \| (\mathbf{I} - \tilde{\mathbf{H}}\tilde{\mathbf{H}}^\dagger)(\tilde{\mathbf{H}}\mathbf{x}^\circ + \tilde{\mathbf{c}} + \tilde{\boldsymbol{\epsilon}} - \tilde{\mathbf{c}}) \|^2 \tag{101}$$

$$= \| (\mathbf{I} - \tilde{\mathbf{H}}\tilde{\mathbf{H}}^\dagger)(\tilde{\mathbf{H}}\mathbf{x}^\circ + \tilde{\boldsymbol{\epsilon}}) \|^2 \tag{102}$$

$$= \| (\mathbf{I} - \tilde{\mathbf{H}}\tilde{\mathbf{H}}^\dagger) \tilde{\boldsymbol{\epsilon}} \|^2 \tag{103}$$

Now note that $\mathbf{I} - \tilde{\mathbf{H}}\tilde{\mathbf{H}}^\dagger$ is the orthogonal projection onto the nullspace of $\tilde{\mathbf{H}}^\mathrm{T}$ with dimension $r \triangleq \dim(\mathbf{z}) - \dim(\mathbf{x})$. Let $\{\mathbf{u}_i\}_{i=1}^r$ be an orthonormal basis for the nullspace of $\tilde{\mathbf{H}}^\mathrm{T}$. Therefore, $(\mathbf{I} - \tilde{\mathbf{H}}\tilde{\mathbf{H}}^\dagger)\tilde{\boldsymbol{\epsilon}} = \sum_{i=1}^r (\mathbf{u}_i^\mathrm{T} \tilde{\boldsymbol{\epsilon}}) \mathbf{u}_i$. Now note that

$$\| (\mathbf{I} - \tilde{\mathbf{H}}\tilde{\mathbf{H}}^\dagger) \tilde{\boldsymbol{\epsilon}} \|^2 = \tilde{\boldsymbol{\epsilon}}^\mathrm{T} (\mathbf{I} - \tilde{\mathbf{H}}\tilde{\mathbf{H}}^\dagger) \tilde{\boldsymbol{\epsilon}} \tag{104}$$

$$= \tilde{\boldsymbol{\epsilon}}^\mathrm{T} \sum_{i=1}^r (\mathbf{u}_i^\mathrm{T} \tilde{\boldsymbol{\epsilon}}) \mathbf{u}_i \tag{105}$$

$$= \sum_{i=1}^r (\mathbf{u}_i^\mathrm{T} \tilde{\boldsymbol{\epsilon}})^2 \tag{106}$$

This concludes the proof since $(\mathbf{u}_i^\mathrm{T} \tilde{\boldsymbol{\epsilon}}) \sim \mathcal{N}(0, 1)$ and, moreover, $(\mathbf{u}_i^\mathrm{T} \tilde{\boldsymbol{\epsilon}})$ and $(\mathbf{u}_j^\mathrm{T} \tilde{\boldsymbol{\epsilon}})$ are independent for $i \neq j$. □

Proof of Proposition 5. Recall that $\mathbf{L} = \mathbf{A}\mathbf{A}^\top$ where $\mathbf{A}$ is the reduced incidence matrix of $\mathcal{G}$. It is clear that $\lambda_{\min}(\mathbf{L}) \geqslant 0$. $\mathbf{L}$ is singular iff the graph is disconnected. Let $\mathbf{a}$ be the reduced incidence vector of a new edge. The reduced Laplacian of graph after adding this new edge is $\mathbf{L} + \mathbf{a}\mathbf{a}^\top$. Thus, we need to show that $\lambda_{\min}(\mathbf{L} + \mathbf{a}\mathbf{a}^\top) \geqslant \lambda_{\min}(\mathbf{L})$, which follows from Weyl's inequality (Horn and Johnson, 1990). Now the fact that $\lambda_{\min}(\mathbf{L})$ is maximized when $\mathcal{G}$ is complete follows from the monotonicity of $\lambda_{\min}(\mathbf{L})$ in the edge set. The spectrum of the Laplacian matrix of $\mathcal{K}_n$ consists of a zero eigenvalue, and $\lambda = n$ with multiplicity $n - 1$ (Godsil and Royle, 2001). From Cauchy's interlace theorem (Godsil and Royle, 2001: Theorem 9.1.1) it follows that the spectrum of the *reduced* Laplacian of $\mathcal{K}_n$ consists of $\lambda_{\min}(\mathbf{L}) = 1$, and $\lambda = n$ with multiplicity $n - 1$. Finally, the last statement follows directly from Pirani and Sundaram (2014: Theorem 1). □

Proof of Theorem 1. Recall that the FIM in $\mathbb{R}^d$-Sync and d-dimensional Compass-SLAM is given by $\mathbb{I} = \mathbf{L}_w \otimes \mathbf{I}_d$ in which $\mathbf{L}_w$ is the reduced weighted Laplacian matrix of the graph. According to Cauchy's interlace theorem (Godsil and Royle, 2001: Theorem 9.1.1), $0 < \lambda_1(\mathbf{L}) \leqslant \lambda_2(\mathbf{L}_{\circ_w})$. We have

$$\lambda_{\max}(\mathbb{C}\mathrm{ov}[\mathbf{x}^\star]) = \lambda_1(\mathbb{I})^{-1} \tag{107}$$

$$= \lambda_1(\mathbf{L}_w \otimes \mathbf{I}_d)^{-1} \tag{108}$$

$$= \lambda_1(\mathbf{L}_w)^{-1} \tag{109}$$

$$\geqslant \lambda_2(\mathbf{L}_{\circ_w})^{-1} \tag{110}$$

The other statement follows from the above identity and the fact that $\lambda_1(\mathbf{L}_w) \leqslant w^{\ddagger}_{\max}$ (Proposition 5). □

Proof of Theorem 2. Based on Propositions 1 and 2 and the weighted matrix-tree theorem (Theorem 12), we have

$$\log\det(\mathbb{C}\mathrm{ov}[\mathbf{x}^\star]) = -\log\det(\mathbb{I}) \tag{111}$$

$$= -\log\det(\mathbf{L}_w \otimes \mathbf{I}_d) \tag{112}$$

$$= -d\tau_w(\mathcal{G}) \tag{113}$$

□

Proof of Theorem 3. Let us first define

$$\mathbf{P}^\perp_{w_p} \triangleq \mathbf{I} - \boldsymbol{\Gamma}^\top (\mathbf{A}^\top_{w_p} \mathbf{L}^{-1}_{w_p} \mathbf{A}_{w_p} \otimes \mathbf{I}_2) \Gamma \tag{114}$$

Applying Schur's determinant formula (Lemma 7) on the top-left block of (18) and using

$$\log\det(\mathbf{L}_{w_p} \otimes \mathbf{I}_2) = 2 \cdot \log\det \mathbf{L}_{w_p} \tag{115}$$

$$= 2 \cdot \tau_{w_p}(\mathcal{G}) \tag{116}$$

results in $\log\det \mathbb{I}(\mathbf{x}) = 2 \cdot \tau_{w_p}(\mathcal{G}) + \log\det(\mathbf{L}_{w_\theta} + \mathbf{E})$ where $\mathbf{E} \triangleq \boldsymbol{\Delta}_{w_p}^\top \mathbf{P}_{w_p}^\perp \boldsymbol{\Delta}_{w_p}$. Note that $\mathbf{P}_{w_p} \triangleq \mathbf{I} - \mathbf{P}_{w_p}^\perp \succeq \mathbf{0}$ and $\mathbf{P}_{w_p}^\perp \succeq \mathbf{0}$ since they are orthogonal projections. Consequently $\mathbf{E} \succeq \mathbf{0}$. To show that $\epsilon(\mathbf{x}) \geqslant 0$ we just need to apply Lemma 8 on the above identity:

$$\log\det \mathbb{I}(\mathbf{x}) = 2 \cdot \tau_{w_p}(\mathcal{G}) + \log\det(\mathbf{L}_{w_\theta} + \mathbf{E}) \tag{117}$$

$$\geqslant 2 \cdot \tau_{w_p}(\mathcal{G}) + \log\det \mathbf{L}_{w_\theta} \tag{118}$$

$$= 2 \cdot \tau_{w_p}(\mathcal{G}) + \tau_{w_\theta}(\mathcal{G}) \tag{119}$$

$$= \ell_\tau(\mathcal{G}) \tag{120}$$

Now we have

$$\epsilon(\mathbf{x}) \triangleq \log\det \mathbb{I}(\mathbf{x}) - \ell_\tau(\mathcal{G}) \tag{121}$$

$$\leqslant \log\det(\mathbf{L}_{w_\theta} + \boldsymbol{\Delta}_{w_p}^\top \boldsymbol{\Delta}_{w_p}) - \log\det \mathbf{L}_{w_\theta} \tag{122}$$

$$\leqslant \log\det(\mathbf{L}_{w_\theta} + \delta \mathbf{I}) - \log\det \mathbf{L}_{w_\theta} \tag{123}$$

$$= \log \prod_{i=1}^{n} \frac{\lambda_i(\mathbf{L}_{w_\theta}) + \delta}{\lambda_i(\mathbf{L}_{w_\theta})} \tag{124}$$

$$= \log \prod_{i=1}^{n} (1 + \delta/\lambda_i(\mathbf{L}_{w_\theta})) \tag{125}$$

$$\leqslant \log(1 + \delta/\lambda_1(\mathbf{L}_{w_\theta}))^n \tag{126}$$

$$= n \log(1 + \delta/\lambda_1(\mathbf{L}_{w_\theta})) \tag{127}$$

Note that (122) follows from applying Fischer's inequality (Horn and Johnson, 1990) on the FIM (18). Finally, (123) follows from Lemma 8 applied to

$$\log\det(\underbrace{\mathbf{L}_{w_\theta} + \boldsymbol{\Delta}_{w_p}^\top \boldsymbol{\Delta}_{w_p}}_{\succ \mathbf{0}} + \underbrace{\delta \mathbf{I} - \boldsymbol{\Delta}_{w_p}^\top \boldsymbol{\Delta}_{w_p}}_{\succeq \mathbf{0}}) \tag{128}$$

respectively; see (19).

Proof of Lemma 1. Let $w_\alpha : e \mapsto \alpha w(e)$ be the scaled weight function. It is easy to show that $t_{w_\alpha}(\mathcal{G}) = \alpha^{n-1} t_w(\mathcal{G})$. Therefore, $t_w(\mathcal{G}) \geqslant t_w(\mathcal{H}) \Leftrightarrow t_{w_\alpha}(\mathcal{G}) \geqslant t_{w_\alpha}(\mathcal{H})$ for any $\mathcal{G}$ and $\mathcal{H}$ with n vertices. □

Proof of Theorem 4.

1. Normalized: By definition $\Phi_w(\varnothing) = \tau_{n,w}(\mathcal{E}_{\text{init}}) - \tau_{n,w}(\mathcal{E}_{\text{init}}) = 0$.
2. Monotone: We need to show that $\Phi_w(\mathcal{E} \cup \{e\}) \geqslant \Phi_w(\mathcal{E})$. This is equivalent to showing that,

$$\tau_{n,w}(\mathcal{E}_{\text{init}} \cup \mathcal{E} \cup \{e\}) \geqslant \tau_{n,w}(\mathcal{E}_{\text{init}} \cup \mathcal{E}) \tag{129}$$

Now note that $(\mathcal{V}, \mathcal{E}_{\text{init}} \cup \mathcal{E})$ is connected since $(\mathcal{V}, \mathcal{E}_{\text{init}})$ was assumed to be connected. Therefore, we can apply Lemma 5 on the left-hand side of (129); i.e.,

$$\tau_{n,w}(\mathcal{E}_{\text{init}} \cup \mathcal{E} \cup \{e\}) = \tau_{n,w}(\mathcal{E}_{\text{init}} \cup \mathcal{E}) + \log(1 + w_e \Delta_e) \tag{130}$$

Thus we need to show that $\log(1 + w_e \Delta_e)$ is non-negative. Since $(\mathcal{V}, \mathcal{E}_{\text{init}})$ is connected, $\mathbf{L}$ is positive definite. Consequently $w_e \Delta_e = w_e \mathbf{a}_e^\top \mathbf{L}^{-1} \mathbf{a}_e > 0$ and, hence, $\log(1 + w_e \Delta_e) > 0$.

3. Submodular: Φ_w is submodular iff for all $\mathcal{E}_1 \subset \mathcal{E}_2 \subset \mathcal{E}(\mathcal{K}_n)$ and all $e \in \mathcal{E}(\mathcal{K}_n) \backslash \mathcal{E}_2$,

$$\Phi_w(\mathcal{E}_1 \cup \{e\}) - \Phi_w(\mathcal{E}_1) \geqslant \Phi_w(\mathcal{E}_2 \cup \{e\}) - \Phi_w(\mathcal{E}_2) \tag{131}$$

After canceling out $\tau_{n,w}(\mathcal{E}_{\text{init}})$ we need to show that

$$\begin{aligned} &\tau_{n,w}(\mathcal{E}_1 \cup \mathcal{E}_{\text{init}} \cup \{e\}) - \tau_{n,w}(\mathcal{E}_1 \cup \mathcal{E}_{\text{init}}) \\ &\geqslant \tau_{n,w}(\mathcal{E}_2 \cup \mathcal{E}_{\text{init}} \cup \{e\}) - \tau_{n,w}(\mathcal{E}_2 \cup \mathcal{E}_{\text{init}}) \end{aligned} \tag{132}$$

If $e \in \mathcal{E}_{\text{init}}$, both sides of (132) become zero. Hence, we can safely assume that $e \notin \mathcal{E}_{\text{init}}$. For convenience, let us define $\mathcal{E}_1^* \triangleq \mathcal{E}_i \cup \mathcal{E}_{\text{init}}$ for i = 1,2. Therefore, (132) can be rewritten as

$$\tau_{n,w}(\mathcal{E}_1^* \cup \{e\}) - \tau_{n,w}(\mathcal{E}_1^*) \geqslant \tau_{n,w}(\mathcal{E}_2^* \cup \{e\}) - \tau_{n,w}(\mathcal{E}_2^*) \tag{133}$$

Recall that we assumed that $(\mathcal{V}, \mathcal{E}_{\text{init}})$ is connected. Thus, $(\mathcal{V}, \mathcal{E}_i^*)$ is connected for $i = 1, 2$, and we can apply Lemma 5 on both sides of (133). After doing so we have to show that

$$\log(1 + w_e \Delta_e^{\mathcal{G}_1}) \geqslant \log(1 + w_e \Delta_e^{\mathcal{G}_2}) \tag{134}$$

where $\mathcal{G}_i \triangleq (\mathcal{V}, \mathcal{E}_i \cup \mathcal{E}_{\text{init}}, w)$ for $i = 1, 2$. It is easy to see that (134) holds iff $\Delta_e^{\mathcal{G}_1} \geqslant \Delta_e^{\mathcal{G}_2}$. Now note that

$$\Delta_e^{\mathcal{G}_1} - \Delta_e^{\mathcal{G}_2} = \mathbf{a}_e^{\mathrm{T}} (\mathbf{L}_{\mathcal{G}_1}^{-1} - \mathbf{L}_{\mathcal{G}_2}^{-1}) \mathbf{a}_e \geqslant 0 \tag{135}$$

since $\mathbf{L}_{\mathcal{G}_2} \succeq \mathbf{L}_{\mathcal{G}_1}$ ($\mathcal{G}_1$ is a spanning subgraph of $\mathcal{G}_2$), and therefore according to Lemma 9, $\mathbf{L}_{\mathcal{G}_1}^{-1} \succeq \mathbf{L}_{\mathcal{G}_2}^{-1}$.

□

Proof of Lemma 2. This follows readily from Corollary 13. □

Proof of Lemma 3. This immediately follows from the fact that $\mathbf{p}^\star \in \{0,1\}^c$ is a feasible solution for k-ESP that is optimal for the relaxed problem. □

Proof of Theorem 5. The first statement holds since

$$\mathbb{E}[k^*] = \mathbb{E}\left[\sum_{i=1}^{c} \boldsymbol{\pi}_i\right] \tag{136}$$

$$= \sum_{i=1}^{c} \mathbb{E}[\boldsymbol{\pi}_i] \tag{137}$$

$$= \sum_{i=1}^{c} p_i \tag{138}$$

For the second statement, see the Proof of Theorem 7. This statement is a special case of Theorem 7 when $\{\mathbf{u}_i\}$ and $\{\mathbf{v}_i\}$ are the columns of the reduced (weighted) incidence matrix.

Proof of Theorem 6. Let $p_1^\star \leqslant p_2^\star \leqslant \cdots \leqslant p_c^\star$ be the sorted version of $\{\mathbf{p}_i^\star\}_{i=1}^c$. Now for all $i \in [c]$, define $q_i^\star \triangleq 1 - p_i^\star$. Hence, we have $q_1^\star \geqslant q_2^\star \geqslant \ldots \geqslant q_c^\star$. Thus maximizing (54) can be expressed as

$$\underset{\mathcal{A}\subset[c],\,|\mathcal{A}|=k}{\text{maximize}} \prod_{i\in\mathcal{A}} p_i^\star \prod_{j\in[c]\setminus\mathcal{A}} q_j^\star \tag{139}$$

It is easy to see that this expression is maximized by picking $\mathcal{A}^\star = \{c-i\}_{i=0}^{k-1}$, which corresponds to the set selected by the deterministic rounding procedure. □

Proof of Theorem 7. We begin by applying the Cauchy–Binet formula:

$$\mathbb{E}_\pi\left[\det\left(\sum_{i=1}^m \pi_i \mathbf{u}_i \mathbf{v}_i^\mathrm{T}\right)\right] \tag{140}$$

$$= \mathbb{E}_\pi\left[\sum_{\mathcal{Q}\in\binom{[m]}{n}} \det\left(\sum_{k\in\mathcal{Q}} \pi_k \mathbf{u}_k \mathbf{v}_k^\mathrm{T}\right)\right] \tag{141}$$

$$= \sum_{\mathcal{Q}\in\binom{[m]}{n}} \mathbb{E}_\pi\left[\det\left(\sum_{k\in\mathcal{Q}} \pi_k \mathbf{u}_k \mathbf{v}_k^\mathrm{T}\right)\right] \tag{142}$$

Since $|\mathcal{Q}| = n$ we have

$$\operatorname{rank}\left(\sum_{k\in\mathcal{Q}} \pi_k \mathbf{u}_k \mathbf{v}_k^\top\right) = \begin{cases} n & \text{iff } \pi_\mathrm{k} = 1 \text{ for all } k \in \mathcal{Q} \\ \gamma < n & \text{otherwise} \end{cases} \tag{143}$$

Hence, the determinant can be non-zero only when $\pi_k = 1$ for all $k \in \mathcal{Q}$. Therefore,

$$\det\left(\sum_{k\in\mathcal{Q}} \pi_k \mathbf{u}_k \mathbf{v}_k^\top\right) \tag{144}$$

$$= \begin{cases} \det\big(\sum_{k\in\mathcal{Q}} \mathbf{u}_k \mathbf{v}_k^\mathrm{T}\big) & \text{iff } \pi_k = 1 \text{ for all } k \in \mathcal{Q} \\ 0 & \text{otherwise} \end{cases} \tag{145}$$

However, from the independence assumption we know that,

$$\mathbb{P}\Big[\bigwedge_{k\in\mathcal{Q}} \pi_k = 1\Big] = \prod_{k\in\mathcal{Q}} p_k \tag{146}$$

Each individual expectation in (142) can be computed as follows:

$$\mathbb{E}_\pi\left[\det\left(\sum_{k\in\mathcal{Q}} \pi_k \mathbf{u}_k \mathbf{v}_k^\mathrm{T}\right)\right] = \det\left(\sum_{k\in\mathcal{Q}} \mathbf{u}_k \mathbf{v}_k^\mathrm{T}\right) \prod_{k\in\mathcal{Q}} p_k \tag{147}$$

$$= \det\left(\sum_{k\in\mathcal{Q}} p_k \mathbf{u}_k \mathbf{v}_k^\mathrm{T}\right) \tag{148}$$

Plugging (148) back into (142) yields

$$\mathbb{E}_\pi\left[\det\left(\sum_{i=1}^m \pi_i \mathbf{u}_i \mathbf{v}_i^\top\right)\right] = \sum_{\mathcal{Q}\in\binom{[m]}{n}} \det\left(\sum_{k\in\mathcal{Q}} p_k \mathbf{u}_k \mathbf{v}_k^\top\right) \tag{149}$$

Note that (149) is nothing but the Cauchy–Binet expansion of $\det\big(\sum_{i=1}^m p_i \mathbf{u}_i \mathbf{v}_i^\mathrm{T}\big)$. This concludes the proof. □

Proof of Lemma 4. See the proof of Lemma 3. □