UNIVERSITY OF TECHNOLOGY, SYDNEY

Exploiting the Intrinsic Structures of Simultaneous Localization and Mapping

by

Kasra Khosoussi

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

at the

Centre for Autonomous Systems
Faculty of Engineering and Information Technology
University of Technology Sydney

March 2017

Declaration

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature:

Date:

Abstract

Imagine a *robot* which is assigned a *complex task* that requires it to navigate in an *unknown environment*. This situation frequently arises in a wide spectrum of applications; e.g., surgical robots used for diagnosis and treatment operate inside the body, domestic robots have to operate in people's houses, and Mars rovers explore Mars. To navigate safely in such environments, the robot needs to constantly estimate its location and, simultaneously, create a consistent representation of the environment by, e.g., building a map. This problem is known as simultaneous localization and mapping (SLAM). Over the past three decades, SLAM has always been a central topic in mobile robotics. Tremendous progress has been made during these years in efficiently solving SLAM using a variety of sensors in large-scale environments. However, some of the intrinsic structures of SLAM are yet to be fully understood and utilized. This thesis is devoted to the study of some of these structures.

This thesis is comprised of two main parts:

In the first part, we examine the specific structure of the standard measurement models in pose-graph and feature-based SLAM. These standard models are affine with respect to the position of robot and landmarks. Consequently, given the orientation of the robot, the conditionally-optimal estimate for the robot's and landmarks' positions can be recovered instantly by solving a sparse linear least squares problem. We propose an algorithm to exploit this intrinsic property of SLAM, by stripping the problem down to its nonlinear core, while maintaining its natural sparsity. By taking advantage of the separable structure of SLAM, we gain a global perspective that guides the conventional local search algorithms towards a potential solution, and hence achieve a fast and stable convergence.

In the second part of this thesis, we investigate the impact of the graphical structure of SLAM and several other estimation-over-graph problems, on the estimation error covariance. We establish several connections between various graph-connectivity measures and estimation-theoretic concepts. For example, we prove that, under certain conditions, the

determinant of the estimation error covariance matrix of the maximum likelihood estimator can be estimated by counting the weighted number of spanning trees in the corresponding graph, without solving the optimization problem or using any information about the "geometry" of the scene (e.g., numerical values of the measurements). This surprising result shows that the graphical structure of SLAM provides a compact, but rich representation of the underlying estimation problem, that can be used—e.g., in decision making and active SLAM—to predict and reason about the resulting estimation error covariance. Consequently, we tackle the combinatorial optimization problem of designing sparse graphs with the maximum weighted number of spanning trees. Characterising 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 in connected graphs. By exploiting these structures, we design a complementary pair of near-optimal efficient approximation algorithms with provable performance guarantees and near-optimality certificates. We discuss several applications of our results and apply our algorithms on the measurement selection problem in SLAM to design sparse near-D-optimal (determinant-optimal) SLAM problems.

Acknowledgements

The past five years have been a mysterious journey, filled with countless moments of joy and despair, triumph and failure, optimism and pessimism, and so many other contradictory emotions. Five years ago, I couldn't have imagined the adventures, people and places that this roller coaster would take me to.

First and foremost, I would like to thank my Ph.D. supervisors, Shoudong Huang and Gamini Dissanayake, for sharing this amazing journey alongside me. Their vision, wisdom, and guidance along the way have helped me to stay on the right track. Thank you for asking the right questions at the right time, for your enduring trust, patience, and unconditional support. No words can describe how grateful I am for the invaluable lessons I have learnt from them as great mentors, and wonderful colleagues.

I am grateful to my external examiners, Professors Simon Julier and Stefan Williams, for carefully reviewing this thesis and providing me with helpful comments. I am also thankful to the Australian Research Council and UTS for funding my Ph.D. studies.

I would like to express my sincere gratitude to Gaurav S. Sukhatme for giving me the incredible opportunity to collaborate with him during my 8 memorable months of stay at USC. My interaction with Gaurav has been extremely rewarding and enjoyable for me. I have been inspired by his passion for tackling intellectually challenging problems, and I admire him for his clarity of thought. I am forever indebted to him for providing me with guidance, his generosity with his time, and long lasting support.

During the past five years I have been lucky to collaborate with many talented researchers. Special thanks go to Gibson Hu, Heng Wang, Udo Frese, and Teresa Vidal-Calleja. I would like to thank Luca Carlone for bringing many exciting aspects of SLAM to my attention through his influential contributions to this field during the past few years. Since the RSS 2015 in Rome, I have been lucky to have him as a wonderful friend.

I am grateful to John J. Leonard and Gabriel Agamennoni for hosting my seminars at MIT and ETH Zürich. I truly enjoyed the experience and my fruitful discussions with the talented members of the Marine Robotics Group and the Autonomous Systems Lab.

I was lucky to be surrounded by so many amazing friends during my stay in Los Angeles. Special thanks go to my friend Mohammad Reza Rajati whom I will never forget his generosity and kindness. I would also like to thank Alireza Divsalar for our "cross-departmental" collaborations. I am thankful

to the members of the Robotic Embedded Systems Lab for so many enjoyable moments. Particularly thanks to Lantao Liu, Karol Hausman, Stephanie Kemna, Artem Molchanov, James Preiss, Christian Potthast, Oliver Kroemer, and David Kim.

I have always been blessed with the best flatmates in the world. In Sydney, thanks go to Ainsley and Lachlan, and Kaori and Hideki. In Los Angeles, thanks go to Elisabetta Missinato and to my extraordinarily talented musician friend, Mario Sevayega, for creating countless unforgettable moments, our jam sessions, authentic Italian cuisines, and weekly movie nights.

After five magnificent years, I will be departing Sydney in the coming days. Thanks to my good friend, Syamak Castle, for a warm welcome. Special thanks to my amazing friend, Maani Ghaffari Jadidi, with whom I have shared five years of unspeakable memories. I would like to express my gratitude to my wonderful friends and labmates in CAS. In particular, special thanks to Gibson Hu, Lakshitha Dantanarayana (a.k.a. Lakipedia[1]), Raphael Falque, Kanzhi Wu, Liye Sun, Buddhi Wijerathna, Teng Zhang, Andrew To, Daobilige Su, Mahdi Hassan, Philip Quin, Linh Van Nguyen, Leo Shi, Kim Doang Nguyen, Behnam Maleki, Hasti Hayati, Teresa Vidal-Calleja, Brad Skinner, Liang Zhao, Alen Alempijevic, Jaime Valls Miro, and many others.

My journey really began when I joined the ARAS Lab in Iran as an undergraduate student. For that, I am always grateful to Hamid D. Taghirad and Ali Agha for their patience, commitment to excellence, and trust in me. I am also grateful to the SLAM community for sharing their codes and datasets through community-driven efforts such as Radish and OpenSLAM. In particular, special thanks to Rainer Kümmerle and his colleagues for g2o which played a significant role in my work.

Finally, I would like to express my deepest gratitude to my family for their selflessness, unconditional love, and never-ending support. Thanks to: my magnificent mom, Kamila, for being a ceaseless source of inspiration; my extraordinary dad, Hamid, for being my best friend; my precious sister, Sadaf, to whom I owe everything; and finally, Kourosh who is the most wonderful brother I could ever ask for. Staying apart from you has been the most difficult part of this journey. I wouldn't have been here if it weren't for your patience, love, and support. You are my personal heros. Last but not least, special thanks to my soulmate, Christine, with whom I have shared unbelievable moments of joy during the past two years. Thanks for never giving up on me. You are my rock.

[1] Wikipedia, but without any error.

To my family.

Contents

List of Figures

List of Tables

CHAPTER 1

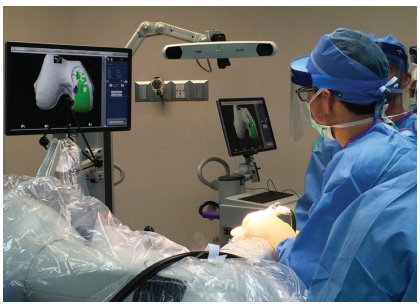
Introduction

Today, more than at any other time since the advent of the first robotic systems, *mobile robots* are being deployed in a wide variety of unknown environments. Autonomous cars, robotic vacuum cleaners, surgical robots, and Mars rovers are just few examples (Figure 1.1) These robots typically have to navigate through their environment to perform their tasks. To navigate safely in such challenging environments, a mobile robot needs to constantly estimate its location and, simultaneously, create and maintain a consistent representation of its surroundings (e.g., by building a "map"). In robotics, this problem is known as simultaneous localization and mapping (SLAM). Research on SLAM started in the 1980's—although it took another decade for the mobile robotics community to agree on a single acronym [47]. Since its early days, SLAM has always been central to mobile robotics. Considerable progress has been made during the last three decades.

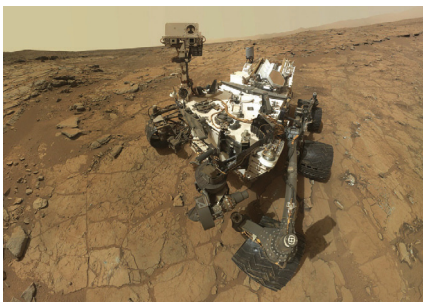
Earlier efforts were focused on building a solid foundation for a consistent estimation-theoretic framework. Soon, it became clear that the "localization" and "mapping" tasks are tightly coupled to each other in SLAM. As the robot makes more observations, strong correlations begin to emerge between the landmarks. This insight gave a truly unique character to SLAM, as such correlations had not been observed in pure mapping. This phenomenon has been referred to as the "correlations problem", mainly because maintaining and updating these correlations is computationally costly. At the time, a commonly accepted workaround was to avoid these correlations altogether by somehow eliminating the dependencies between the landmarks and robot poses (i.e., decoupling the localization and mapping problems). However, soon after this it was realized that the "curse of correlation" is in fact a blessing when it comes to the convergence: it is through these correlations that the information acquired locally is propagated across the entire map [33]. This realization is considered the first major breakthrough in SLAM that led to a generation of estimation-theoretic SLAM algorithms

(a) Google's self-driving car (Waymo)

(b) Dyson 360 eye vacuum cleaner



(c) Orthopedic Surgical robot



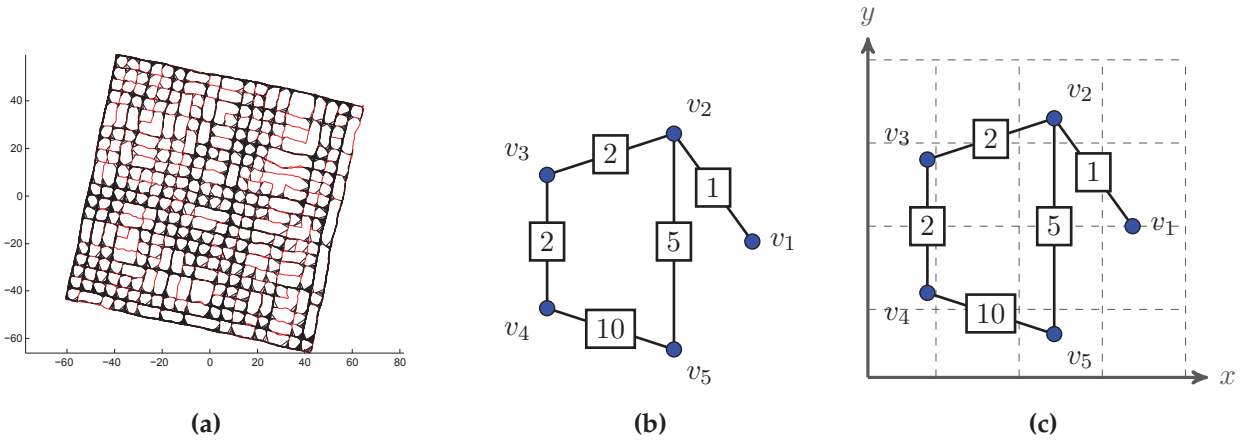
(d) NASA's Curiosity on Mars

Figure 1.1: Four applications of SLAM in different environments. In all of these applications, a moving robot deployed in an unknown environment has to answer the "where am I?" and "what does the world look like?" questions.[1]

based on the Kalman filter (KF).

KF-based algorithms are not scalable due to the curse of correlations. Moreover, the error introduced in the extended Kalman filter as a result of linearization can lead to inconsistency. Addressing these limitations was the most important goal of SLAM research in the 2000's. It had been recognised earlier that the graphical representation of SLAM problems tends to be sparse, simply because the robot normally does not observe all landmarks at every time step. However, the sparse nature of SLAM is not manifested in the covariance matrix (i.e., correlations problem). Without a doubt, the second major breakthrough in SLAM was the realization that this sparsity can be *manifested* using the canonical representation, *preserved* by not marginalizing out older robot poses, and *exploited* using modern iterative or direct sparse linear solvers. These insights are still the main principles behind today's state-of-the-art SLAM algorithms. The research on the practical aspects of SLAM (e.g., large-scale implementations and the use of various sensor modalities) flourished in the last decade. During this time, we also witnessed a broad range of contributions focused on various topics such as *reliability* against local minima and divergence, *robustness* against data association errors (outliers and spurious measurements), and *lifelong SLAM*.

[1] Images courtesy of Wikipedia (Grendelkhan), Dyson, OrthoSpineNews, and NASA.



(a) **(b)** **(c)**

Figure 1.2: Figure 1.2a shows a typical pose-graph SLAM dataset (City10K [83]). Figure 1.2b shows the graphical representation of a simple SLAM problem, where the vertices correspond to the robot poses, the edges correspond to the relative pairwise measurements, and the edge weights represent the accuracy of the corresponding measurements. Solving SLAM as an *estimation-over-graph* problem is equivalent to finding the optimal *drawing* of this graph in some geometric space (Figure 1.2c).

1.1 Motivation

Three decades of research on SLAM has resulted in scalable batch and incremental solvers that can utilize a broad range of sensor modalities. By contrast, our understanding of the theoretical aspects of SLAM has not progressed as far as it was expected. Most of the existing theoretical analyses are based on oversimplified scenarios, and rely on restrictive and unrealistic assumptions. We are currently unable to explain why our solvers succeed miraculously in solving some challenging scenarios [26, 78] while they may fail miserably in some other rather simple cases. Such empirical observations suggest that perhaps there are some other structures in SLAM waiting to be discovered and understood. This thesis is devoted to the study of such overlooked structures.

This thesis is comprised of two main parts:

1. The first overlooked structure we investigate in this thesis is the nonlinear structure of the "standard" (i.e., relative pose-pose and pose-point) measurement models used in 2D and 3D, pose-graph and feature-based SLAM (Chapter 3). Since the early days of SLAM, it has been well known that SLAM reduces to a linear estimation problem if the orientation of robot $\boldsymbol{\theta}$ is known [44]. Based on this realization, one can design an exact dimensionality-reduction scheme in which the original problem is *separated* into two smaller optimization problems. The first problem aims to find the optimal estimate for robot's orientation, $\boldsymbol{\theta}^\star$, while the second problem recovers the optimal estimate for robot's and landmarks' positions $\mathbf{p}$ given $\boldsymbol{\theta}^\star$. Figure 1.2a shows a synthetic pose-graph SLAM dataset with about $10{,}000$ poses. In this case, the

state of each pose can be described by three parameters. Hence, to find the optimal estimate of the robot's trajectory, we have to search in $\mathbb{R}^{2\cdot 10^4} \times [-\pi,\pi)^{10^4}$. In recent years, it is not uncommon to deal with large-scale datasets that are even up to an order of magnitude larger than this case. This is especially the case in feature-based SLAM; e.g., by having ℓ landmarks,[2] the search space grows to $\mathbb{R}^{2(10^4+\ell)} \times [-\pi,\pi)^{10^4}$. By exploiting the *separable* structure of SLAM, we can reduce the search space significantly down to $[-\pi,\pi)^{10^4}$, find the exact solution for $\boldsymbol{\theta}$ via local search, and easily recover the globally conditionally-optimal estimate for the rest of variables (i.e., $\mathbf{p}$) by solving a sparse convex program.

This idea is recognised in the recent literature and has been used for theoretical analysis [20, 155] and designing approximate solvers [24]. Furthermore, Zikos and Petridis [159] and Wang et al. [156] proposed to exploit this structure for solving SLAM. Zikos and Petridis [159] proposed a Rao-Blackwellised particle filter whose performance degrades over time. Wang et al. [156] proposed a dimensionality-reduction scheme, similar to the one described above. Their technique relies on a restrictive assumption on the noise covariance matrix. Interestingly, the algorithm proposed by Wang et al. [156] is a special case of the variable projection algorithm proposed by Golub and Pereyra [60] in the 1970's. Unfortunately, none of these algorithms are able to preserve the sparse structure of SLAM.[3] In Chapter 3 of this thesis, we demonstrate how to simultaneously benefit from both structures (sparsity and separability) using a modified version of the original variable projection algorithm. Exploiting sparsity guarantees scalability, while exploiting the separable structure of SLAM leads to a fast and stable convergence.

2. SLAM can be naturally represented by a graph, whose vertices correspond to robot poses and landmarks, and its edges represent pairwise measurements between the corresponding vertices. An example is shown in Figure 1.2b in which edge weights represent the accuracy of the corresponding measurements. As we can see in Figure 1.2c, from this perspective, solving the SLAM problem is equivalent to finding the optimal embedding (drawing) of this graph in some geometric space (e.g., $\mathrm{SE}(d)^n$ where each of the n poses belongs to $\mathrm{SE}(d)$). It is obvious that the topology of this graph impacts the estimation quality. Intuitively speaking, the more "connected" this graph is, the more constrained the vertices are, and hence, the smaller[4] the

[2] Note that the value of ℓ depends on how rich the sensory readings are and it can be arbitrarily large (i.e., much larger than the number of poses).

[3] Loss of sparsity is not mentioned in [159]. It is important to note that, unlike FastSLAM, the landmarks' positions are *not* conditionally independent given *only* the orientation of robot. Hence, a more accurate implementation has to employ a "large" Kalman filter with a dense covariance matrix.

[4] This is formally defined with respect to the spectrum of the covariance (or information) matrix.

resulting estimation error covariance will be. However, the exact mechanisms through which a particular "graph-connectivity measure" influences an aspect of "estimation quality" are not clear. Understanding these mechanisms enables us to predict the estimation quality to some extent by assessing the structure of the underlying graph. This knowledge can ultimately be used as a powerful tool for evaluating decisions and plans (e.g., in the context of active SLAM) based on the topology of the resulting graphs. The decisions and plans made by such an approach have the advantage of not being tied to a particular (metric) trajectory.

We investigate this structure in Chapter 4. This aspect of SLAM is mostly overlooked. The motivation behind our analysis originally comes from the work of Olson and Kaess [123], in which they insightfully pointed out the importance of graph connectivity. They empirically observed that as the average degree in pose-graphs increases, the value of the negative log-likelihood cost function at the maximum likelihood estimate approaches its value at the ground truth. Motivated by [123], first we provide a theoretical explanation for this observation. We also note that the graph Laplacian matrix appears in the derivation of the Fisher information matrix in SLAM. This observation enables us to establish additional links between the graphical and estimation-theoretic aspects of SLAM. In particular, the weighted number of spanning trees emerges as a promising connectivity measure. We prove that the weighted number of spanning trees in SLAM is closely related to the determinant of the Fisher information matrix (and consequently, that of the Cramér-Rao bound and the estimation error covariance matrix of the maximum likelihood estimator).

This result motivates us to address the synthesis problem in Chapter 5. Our goal in this chapter is to design *sparse D-optimal* (determinant-optimal) SLAM problems by designing *sparse* graphs with the maximum weighted number of spanning trees; i.e., without using the realized measurements or any knowledge about the "geometry of the scene" (e.g., true or nominal trajectory). First, we we establish several new theoretical results in this chapter, including the monotone log-submodularity of the weighted number of spanning trees. By exploiting these structures, we design a greedy-convex pair of efficient approximation algorithms with complementary characters, provable guarantees and near-optimality certificates. Although our results are originally motivated by the measurement selection problem in SLAM, our near-optimal network design framework can be readily used in a few other applications across science and engineering (such as Chemistry, RNA modelling, and the design of reliable communication networks) where maximizing the weighted number of spanning trees is desired.

1.2 Main Contributions

The main contributions of this thesis are summarized below:

◇ **Chapter 3** — [91, 93]

- Establishing new links to the rich literature on separable nonlinear least squares (NLS) problems and conditionally linear-Gaussian state-space models.
- Designing a new family of SLAM back-ends, capable of exploiting separability to achieve fast convergence while retaining the sparse structure of SLAM.

◇ **Chapter 4** — [90, 92, 94]

- Providing a theoretical explanation for the empirical observations reported in [123] regarding the role of average degree.
- Characterizing the impact of algebraic connectivity on the worst-case estimation error variance in two families of linear-Gaussian estimation problems over graphs.
- Revealing the impact of the weighted number of spanning trees on the determinant of the Fisher information matrix in SLAM and two families of linear-Gaussian estimation problems over graphs.
- Introducing the notion of "normalized tree-connectivity", which is used for comparing the graphical structure of various SLAM problems with different numbers of variables.

◇ **Chapter 5** — [95, 96]

- A number of new theoretical results, such as:
  - Introducing a new log-submodular graph invariant: weighted tree-connectivity in connected graphs.
  - Presenting a closed-form expression for computing the expected value of the determinant of random sum of rank-one matrices. A special case of this result gives the expected weighted number of spanning trees in anisotropic random graphs.
- Designing a greedy-convex pair of efficient approximation algorithms for designing sparse graphs with the maximum weighted tree-connectivity, with performance guarantees and near-optimality certificates.
- Extending our near-t-optimal graph synthesis framework and its analysis to several other formulations.

- Providing a novel probabilistic explanation for the proposed convex relaxation graph synthesis algorithm, and its extension to the more general case that had been studied, e.g., by Joshi and Boyd [81] and in the theory of optimal experimental design.
- Presenting a new probabilistic justification for the deterministic rounding scheme used in our graph synthesis algorithm, and its extension to the more general case that had been studied by Joshi and Boyd [81].
- Proposing a novel graph-theoretic near-D-optimal measurement selection framework for SLAM.

1.3 The Structure of the Thesis

This thesis is organized as follows:

- **Chapter 2**: In this chapter, we review some of the key contributions made over the last three decades that have improved the understanding of SLAM or led to significant breakthroughs.
- **Chapter 3**: In this chapter, we investigate the nonlinear structure of the measurement models in SLAM and propose a new algorithm that can benefit from exploiting the separable structure of SLAM while preserving the sparse structure of the problem.
- **Chapter 4**: In this chapter, we study the impact of several graph connectivity measures on several estimation-theoretic concepts.
- **Chapter 5**: In this chapter, we develop a near-optimal graph synthesis framework for designing graphs with strong weighted tree-connectivity. We apply our framework to the measurement selection problem in SLAM.
- **Chapter 6**: Finally, in this chapter we discuss challenges, open directions and our future work.
- **Appendix A**: In this chapter, we briefly review some preliminaries.
- **Appendix B**: In this chapter, we present extended proofs for our theoretical results.

1.4 Publications

The contents of this thesis are partially based on the following papers:

Journal Papers

1. "**A Sparse Separable SLAM Back-End**" — IEEE Transactions on Robotics (2016)
   Kasra Khosoussi, Shoudong Huang and Gamini Dissanayake.

2. "**Dimensionality Reduction For Point Feature SLAM Problems With Spherical Covariance Matrices**" — Automatica (2015)
   Heng Wang, Shoudong Huang, Kasra Khosoussi, Udo Frese, Gamini Dissanayake and Bingbing Liu

Conference Papers

1. "**Designing Sparse Near-D-Optimal Pose-Graph SLAM Problems: A Graph-Theoretic Approach**" — International Workshop on the Algorithmic Foundations of Robotics (WAFR) 2016
   Kasra Khosoussi, Gaurav S. Sukhatme, Shoudong Huang and Gamini Dissanayake.

2. "**Tree-Connectivity: Evaluating the Graphical Structure of SLAM**" — IEEE International Conference on Robotics and Automation (ICRA) 2016
   Kasra Khosoussi, Shoudong Huang and Gamini Dissanayake.

3. "**Exploiting the Separable Structure of SLAM**" — Robotics: Science and Systems (RSS) 2015
   Kasra Khosoussi, Shoudong Huang and Gamini Dissanayake.

4. "**Novel Insights into the Impact of Graph Structure on SLAM**" — IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2014
   Kasra Khosoussi, Shoudong Huang and Gamini Dissanayake.

5. "**Towards a Reliable SLAM Back-End**" — IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2013
   Gibson Hu, Kasra Khosoussi and Shoudong Huang.

Referred Workshop Papers

1. "**Good, Bad and Ugly Graphs for SLAM**" — Robotics: Science and Systems (RSS) 2015 Workshop on The Problem of Moving Sensors
   Kasra Khosoussi, Shoudong Huang and Gamini Dissanayake.

Preprints

1. "**Maximizing the Weighted Number of Spanning Trees: Near-t-Optimal Graphs**" — arXiv:1604.01116
   Kasra Khosoussi, Gaurav S. Sukhatme, Shoudong Huang and Gamini Dissanayake.

1.5 General Notation

Throughout this thesis, 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 $\{i \in \mathbb{N} : i \leq n\}$. The eigenvalues of symmetric matrix $\mathbf{M}$ are denoted by $\lambda_{\min}(\mathbf{M}) = \lambda_1(\mathbf{M}) \leq \cdots \leq \lambda_n(\mathbf{M}) = \lambda_{\max}(\mathbf{M})$. $\mathbf{1}$, $\mathbf{I}$ and $\mathbf{0}$ denote the vector of all ones, the identity and the zero matrices with appropriate sizes, respectively. $\mathbf{S}_1 \succ \mathbf{S}_2$ (resp. $\mathbf{S}_1 \succeq \mathbf{S}_2$) means $\mathbf{S}_1 - \mathbf{S}_2$ is positive definite (resp. positive semidefinite). $\|\cdot\|$ denotes the Euclidean norm, and the weighted Euclidean norm of vector $\mathbf{e}$ with respect to matrix $\mathbf{W} \succ \mathbf{0}$ is denoted by $\|\mathbf{e}\|_{\mathbf{W}} \triangleq \sqrt{\mathbf{e}^\top \mathbf{W}\, \mathbf{e}}$. $\mathbb{S}^n_{>0}$ and $\mathbb{S}^n_{\geq 0}$ denote the sets of positive definite and positive semidefinite $n \times n$ matrices, respectively. Kronecker product is denoted by $\otimes$. $\operatorname{diag}(\mathbf{W}_1, \ldots, \mathbf{W}_k)$ is the block-diagonal matrix whose main diagonal blocks are $\mathbf{W}_1, \ldots, \mathbf{W}_k$. Finally, see Appendix A for our domain-specific notations.

CHAPTER 2

Simultaneous Localization and Mapping: A Survey

The journey of SLAM research began with the search for rigorous and consistent estimation-theoretic formulations. Today, after three decades, we have a deeper understanding of the problem and its intrinsic structures, a number of efficient solvers based on several different paradigms, many successful applications in a wide range of scenarios in and beyond robotics, and several highly-optimized, robust, free and open source (FOSS) implementations. In this chapter, we look back at the history of SLAM and highlight some of the key contributions that have been pivotal to the success of SLAM research. In particular, we describe the existing paradigms and explain the insights that led to their creation. Our survey mainly focuses on how SLAM has been solved as an estimation/optimization problem (i.e., SLAM back-end). Hence, we do not cover several other crucial aspects of SLAM research such as data association or sensor-specific solvers; for those aspects of SLAM see the comprehensive surveys due to Durrant-Whyte and Bailey [47] [4] and Cadena et al. [17]. In addition to this chapter, we review more specific related works separately in each chapter.

2.1 Classic Paradigms

Robots observe the environment and interact with it through their sensors and actuators. For obvious reasons, these sensors and actuators are imperfect and noisy. Moreover, in many real applications, the environment is often not fully known in advance. Furthermore, to limit the complexity of our models, they are often incomplete and only capture the most significant aspects of robot motion and sensing. Such errors are typically unpredictable and exhibit "random" behaviors. Probabilistic

robotics [146] is a well-established robotic paradigm in which perception and control errors are interpreted as *uncertainty* and are captured with *probability distributions*. An important lesson learned from the success of probabilistic methods in robotics in the past 30 years is that uncertainty can neither be ignored nor treated as a secondary issue; it has to be (i) adequately represented, (ii) maintained as new bits of information arrive or when the state evolves, and (iii) taken into account during reasoning and decision making (see [146] and the references therein).

This is why, first and foremost, SLAM is a statistical estimation problem, whose goal is to estimate spatial quantities (e.g., robot pose, landmark position) from noisy sensor measurements and imperfect models. Estimation-theoretic approaches to SLAM started with Smith and Cheeseman [138] and Durrant-Whyte [46] who developed basic frameworks and guidelines for dealing with spatial and geometric uncertainty that arise when an imperfect mobile sensor (e.g., mounted on a mobile robot) observes a set of landmarks. Smith et al. [139] later developed one of the first estimation-theoretic SLAM frameworks by introducing what they called a *stochastic map* (see also the work by Moutarlier and Chatila [114]). They employed the extended Kalman filter (EKF) [2] to maintain and incrementally update a Gaussian belief. Their framework ultimately became the basis for the first generation of SLAM solutions known as EKF-SLAM algorithms [47]. A distinct feature of SLAM—that is *not* present in pure mapping (i.e., known robot pose)—is the *correlation* between the estimates of every landmark position. These correlations spread *through the robot pose* every time a landmark is observed. Note that in pure mapping, the robot pose is known and therefore the estimates of the landmarks remain statistically independent (assuming independence between the measurements' noise). Maintaining and updating these inter-dependencies is costly. This is one of the critical limitations of EKF-SLAM. Therefore, it was widely assumed that the right strategy is to eliminate such undesired inter-dependencies [47]. Although the existence of such inter-dependencies was recognised in [139], their key role in the convergence of EKF was not fully understood at the time. For example, Leonard and Durrant-Whyte [104] addressed the "correlations problem" by proposing a lossy strategy for decoupling the estimates of the landmarks (or, equivalently, decoupling localization and mapping) by controlling the spread of correlations. Roughly speaking, they do so by breaking SLAM into a sequence of almost-pure localization and almost-pure mapping whenever, respectively, the uncertainty in the estimates of landmarks and poses is less than some thresholds—otherwise, they discard the measurement. It was in the late 90's that Csorba [33] theoretically demonstrated that these correlations are, in fact, essential for the convergence of EKF-SLAM. It is through the correlations between the estimates of the landmarks that observing any single landmark, leads to the spread of informa-

tion to the *entire map*. Subsequently, Dissanayake et al. [38] shed more light on the convergence and steady-state behaviour of Kalman filter-based simultaneous localization and mapping by proving the following results—albeit for the linear-Gaussian case (see [38] for the exact model):

1. The determinant of any submatrix of the map covariance matrix decreases monotonically as successive observations are made.

2. In the limit, the landmark estimates become fully correlated.

3. In the limit, the lower bound on the covariance matrix associated with any single landmark estimate is determined only by the initial covariance in the vehicle estimate at the time of the first sighting of the first landmark.

The work by Dissanayake et al. [38] is also remarkable for describing a complete EKF-SLAM implementation and addressing important problems such as map management, data association and landmark initialization.

EKF-SLAM has two majors drawbacks: (i) computational cost, and (ii) inconsistency. The time and space complexity of EKF-SLAM are both quadratic in the number of landmarks. Hence, a naive implementation of EKF-SLAM is limited to "small" environments (e.g., 100 landmarks). The vast majority of the measurement and motion models in SLAM are nonlinear. Linearization errors in EKF can lead to strange outcomes and inconsistent (overconfident) estimates. This phenomenon was first observed by Julier and Uhlmann [82] and was further investigated empirically and theoretically in [5, 75, 76].

Submapping SLAM algorithms address these concerns by using EKF-SLAM in confined local maps; see, e.g., [14, 49, 105, 145, 157]. These local maps are then fused together—usually with some approximation—to obtain a global map. By limiting the number of landmarks in any single local map, building each local map takes $O(1)$ time. The number of local maps grows linearly with the number of landmarks. Submapping algorithms can also improve local consistency by preventing large global uncertainties; if the uncertainty in any local map reaches a certain threshold, we simply start a new local map. Following this paradigm, Guivant and Nebot [66] proposed compressed EKF-SLAM (CEKF-SLAM). In this method, observations are first used to update the current local map. The information acquired by observing the landmarks in the local region is ultimately fused into the global map when the robot leaves the local region [65].

The first Rao-Blackwellised particle filtering SLAM (RBPF-SLAM) algorithm is due to Murphy [115]. Particle filters—also known as sequential Monte Carlo (SMC) methods [39, 43]—had been

successfully used before [115] for pure localization in Monte Carlo localization (MCL) [36]. In [115], the environment is represented by an occupancy grid and the robot lives in a discrete space, i.e., the grid world. The key insight in [115] is the realization that given the trajectory of robot, map elements are *conditionally independent* from each other. As we noted earlier, this is the key difference between pure mapping and SLAM. In RBPF [42, 115], the state variables (up to time t) are partitioned into two sets $\mathbf{x}^{\circ}_{1:t}$ and $\mathbf{x}^{\diamond}_{1:t}$. Then, a particle filter—e.g., sequential importance resampling (SIR) filter [39]—is used to describe the marginal posterior distribution $p(\mathbf{x}^{\circ}_{1:t}|\mathbf{z}_{1:t})$, while $p(\mathbf{x}^{\diamond}_{1:t}|\mathbf{x}^{\circ}_{1:t},\mathbf{z}_{1:t})$ is handled analytically. Note that these two distributions, together, give the full posterior distribution,

$$p(\mathbf{x}^{\diamond}_{1:t},\mathbf{x}^{\circ}_{1:t}|\mathbf{z}_{1:t}) = p(\mathbf{x}^{\circ}_{1:t}|\mathbf{z}_{1:t})\, p(\mathbf{x}^{\diamond}_{1:t}|\mathbf{x}^{\circ}_{1:t},\mathbf{z}_{1:t}). \tag{2.1}$$

Representing distributions by weighted samples can be highly inefficient in high dimensional spaces (curse of dimensionality) [39, 41, 43]. Hence, in comparison to using a particle filter for handling the entire posterior $p(\mathbf{x}^{\diamond}_{1:t},\mathbf{x}^{\circ}_{1:t}|\mathbf{z}_{1:t})$, RBPF reduces the dimensionality of the particle filter. More importantly, partitioning the state variables has to be done such that representing $p(\mathbf{x}^{\diamond}_{1:t}|\mathbf{x}^{\circ}_{1:t},\mathbf{z}_{1:t})$ analytically is "convenient"—e.g., (i) when exact inference is possible, and/or (ii) when $p(\mathbf{x}^{\diamond}_{1:t}|\mathbf{x}^{\circ}_{1:t},\mathbf{z}_{1:t})$ factorizes based on conditional independence. Therefore, it is crucial to look for structures that may exist in the problem before partitioning the variables in RBPF. Two classic examples are conditionally linear-Gaussian state-space models and conditionally finite state-space hidden Markov models (for discrete variables); see [42, 115]. Subsequently, Montemerlo et al. [113] in FastSLAM applied the idea of Rao-Blackwellised particle filtering to the conventional formulation of SLAM that had been used before in EKF-SLAM; i.e., where robot pose takes continuous values and the map is represented by a set of landmarks in $\mathbb{R}^d$ ($d \in \{2,3\}$). By exploiting the aforementioned conditional independence property of SLAM, FastSLAM breaks the $O(n^2)$ correlations barrier (for maintaining correlations between the estimates of the landmarks) in EKF-SLAM (here n being the number landmarks). For each (weighted) particle of the robot's trajectory, FastSLAM solves a pure mapping problem by maintaining n separate EKF (one for each landmark), each of which can be updated in $O(1)$ time. Montemerlo et al. [113] described an efficient implementation that requires $O(k \log n)$ time and $O(k\, n)$ space where k is the number of particles. Another advantage of exploiting the conditional independence in FastSLAM is that, unlike EKF-SLAM, FastSLAM naturally tracks parallel data association hypotheses (one for each particle). Nevertheless, FastSLAM, like any other particle filtering algorithm, suffers from two inevitable drawbacks, namely the filter degeneracy and sample impoverishment problems

[39, 41, 43]. After running the particle filter for a few iterations, we observe that one particle will have a normalized weight close to one, while the rest of the particles have near-zero normalized weights. This problem is attributed to the increasing unconditional variance of the importance weights [43]. As a remedy, in SIR the filter is "reset" by resampling particles. However, resampling destroys diversity. Lack of diversity is the second major defect in sequential Monte Carlo methods and is often referred to as the sample impoverishment problem. A sensible strategy for delaying degeneracy is to choose the proposal distribution at time t such that the conditional variance of the importance weights is minimized. This distribution is referred to as the *optimal proposed distribution*. In FastSLAM 2.0, Montemerlo and Thrun [112] improved their earlier work by sampling from the optimal proposal distribution. For the nonlinear measurement models of SLAM, this distribution and the corresponding weight function cannot be computed in closed form. Consequently, FastSLAM 2.0 approximates these terms with Gaussian density functions by linearizing the nonlinear measurement model. Later, Hahnel et al. [69], Eliazar and Parr [48], and Grisetti et al. [62] developed similar RBPF-SLAM algorithms using occupancy grid maps. Despite these and other efforts (see, e.g., [43]), degeneracy and sample impoverishment are two inevitable drawbacks of SMC methods that, for a fixed number of particles, sooner or later will occur and, therefore, will affect the performance of the filter [41].

From an estimation-theoretic point of view, EKF-SLAM and its numerous variants [38, 47] are classified as *filtering* methods [2]; i.e., at time t, the momentary robot pose at time t and time-invariant locations of stationary landmarks are estimated using the data collected up to time t. The filtering methods in SLAM are also known as *online SLAM* algorithms [146]. Alternatively, in the *smoothing* techniques, the entire trajectory is estimated at every time step using the data collected up to that time [2]. Clearly smoothing methods have the advantage of utilizing future observations to improve the estimates of older states. Hence, it is sensible to expect that smoothing methods outperform the corresponding filtering approaches in terms of accuracy (i.e., error covariance matrix). This advantage, however, *usually* comes at the cost of increased computational efforts, mainly because the dimension of the state space grows linearly with time. Estimating the entire robot trajectory and the time-invariant locations of stationary landmarks in SLAM is referred to as *full SLAM* [146]. For example, FastSLAM [112] is considered a full SLAM algorithm—recall that each particle is a sample of the entire robot trajectory.

Another family of online SLAM algorithms is based on the extended information filter (EIF). (Extended) information filter is the dual of (extended) Kalman filter, in which instead of tracking the (approximately) Gaussian belief with its mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, the same belief is de-

scribed and tracked by the *information vector* $\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$ and *information*—or, *precision—matrix* $\boldsymbol{\Sigma}^{-1}$. This representation is referred to as the *canonical representation*. There is an elegant duality between the prediction and update stages in EKF and EIF (see, e.g., [55, 148]). Another interesting property of this representation is that the sparsity pattern of the information matrix of the (approximately) Gaussian belief captures the conditional independence relations among the variables; i.e., the (i,j) entry of the information matrix is zero if and only if (iff) the corresponding variables are conditionally independent given other variables [133]. Note a zero (i,j) entry in the covariance matrix (of a Gaussian vector) implies that the corresponding variables are independent. As noted in [148], the sparsity pattern encodes the adjacency pattern of the associated Gaussian Markov network (Gaussian Markov random field) $\mathcal{G} = (\mathcal{V},\mathcal{E})$; i.e., for all $i,j \in \mathcal{V}$, i and j are adjacent iff the (i,j) entry of the information matrix is non-zero. The time complexity of a naive implementation of EIF-SLAM is cubic in the number of landmarks.

The significance of EIF-SLAM goes beyond being merely a dual of EKF-SLAM. It was first empirically observed and reported by Thrun et al. [148] that the normalized precision matrix in online SLAM is "almost sparse"; i.e., many off-diagonal entries are significantly smaller than the corresponding diagonal terms. Frese [54] later provided a proof for this phenomenon. Based on this observation, Thrun et al. [148] proposed an approximation of the EIF algorithm called sparse extended information filter (SEIF), in which the information matrix is *sparsified* by systematically zeroing out some of the small entries (i.e., creating new conditional independence relations that did not exist in the original SLAM problem). Through this approximation, SEIF can achieve an approximate solution in constant time [148].

SEIF demonstrated the importance of maintaining (with approximation) and exploiting sparsity. Moreover, Thrun et al. [148] realized that the *fill-in*—i.e., new non-zero blocks in the information matrix—is created in the prediction stage (also known as, motion update) of EIF, where the old robot pose $\mathbf{x}_{t-1}$ is marginalized out (after augmenting the new robot pose $\mathbf{x}_t$). In the SLAM literature, the connection between marginalization and fill-in in the information matrix had been insightfully noted earlier by Frese and Hirzinger [55] through the notion of Schur complement. Therefore, it can be concluded that exact sparsity of the information matrix is a natural consequence of maintaining robot trajectory in the state vector of SLAM—i.e., taking a smoothing approach to SLAM, or full SLAM. See [50, 55] for a discussion on this topic. Although smoothing suffers from the aforementioned computational drawbacks, it brings (exact) sparsity (of the information matrix) which helps to break the correlation barrier in EKF-SLAM. It is important to note that, as in online SLAM, the

covariance matrix in full SLAM is generally dense (i.e., estimates of the robot trajectory and the locations of landmarks are still highly correlated). It is also noteworthy that the computational efficiency of FastSLAM—as a smoothing approach to SLAM—is also a consequence of this fact (albeit with a different representation and by directly factorizing the smoothing posterior). As mentioned earlier, in Gaussian distributions, the (i,j) entry of the information matrix is zero if and only if the corresponding variables are conditionally independent given the rest of variables. Hence, sparsity of the information matrix in full SLAM can be seen as the consequence of the conditional independence property that was exploited in FastSLAM; i.e., estimates of landmarks are conditionally independent given the robot trajectory.

Note that, at a more fundamental level, sparsity in SLAM is mainly due to the "local" nature of observations and is affected by various factors such as the robot's sensing field and motion. For instance, the smoothing approach to SLAM would no longer be sparse if the robot can observe a substantial fraction of landmarks at any time, or if the measurement noise samples affecting the measurements are not statistically independent. These conditions generally hold in the SLAM problems that emerge from practical scenarios.

2.2 Modern Paradigms

The smoothing formulation of feature-based SLAM is closely related to the *bundle adjustment* problem in photogrammetry and computer vision; see [149] for a comprehensive survey. According to Triggs et al. [149], this problem has a rich history that dates back to the 1950's. Many of the techniques and ideas that originated from bundle adjustment, resurfaced in the SLAM literature (e.g., using sensors other than camera) in the past 10 years.

Lu and Milios [108] pioneered the smoothing approach to SLAM. They used raw laser scans and odometry measurements to create pairwise pose relations. In [108], odometry data were transformed into relative-pose measurements. Moreover, additional pairwise pose relations were created from laser scans by performing scan matching between the corresponding laser scans. They used a network to represent this problem, whose nodes represented robot poses and edges corresponded to pairwise pose relations. Lu and Milios [108] solved the estimation problem by formulating a nonlinear least squares problem. Note that, as explained above, due to the smoothing nature of this work the resulting nonlinear least squares can be solved efficiently by exploiting the sparse structure of SLAM. The global minimizer of this problem corresponds to the maximum likelihood estimate un-

der the assumption of additive Gaussian noise. In today's SLAM literature, this approach is referred to as *pose-graph SLAM* or *pose-graph optimization*; several other names have been used as well, including *view-based* SLAM as opposed to *feature-based* SLAM, and *delayed-state* SLAM [51]. This work was extended by Gutmann and Konolige [68], who used Lu and Milios [108]'s algorithm, made it incremental (i.e., incrementally detecting new pairwise relations) and proposed a more reliable *loop closure* detection. Their method successfully built dense maps of indoor environments using raw laser scans. Gutmann and Konolige [68] also noted that the resulting pose-graph is sparsely connected, and suggested making use of this structure.

Over the past 10 years, numerous solvers have been proposed and applied to the smoothing formulation of—both feature-based and pose-graph—SLAM. Duckett et al. [44] employed Gauss-Seidel relaxation for solving a linear instance of SLAM with known robot orientation. The original nonlinear problem was considered in [45]. Subsequently, Frese et al. [56] addressed the slow convergence of Gauss-Seidel [45] (when closing large loops) by proposing a *multilevel* relaxation scheme inspired by the multigrid methods used for solving differential equations. Folkesson and Christensen [53] incorporated data association in the optimization problem and suggested to occasionally prune the nodes of the graph. Graph sparsification by pruning nodes and edges is critical in large problems that emerge from lifelong SLAM [19, 74, 99, 110, 152].[1] The idea of incorporating data association in the optimization process was later investigated by Olson and Agarwal [122] and Sunderhauf and Protzel [144] which led to the development of robust SLAM solvers; see, e.g., [1, 25, 103].

Thrun and Montemerlo [147] proposed a full feature-based SLAM. They noted that the landmarks can be efficiently marginalized out. Exploiting this structure is especially useful when the number of landmarks is much greater than the number of robot poses. This technique is known as the Schur-complement trick and had been known to the computer vision community [149]. They employed the conjugate gradient descent method for solving the linear system.

Dellaert and Kaess [35] in square root smoothing and mapping (SAM) adopted direct sparse solvers based on sparse QR and Cholesky decomposition of the Jacobian and information matrix, respectively. They emphasized employing a good (heuristic) fill-reducing variable ordering. Subsequently, Kaess et al. [83] in incremental smoothing and mapping (iSAM) employed an incremental sparse QR factorization method using the Givens rotations. They demonstrated that periodic variable reordering and relinearization can effectively control fill-in and provide a good convergence to the optimal estimate. Such periodic reordering and relinearization steps are basically equivalent to

[1] We survey these works in Chapter 5.

batch iterations. The need for these inefficient batch iterations is eliminated in iSAM2 [85], in which variable reordering and relinearization is done incrementally based on insights from graphical models. Rosen et al. [131] proposed a similar trust-region incremental solver based on Powell's Dog-Leg algorithm [127].

Olson et al. [124] proposed a pose-graph optimization algorithm based on the stochastic gradient descent method with an incremental parameterization $\mathbf{x}_i^{\text{inc}} \triangleq \mathbf{x}_i - \mathbf{x}_{i-1}$. Their parameterization provides a simple way for capturing the "cumulative nature" of robot's motion. Grisetti et al. [63] extended this work by parameterizing nodes based on a spanning tree (other than the odometry path). Both works [63, 124] are reported to be robust against bad initial guess, while [63] is computationally more efficient.

Konolige et al. [98] described an efficient implementation of pose-graph SLAM. They employed a variant of the Levenberg-Marquardt algorithm and used sparse Cholesky factorization for solving the damped normal equations. Kuemmerle et al. [102] subsequently proposed another efficient implementation. Strasdat et al. in [142, 143] compared the computation cost and accuracy of filtering and smoothing approaches in the context of real-time visual SLAM. Based on their accuracy/cost analysis, it was concluded that filtering is the right choice only in the case of small processing budgets. Furthermore, they reported that increasing the number of features is generally more "profitable" than increasing the number of poses (i.e., camera frames). Strasdat et al. [143] noted that smoothing approaches can handle a large number of features using the aforementioned Schur-complement trick and by exploiting the sparse structure of SLAM.

The computational efforts needed when using sparse direct solvers can be prohibitive in "very large" (i.e., say more than 10^5 measurements) SLAM problems that arise in lifelong SLAM. Dellaert et al. [37] explored the idea of combining iterative (preconditioned conjugate gradients) and direct linear solvers. An alternative approach is to approximate the problem by actively pruning the graph (e.g., by pruning edges or marginalizing out nodes). This thesis reviews these methods in Chapter 5. Another class of solutions are based on a modern interpretation of the original filtering submapping algorithms; see, e.g., [77, 119].

Solving the nonlinear least squares problem in SLAM needs an initialization scheme to provide an initial guess. The task of finding an initial guess for iterative optimization methods is referred to as *bootstrapping* [73]. An ideal bootstrapping method needs to be (i) fast, and (ii) able to find a reliable initial guess (i.e., "sufficiently close" to the ML/MAP estimate). The simplest bootstrapping technique is dead reckoning (i.e., composing odometry readings). While dead reckoning is fast,

the resulting initial guess can be extremely unreliable due to the accumulation of error. Konolige et al. [98] addressed this drawback by composing pairwise measurements (either odometry or loop closures) along the shortest-path spanning tree rooted at the anchored node (i.e., origin). Carlone et al. [23] [24] proposed linear approximation for 2D graph optimization (LAGO). LAGO breaks the nonlinear least squares into two linear least squares subproblems; one for estimating the orientations and the other for estimating positions (given an estimate for robot's orientation). Motivated by LAGO, the problem of finding the maximum likelihood estimate of robot orientation in 2D pose-graphs from relative angular measurements was carefully investigated by Carlone and Censi [21]. Hu et al. [73] proposed a sequential bootstrapping framework, in which each bootstrapping block initialises the next block, until at the end a reliable initial guess for the original problem is obtained. They demonstrated that this idea can be implemented by using M-estimators with a re-descending influence function [73]. Liu et al. [106] proposed a convex relaxation approach by formulating the 2D pose-graph optimization problem as a quadratically constrained quadratic program (QCQP) which admits a straightforward semidefinite programming (SDP) convex relaxation. They mapped the solution of the convex program into a feasible suboptimal estimate for SLAM, and used that estimate to initialize conventional nonlinear least squares solvers. More recently, Rosen et al. [132] proposed a similar approach by using a different convex relaxation for the more general case of M-estimation.

Huang et al. [78] reported surprisingly good convergence results from poor initial values in the case of spherical (isotropic) noise covariance matrices. Subsequently, Wang et al. [154] [79, 155, 156] analyzed the number of local minima in simple scenarios under isotropic noise. They proved that the simplified problems have at most two minima, one of which appears only when the data are extremely noisy. They also developed a dimensionality-reduction scheme by exploiting the partially-linear structure of standard measurement models. Carlone [20] analyzed the convergence of Gauss-Newton in 2D pose-graph SLAM with isotropic noise. He concluded that the global convergence is influenced by the graph structure, inter-nodal distances and the covariance of measurement noise. Carlone and Dellaert [22] derived the Lagrange dual of a modified version of the nonlinear least squares cost function of 2D pose-graph SLAM. Based on Lagrange duality, solving the dual problem—in this case, a SDP—gives a lower bound on the minimum of the primal problem. Hence, the optimal value of the dual problem can be used to assess the quality of any potential solution found by iterative solvers. Carlone and Dellaert [22] also empirically observed that in many 2D pose-graph SLAM problems, the duality gap is zero when the orientation noise is "reasonable". In such cases, the dual problem can be used to *verify* whether a potential solution found by iterative

optimization methods is the true ML estimate. It is worth noting that the SDP relaxation scheme proposed by Liu et al. [106] coincides with the Lagrange dual of their QCQP problem [15]. Recently, Carlone et al. [26] extended their earlier work by presenting sufficient conditions to verify whether strong duality holds in the modified 2D pose-graph SLAM. If this condition holds, they can find the optimal estimate efficiently (i.e., without solving the dual problem).

2.3 Nonlinear Least Squares Formulation

In this section, we present a general formulation of the nonlinear least squares approach to full SLAM. A more detailed problem formulation is presented in each of the following chapters. Let $\mathbf{x}$ represent the state vector. In the Bayesian approach, $\mathbf{x}$ is modeled as a random vector with a Gaussian prior $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}})$. Moreover, the measurement model is assumed to be given by $p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mathbf{h}(\mathbf{x}), \boldsymbol{\Sigma})$. The maximum *a posteriori* (MAP) estimate is the maximizer of the posterior density,

$$p(\mathbf{x}|\mathbf{z}) \propto p(\mathbf{z}|\mathbf{x})\ p(\mathbf{x}). \tag{2.2}$$

Therefore, the MAP estimate $\hat{\mathbf{x}}_{\text{MAP}}$ is given by

$$\begin{aligned}
\hat{\mathbf{x}}_{\text{MAP}} &\triangleq \arg\max_{\mathbf{x}}\ p(\mathbf{x}|\mathbf{z}) && (2.3)\\
&= \arg\min_{\mathbf{x}}\ -\log p(\mathbf{x}|\mathbf{z}) && (2.4)\\
&= \arg\min_{\mathbf{x}}\ -\log p(\mathbf{z}|\mathbf{x}) - \log p(\mathbf{x}) && (2.5)\\
&= \arg\min_{\mathbf{x}}\ \|\mathbf{z} - \mathbf{h}(\mathbf{x})\|^2_{\boldsymbol{\Sigma}^{-1}} + \|\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}}\|^2_{\boldsymbol{\Sigma}_{\mathbf{x}}^{-1}}. && (2.6)
\end{aligned}$$

In the absence of an informative prior over $\mathbf{x}$,[2] one may seek the maximizer of the likelihood function $p(\mathbf{z}|\mathbf{x})$ in order to obtain the ML estimate $\hat{\mathbf{x}}_{\text{ML}}$. Dropping the prior in (2.6) gives $\hat{\mathbf{x}}_{\text{ML}}$,

$$\begin{aligned}
\hat{\mathbf{x}}_{\text{ML}} &\triangleq \arg\max_{\mathbf{x}}\ p(\mathbf{z}|\mathbf{x}) && (2.7)\\
&= \arg\min_{\mathbf{x}}\ \|\mathbf{z} - \mathbf{h}(\mathbf{x})\|^2_{\boldsymbol{\Sigma}^{-1}}. && (2.8)
\end{aligned}$$

It is fairly rare to have an informative prior over $\mathbf{x}$ in practice. To simplify our notation, we often use $\mathbf{x}^\star$ for referring to $\hat{\mathbf{x}}_{\text{ML}}$ or $\hat{\mathbf{x}}_{\text{MAP}}$.

A standard iterative method for solving (2.6) and (2.8) is to use the Gauss-Newton method. First,

[2] In this case, $\mathbf{x}$ is treated as the vector of unknown deterministic *parameters* in the measurement model.

we need an initial guess $\mathbf{x}_0$ that is "sufficiently close" to $\mathbf{x}^\star$. Then, in the ith iteration of Gauss-Newton, the measurement model is linearized by computing the Jacobian matrix $\mathbf{J}_i \triangleq \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}$ evaluated at $\mathbf{x} = \mathbf{x}_i$. What remains after linearization is a linear least squares problem which can be solved by solving the so-called normal equations,

$$\mathbf{J}_i^\top \boldsymbol{\Sigma}^{-1} \mathbf{J}_i \delta \mathbf{x}_i = \mathbf{J}_i^\top \boldsymbol{\Sigma}^{-1} \Big(\mathbf{z} - \mathbf{h}(\mathbf{x}_i) \Big) \tag{2.9}$$

and updating the current estimate according to $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \cdot \delta \mathbf{x}_i$ where α_i is a suitable step size (e.g., satisfying the Armijo rule or Wolfe conditions; see [120]). This process is repeated until we converge to a (locally optimal) solution. So far, we implicitly assumed that both $\mathbf{x}$ and $\mathbf{z}$ belong to the Euclidean space (where, e.g., vector addition and subtraction are well-defined operations), which is not an accurate assumption as both are comprised of rotational components. As it will become clear shortly, this problem can be avoided to some extent in 2D. Nevertheless, this problem has been investigated recently in the SLAM literature; see, e.g., [6, 70, 141]. A common approach is to rewrite (2.9) and the update rule such that $\delta \mathbf{x}_i$ lives in a tangent space [70, 141] and is mapped back to the manifold using the exponential map. We discuss this idea in Chapter 3.

2.4 Summary

We presented a survey of some of the key contributions in SLAM during the past 30 years. To sum up, two fundamental insights that led to significant breakthroughs are the following:

1. The correlations among the elements of the "stochastic map" in SLAM is inevitable. Furthermore, to maintain consistency, these correlations cannot, and should not, be ignored [33, 38].

2. The sparse structure of SLAM emerges from the canonical representation (information matrix) in full SLAM (smoothing). Exploiting sparsity is crucial for computational efficiency.

Part I

SLAM: Sparse Separable Nonlinear Least Squares

CHAPTER 3

A Sparse Separable SLAM Back-End

As we saw in Chapter 2, under fairly standard assumptions, the maximum likelihood (ML) and maximum *a posteriori* (MAP) estimation problems in SLAM can be formulated as nonlinear least squares (NLS) problems. Given a "sufficiently good" initial guess—obtained using, e.g., a reliable bootstrapping method such as [73]—the NLS problem can be solved using iterative schemes such as Gauss-Newton (GN) or Levenberg-Marquardt (LM). In these methods, the measurement function is approximated by its first-order Taylor expansion around the current estimate. Newton-based solvers treat the measurement function as a *generic sufficiently smooth nonlinear function* of state variables. However, the functions associated to the standard relative pose-point and pose-pose measurement models (in 2D and 3D) has a noticeable structure: these functions are *affine* with respect to the x, y (and z) coordinates of robot's and landmark's positions. Hence, given the robot orientation's throughout its trajectory, $\boldsymbol{\theta}$, the optimal choice in the ML/MAP sense for positions $\mathbf{p}$ can be obtained by solving a sparse *linear least squares* problem.

Our goal in this chapter is to investigate this overlooked structure of SLAM. Exploiting this structure—known as *separability*—will help to speed up the optimization process by reducing the number of iterations needed for converging to a locally optimal solution. Unfortunately, naively exploiting separability will significantly increase the computational cost of each iteration due to the loss of sparsity. In this chapter, we propose an algorithm that combines the advantages of exploiting both structures (i.e., cheap iterations and fast convergence) by exploiting separability and sparsity at the same time.

Outline

Section 3.1 introduces several related works. In Section 3.2 we provide a mathematical formulation of SLAM. Section 3.3 describes how the variable projection algorithm can be applied to exploit the separable structure of SLAM. We propose a new algorithm in Section 3.4 to exploit both separability and sparsity. Results obtained using both synthetic and real datasets are presented in Section 3.8. This is followed by a discussion and a summary in Section 3.9 and Section 3.10, respectively.

3.1 Related Works

In the least squares literature, NLS problems with partially linear residuals are called *separable* [10, 61]. Golub and Pereyra developed the original *Variable Projection* (VP) algorithm to solve general separable NLS problems [60]. The main idea behind VP is to explicitly eliminate the linear variables and solve the reduced NLS problem. This technique has been applied to a wide range of applications. Both theoretically [134] and empirically it has been shown that, compared to solving the full NLS problem, variable projection techniques exhibit faster or equal convergence rates (see [61] and references therein). An extensive survey of VP applications can be found in [61]. A theoretical analysis on the convergence properties of variable projection methods is due to Ruhe and Wedin [134, 135]. The high-dimensional state space of SLAM is one of its distinctive features in comparison with many other applications of VP. This is why retaining and exploiting sparsity is vital to the scalability of SLAM solvers.

In statistical terms, state-space models that exhibit "separable" structures are called *conditionally linear-Gaussian* state-space models [40] since estimating the linear variables, given the nonlinear ones and measurements corrupted by Gaussian noise, corresponds to a linear-Gaussian system. A common way of approaching conditionally linear-Gaussian state-space models is to use Rao-Blackwellised particle filters (RBPF), see e.g., [40, 42]. Let $\boldsymbol{\theta}$ denote the orientation of the robot throughout its trajectory, $\mathbf{p}$ be the stacked vector of robot (and landmark) positions, and finally $\mathbf{z}$ be the stacked vector of measurements—i.e., pairwise pose relations such as odometry and loop closures, as well as landmark observations. RBPF typically uses the standard sequential importance resampling (SIR) filter to represent $p(\boldsymbol{\theta}|\mathbf{z})$ using N weighted samples $\{\boldsymbol{\theta}^{[i]}\}_{i=1}^{N}$ drawn independently from a proposal distribution; i.e.,

$$p(\boldsymbol{\theta}|\mathbf{z}) \approx \sum_{i=1}^{N} w_i \cdot \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^{[i]}), \tag{3.1}$$

in which δ is the Dirac delta function. Then, for *each* sample, the *conditionally-optimal estimate* for $\mathbf{p}$ is recovered *analytically* by computing the mean of $p(\mathbf{p}|\boldsymbol{\theta}^{[i]},\mathbf{z})$ using the Kalman filter (i.e., recursive linear least squares).[1] It is worth noting that this approach naturally leads to the minimum mean-square error point estimate (MMSE)—i.e., mean of the joint posterior distribution; not its mode. Zikos and Petridis [159] in L-SLAM [160] apply this idea to exploit the separable structure of feature-based SLAM. In their RBPF, rather than partitioning the variables into landmarks and poses which was used in FastSLAM [113], they divide the variables into $\boldsymbol{\theta}$ and $\mathbf{p}$. L-SLAM has two major drawbacks. First and foremost, any sequential Monte Carlo method employed on a high-dimensional state space will eventually suffer from degeneracy and, subsequently, sample impoverishment [41]. Sample impoverishment has a negative impact on estimating $\boldsymbol{\theta}$ due to the loss of sample diversity; after some time $t \gg t_0$, eventually all of the particles will share the same estimate for $\{\theta_i\}_{i=1}^{t-t_0}$ for any t_0 (i.e., effectively only one sample is drawn from the corresponding region of state-space). Furthermore, by dividing variables into $\mathbf{p}$ and $\boldsymbol{\theta}$, the conditional independence property exploited in FastSLAM to gain computational efficiency will be lost. Hence, L-SLAM has to maintain a Kalman filter with a "large" dense covariance matrix.

In our previous work [156] linear variables of 2D feature-based problems with spherical noise are explicitly eliminated to obtain a smaller optimization problem over $\boldsymbol{\theta}$. This approach is similar to Golub and Pereyra's VP [60], but with numerical differentiation and Newton iterations. This method is computationally beneficial *only* in extremely noisy problems with dense graphs. LAGO [23, 24] uses the separable structure of 2D pose-graph SLAM to bootstrap GN. First a refined estimate for $\boldsymbol{\theta}$ is computed by only considering relative measurements of robot heading. Based on this initial estimate, LAGO then recovers the conditionally-optimal estimate for $\mathbf{p}$. The final outcome is used to bootstrap GN. From this perspective, our algorithm can be roughly interpreted as a constant use of LAGO's bootstrapping approach, without the initial phase of approximating $\boldsymbol{\theta}$ (which makes our algorithm robust to strong correlations between the noise components [24, 73]). Unlike our algorithm, LAGO is limited to 2D pose-graphs.

The equivalence between the minima of the original optimization problem and those of the reduced problem allows researchers to study various properties of SLAM by looking at the reduced problem. For instance, in [154, 155] we analyze the number of local minima in some small special cases based on this idea. Similarly, Carlone [20] used the reduced problem to analyze the convergence of GN in 2D pose-graphs with spherical noise covariance.

[1] It will become clear shortly that the latter stage is equivalent to (3.36) in our approach.

3.2 Problem Formulation

3.2.1 Graphical Representation of SLAM

Let $\mathcal{G} = (\mathcal{V},\mathcal{E})$ be the graphical representation of SLAM.[2] Vertices of $\mathcal{G}$ correspond to state variables (e.g,., robot's pose, landmark's positions) and its edges represent the corresponding relative pairwise measurements. It is easy to see that the SLAM problem will be an ill-posed problem if $\mathcal{G}$ is not weakly connected. Any SLAM problem with $c \geq 1$ connected components is essentially equivalent to c independent SLAM problem with respect to c independent reference frames. Obviously, for $c \geq 2$, the relative transformation between the frames will be unobservable. Moreover, in practice, the odometry spanning tree usually guarantees the connectivity of the graph. Therefore, without loss of generality we can assume $\mathcal{G}$ is weakly connected. The *reduced* incidence matrix of $\mathcal{G}$ after anchoring $\mathbf{x}_0$ to the origin (i.e., deleting the corresponding row from the original incidence matrix) is denoted by $\mathbf{A}$.[3] Let $\mathbf{A}_{\text{odo}}$ denote the reduced incidence matrix of the spanning tree of $\mathcal{G}$ consisting only of robot poses and odometry measurements. Note that the structure of $\mathbf{A}_{\text{odo}}$ is uniquely determined by the number of poses n_p. Finally, the *ℓ-expansion* of $\mathbf{A}$ is defined as $\mathbf{A}_\ell \triangleq \mathbf{A} \otimes \mathbf{I}_\ell$ for $\ell \in \mathbb{Z}_{\geq 2}$.

3.2.2 Measurement Model

The conventional state vector is usually defined as $[\mathbf{x}_1^\top \, \mathbf{x}_2^\top \ldots \mathbf{x}_n^\top]^\top$. As it will become clear shortly, it is more convenient to permute the standard state vector and define our state vector as $\mathbf{x} \triangleq [\mathbf{p}^\top \; \boldsymbol{\theta}^\top]^\top$. In 2D SLAM, $\mathbf{p} \in \mathbb{R}^{2n}$ is the vector of x and y coordinates of robot poses and landmark positions, and $\boldsymbol{\theta} \in [-\pi,\pi)^{n_p-1}$ is the vector of robot orientations. Each relative pairwise observation $\mathbf{z}_{ij}$ (from node i to node j) is corrupted by an independently drawn additive Gaussian noise $\boldsymbol{\epsilon}_{ij} \sim \mathcal{N}(\mathbf{0},\boldsymbol{\Sigma}_{ij})$,

$$\mathbf{z}_{ij} = \mathbf{h}_{ij}(\mathbf{x}_i,\mathbf{x}_j) + \boldsymbol{\epsilon}_{ij}. \tag{3.2}$$

Let $\mathcal{P}$ and $\mathcal{L}$ are the disjoint sets of indices of robot poses and landmarks, respectively. The measurement function $\mathbf{h}_{ij}(\cdot,\cdot)$ for any $(i,j) \in \mathcal{E}$ has the following form. If $j \in \mathcal{L}$,

$$\mathbf{h}_{ij}(\mathbf{x}_i,\mathbf{x}_j) = \mathbf{R}(\theta_i)^\top(\mathbf{p}_j - \mathbf{p}_i)\,, \tag{3.3}$$

[2]Here we treat $\mathcal{G}$ as a directed graph. Nevertheless, as we see in the next chapters, some important properties of SLAM are somewhat insensitive to the orientation of the edges.

[3]See Appendix A.4.

and if $j \in \mathcal{P}$,

$$\mathbf{h}_{ij}(\mathbf{x}_i,\mathbf{x}_j) = \begin{bmatrix} \mathbf{R}(\theta_i)^\top(\mathbf{p}_j - \mathbf{p}_i) \\ \\ \operatorname{wrap}(\theta_j - \theta_i) \end{bmatrix}, \tag{3.4}$$

where $\mathbf{R}(\theta_i) \in \mathrm{SO}(2)$[4] is the rotation matrix corresponding to θ_i, $\operatorname{wrap} : \mathbb{R} \to [-\pi,\pi)$ maps its argument to the equivalent angle in $[-\pi,\pi)$. Define,

$$\mathbf{R}_\theta \triangleq \operatorname{diag}\Big(\mathbf{R}(\theta_{k_1}),\ldots,\mathbf{R}(\theta_{k_m})\Big). \tag{3.5}$$

Here k_i is the index of robot pose making the ith observation. Let $\mathbf{z}_p$ and $\mathbf{z}_\theta$ denote the stacked vector of translational and rotational measurements, respectively. We permute the measurement vector accordingly to obtain $\mathbf{z} \triangleq [\mathbf{z}_p^\top \ \mathbf{z}_\theta^\top]^\top$. Similarly, the stacked vector of noise variables and its covariance matrix are denoted by $\boldsymbol{\epsilon} \triangleq [\boldsymbol{\epsilon}_p^\top \ \boldsymbol{\epsilon}_\theta^\top]^\top$ and $\boldsymbol{\Sigma}$, respectively. Thus the measurement model can be expressed as,[5]

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\boldsymbol{\Sigma}), \tag{3.6}$$

$$p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z};\mathbf{h}(\mathbf{x}),\boldsymbol{\Sigma}). \tag{3.7}$$

Remark 1. *Note that* (3.3) *and* (3.4) *admit both pose-graph and feature-based SLAM problems as special cases. In pose-graph SLAM $\mathcal{L} = \varnothing$, and in feature-based SLAM, pairwise pose measurements are limited to odometry measurements.*

Remark 2. *As the title suggests, in this chapter we focus on developing a new SLAM back-end to exploit the separable structure of SLAM. Consequently, we generally assume the data association is given. Solving the data association problem can be challenging and needs to be dealt with separately in the SLAM front-end [64, 98]. It is also common to make the back-end robust in order to deal with any remaining false positive associations; see, e.g., [1, 144]. Fortunately, the separable structure of SLAM is preserved under such formulations.*

[4] Special orthogonal group.

[5] To simplify our notation, we denote random variables (e.g., $\mathbf{z}$) and their realisation in the same way.

According to (3.3) and (3.4), the stacked measurement function of 2D SLAM is given by

$$\mathbf{h}(\mathbf{x}) = \mathbf{H}(\boldsymbol{\theta})\,\mathbf{x} \triangleq \begin{bmatrix} \mathbf{R}_\theta^\top \mathbf{A}_2^\top & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}^\top \end{bmatrix} \mathbf{x},$$

$$\boldsymbol{\Lambda} = \begin{cases} \mathbf{A} & \text{pose-graph,} \\ \mathbf{A}_{\text{odo}} & \text{feature-based,} \end{cases} \tag{3.8}$$

in which $\mathbf{A}_2 \triangleq \mathbf{A} \otimes \mathbf{I}_2$. We assume the correct regularization terms are applied to the measurements [23, 24]. Rather than distinguishing between feature-based and pose graph scenarios, for convenience we use the unified measurement function (3.8) throughout this chapter. Note that (3.8) can be rewritten as $\mathbf{h}(\mathbf{x}) = \mathbf{H}_1(\boldsymbol{\theta})\mathbf{p} + \mathbf{H}_2\boldsymbol{\theta}$ in which,

$$\mathbf{H}_1(\boldsymbol{\theta}) \triangleq \begin{bmatrix} \mathbf{R}_\theta^\top \mathbf{A}_2^\top \\ \mathbf{0} \end{bmatrix}, \; \mathbf{H}_2 \triangleq \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\Lambda}^\top \end{bmatrix}. \tag{3.9}$$

It is important to note that unlike $\mathbf{H}_2$, $\mathbf{H}_1(\boldsymbol{\theta})$ depends on $\boldsymbol{\theta}$. Nevertheless, with a slight abuse of notation, we drop the argument of $\mathbf{H}_1$ for the sake of simplicity of our notation.

Remark 3. *The standard relative pose-pose and pose-point measurement models in 3D SLAM are also affine with respect to robot's and landmarks' positions. The stacked vector of noise-free translational measurements in 3D SLAM can be written as* $\mathbf{R}_\theta^\top \mathbf{A}_3^\top \mathbf{p}$ *where* $\mathbf{p}$ *is the stacked vector of positions,* $\mathbf{A}_3 \triangleq \mathbf{A} \otimes \mathbf{I}_3$*, and* $\mathbf{R}_\theta$ *is a block-diagonal matrix similar to* (3.5) *but with* 3×3 *rotation matrices. Note that rotational measurements do not depend on* $\mathbf{p}$*. To simplify our notation and without loss of generality, we mainly focus on the 2D SLAM measurement model in the following sections. We will revisit the case of 3D SLAM in Remark 5 and explain how our algorithm can be generalized to 3D pose-graph and feature-based SLAM. Although in this chapter we do not consider a specific choice of sensors, the use of inertial sensors in 3D SLAM is quite common [109]. Therefore, it is worth noting that inertial measurements do not violate the separable structure of SLAM as such measurements are affine in* $\mathbf{p}$ *(see e.g., [109]).*

Remark 4. *This chapter investigates SLAM problems with the standard 2D and 3D relative pairwise pose-pose measurements for pose-graphs, and pose-point measurements for feature-based problems. Such measurements can be obtained from range-bearing sensors after a reparameterization of original measurements (e.g., after performing scan-matching). These models have become standard choices in the past* 10 *years (see, e.g., [64]). However, such reparameterizations often involve nonlinear transformation of the original measurements. Due*

to such transformations, small errors may appear in the covariance matrices. These errors are typically insignificant if the original covariance is "fairly small". In the context of our work, such transformations can be thought of as reparameterization performed to introduce separability. Note that directly using range-bearing measurements does not lead to a separable NLS.

3.2.3 Point Estimation Criterion

In this chapter, we mainly focus on $\hat{\mathbf{x}}_{\text{ML}}$ (2.8) as it is fairly rare to have an informative prior over $\mathbf{x}$ in real applications. Nevertheless, our approach can be straightforwardly generalized to the Bayesian formulation (2.6) as well. For convenience, we denote the optimal estimate for any variable like c with $c^\star$. Here $c^\star$ is either $\hat{c}_{\text{ML}}$ or $\hat{c}_{\text{MAP}}$. Similarly, we denote the (full) NLS cost function by $f(\mathbf{p},\boldsymbol{\theta})$, where f could be either (2.6) or (2.8).

3.3 Exploiting Separability in SLAM

Recall that the MAP and ML estimation problems in SLAM can be expressed as *nonlinear* least squares problems. By further inspection of (3.9) one can see that the nonlinearity is caused by the rotation matrices. Therefore, given the robot orientations $\boldsymbol{\theta}$, measurements are affine in robot and feature positions $\mathbf{p}$. Thus, given $\boldsymbol{\theta}$, the ML and MAP estimates for $\mathbf{p}$ can be computed by solving *linear* least squares problems. NLS problems whose residuals are affine with respect to a subset of variables are often called *separable* NLS [61]. This remarkable structure distinguishes SLAM from the general NLS. In the rest of this chapter, we show how this structure can be exploited to speed up the estimation process.

3.3.1 Variable Projection

Variable projection (VP) is an algorithm proposed by Golub and Pereyra to exploit this structure [60]. Golub and Pereyra [60] proved that, under some regularity conditions, the solution of the original separable NLS problem can be obtained using the following procedure (see [60, Theorem 2.1]). Below, we describe how their approach can be applied to SLAM.

Phase I.	Find $\mathbf{p}^\star(\boldsymbol{\theta})$, the conditionally-optimal estimate for $\mathbf{p}$ as a function of $\boldsymbol{\theta}$ by minimizing the original cost function in $\mathbf{p}$.
Phase II.	Replace $\mathbf{p}$ with $\mathbf{p}^\star(\boldsymbol{\theta})$ in the original problem and minimize the new objective function

in $\boldsymbol{\theta}$ to obtain $\boldsymbol{\theta}^\star$. After solving this problem using, e.g., Newton-based solvers, the optimal $\mathbf{p}^\star = \mathbf{p}^\star(\boldsymbol{\theta}^\star)$ can be recovered instantly by solving a single linear least squares problem.

Phase I – $\mathbf{p}^\star(\boldsymbol{\theta})$: We begin with the positive-definite square root of the noise precision matrix $\boldsymbol{\Sigma}^{-\frac{1}{2}} \succ \mathbf{0}$. To simplify our notation, we express the weighted ℓ^2 norm minimization in (2.8) as the following unweighted least squares,

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}} \|\tilde{\mathbf{z}} - \tilde{\mathbf{H}}_1\mathbf{p} - \tilde{\mathbf{H}}_2\boldsymbol{\theta}\|^2, \tag{3.10}$$

in which $\tilde{\mathbf{z}} \triangleq \boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{z}$, $\tilde{\mathbf{H}}_1 \triangleq \boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{H}_1$ and $\tilde{\mathbf{H}}_2 \triangleq \boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{H}_2$. The fact that the reduced incidence matrix is full rank in weakly connected graphs results in the following lemma.

Lemma 3.3.1. *For the measurement models defined in Section 3.2 (including 2D/3D feature-based or pose-graph), if the corresponding graph is weakly connected, $\tilde{\mathbf{H}}_1$ is full column rank regardless of $\boldsymbol{\theta}$.*

Proof. First note that $\boldsymbol{\Sigma}^{-\frac{1}{2}}$ is full rank. Then, using the rank-nullity theorem it is easy to verify that $\tilde{\mathbf{H}}_1 = \boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{H}_1$ is full column rank if and only if $\mathbf{H}_1$ is full column rank. According to (3.9), $\mathbf{H}_1$ is full column rank if and only if $\mathbf{R}_\theta^\top\mathbf{A}_2^\top$ and $\mathbf{R}_\theta^\top\mathbf{A}_3^\top$ are full column rank. Obviously, $\mathbf{R}_\theta$ is not singular ($\mathbf{R}_\theta^\top\mathbf{R}_\theta = \mathbf{I}$). The reduced incidence matrix $\mathbf{A}$ is full row rank if and only if the corresponding graph is weakly connected [27], which is the case in any well-defined SLAM problem (see the discussion in Section 3.2.1). Consequently, $\mathbf{H}_1$ is full column rank, regardless of the value of $\boldsymbol{\theta}$. □

As mentioned before, given $\boldsymbol{\theta}$, (3.10) is a linear least squares problem in $\mathbf{p}$. Lemma 3.3.1 assures us that, for any given $\boldsymbol{\theta}$, the optimal choice for $\mathbf{p}$ as a function of $\boldsymbol{\theta}$ is uniquely given by,

$$\mathbf{p}^\star(\boldsymbol{\theta}) \triangleq \arg\min_{\mathbf{p}} \|\tilde{\mathbf{z}} - \tilde{\mathbf{H}}_1\mathbf{p} - \tilde{\mathbf{H}}_2\boldsymbol{\theta}\|^2 \tag{3.11}$$

$$= \tilde{\mathbf{H}}_1^\dagger(\tilde{\mathbf{z}} - \tilde{\mathbf{H}}_2\boldsymbol{\theta}), \tag{3.12}$$

in which $\tilde{\mathbf{H}}_1^\dagger \triangleq (\tilde{\mathbf{H}}_1^\top\tilde{\mathbf{H}}_1)^{-1}\tilde{\mathbf{H}}_1^\top$ is the Moore-Penrose pseudoinverse of $\tilde{\mathbf{H}}_1$ (see, e.g., [10]).

Phase II – Reduced NLS: By substituting $\mathbf{p}$ in the original objective function (3.10) with $\mathbf{p}^\star(\boldsymbol{\theta})$ in (3.12) and solving the resulting optimization problem we obtain $\boldsymbol{\theta}^\star \triangleq \arg\min_{\boldsymbol{\theta}} g(\boldsymbol{\theta})$ where

$$g(\boldsymbol{\theta}) \triangleq \|(\mathbf{I} - \tilde{\mathbf{H}}_1\tilde{\mathbf{H}}_1^\dagger)(\tilde{\mathbf{z}} - \tilde{\mathbf{H}}_2\boldsymbol{\theta})\|^2. \tag{3.13}$$

Note that $g(\cdot)$ is a function of only robot headings $\boldsymbol{\theta}$, while the original optimization problem (3.10)

was over both $\mathbf{p}$ and $\boldsymbol{\theta}$. Hence, we have reduced the parameter space from $\mathbb{R}^{2n} \times [-\pi,\pi)^{n_p-1}$ to $[-\pi,\pi)^{n_p-1}$. $\mathbf{P}_\theta \triangleq \tilde{\mathbf{H}}_1\tilde{\mathbf{H}}_1^\dagger$ is the orthogonal projection onto $\mathrm{range}(\tilde{\mathbf{H}}_1)$, while $\mathbf{P}_\theta^\perp \triangleq \mathbf{I} - \tilde{\mathbf{H}}_1\tilde{\mathbf{H}}_1^\dagger$ is its orthogonal complement. Let us define $\mathbf{r}_{\mathrm{vp}} \triangleq \mathbf{P}_\theta^\perp(\tilde{\mathbf{z}} - \tilde{\mathbf{H}}_2\boldsymbol{\theta})$. In order to solve (3.13) using Newton-based NLS solvers we need to compute the Jacobian matrix of $\mathbf{r}_{\mathrm{vp}}$, i.e., $\mathbf{J}_{\mathrm{vp}} \triangleq \frac{\partial}{\partial\boldsymbol{\theta}}\mathbf{r}_{\mathrm{vp}}$. Computing $\mathbf{J}_{\mathrm{vp}}$ requires differentiating pseudoinverses ($\tilde{\mathbf{H}}_1^\dagger$), and therefore is more complex than computing the Jacobian matrix of the original full problem, i.e., $\mathbf{J} \triangleq \frac{\partial}{\partial\mathbf{x}}\mathbf{r}$ in which $\mathbf{r} \triangleq \mathbf{z} - \mathbf{h}(\mathbf{x})$. Although it is possible to approximate $\mathbf{J}_{\mathrm{vp}}$ using finite differences (see [61] and the references therein), here we employ the exact analytical expression for $\mathbf{J}_{\mathrm{vp}}$ which is based on the seminal work of Golub and Pereyra [60].

Computing $\mathbf{J}_{\mathrm{vp}}$: First note that (3.13) has a more general form compared to the case which was originally considered by Golub and Pereyra [60], in that we have an additional term, linear with respect to $\boldsymbol{\theta}$ in our residual, i.e., $-\tilde{\mathbf{H}}_2\,\boldsymbol{\theta}$.

Theorem 3.3.1. *The jth column of $\mathbf{J}_{\mathrm{vp}}$ is given by,*

$$[\mathbf{J}_{\mathrm{vp}}]_{\cdot,j} = -\left((\mathbf{P}_{\boldsymbol{\theta}}^\perp\frac{\partial\tilde{\mathbf{H}}_1}{\partial\theta_j}\tilde{\mathbf{H}}_1^\dagger) + (\mathbf{P}_{\boldsymbol{\theta}}^\perp\frac{\partial\tilde{\mathbf{H}}_1}{\partial\theta_j}\tilde{\mathbf{H}}_1^\dagger)^\top\right)(\tilde{\mathbf{z}} - \tilde{\mathbf{H}}_2\,\boldsymbol{\theta}) - \mathbf{P}_{\boldsymbol{\theta}}^\perp[\tilde{\mathbf{H}}_2]_{\cdot,j}. \tag{3.14}$$

After finding $\boldsymbol{\theta}^\star \triangleq \arg\min g(\boldsymbol{\theta})$ using an iterative NLS solver such as GN, we can recover the optimal $\mathbf{p}^\star$ according to (3.12); i.e., by solving $(\tilde{\mathbf{H}}_1^\top\tilde{\mathbf{H}}_1)\mathbf{p}^\star = \tilde{\mathbf{H}}_1^\top(\tilde{\mathbf{z}} - \tilde{\mathbf{H}}_2\boldsymbol{\theta}^\star)$. In general, VP iterations are computationally more expensive than GN iterations on the full problem. Furthermore, the reduced Jacobian (3.14) unlike the original Jacobian is generally dense. Therefore, directly applying VP to SLAM is impractical and will not lead to an efficient solver.

3.3.2 Kaufman's Algorithm

Kaufman [86] proposed to approximate the jth column of $\mathbf{J}_{\mathrm{vp}}$ according to

$$[\mathbf{J}_{\mathrm{vp}}^{\mathrm{K}}]_{\cdot,j} \triangleq -\left(\mathbf{P}_{\boldsymbol{\theta}}^\perp\frac{\partial\tilde{\mathbf{H}}_1}{\partial\theta_j}\tilde{\mathbf{H}}_1^\dagger\right)(\tilde{\mathbf{z}} - \tilde{\mathbf{H}}_2\,\boldsymbol{\theta}) - \mathbf{P}_{\boldsymbol{\theta}}^\perp[\tilde{\mathbf{H}}_2]_{\cdot,j}. \tag{3.15}$$

In small-residual problems, the impact of the term neglected in $\mathbf{J}_{\mathrm{vp}}^{\mathrm{K}}$ on the convergence rate of the algorithm will be negligible as shown in [86]. The consequences of this approximation is rigorously

Algorithm 1 SLAM solver based on [86]

repeat
  Compute the full QR factorization of $\tilde{\mathbf{H}}_1$ (3.16)
  Recover $\mathbf{p}^\star$ by solving $\mathbf{R}_1\,\mathbf{p}^\star(\boldsymbol{\theta}_{(i)}) = \mathbf{Q}_1^\top(\tilde{\mathbf{z}} - \tilde{\mathbf{H}}_2\boldsymbol{\theta}_{(i)})$
  Compute the modified residual and Jacobian (3.19)
  Construct the normal equations for the reduced problem
  Solve the normal equations to obtain $\delta\boldsymbol{\theta}_{(i)}$
  $\boldsymbol{\theta}_{(i+1)} \leftarrow \boldsymbol{\theta}_{(i)} + \delta\boldsymbol{\theta}_{(i)}$
until convergence
$\mathbf{p}^\star \leftarrow \mathbf{p}^\star(\boldsymbol{\theta}^\star)$ according to (3.12)

investigated in [134] and has been verified numerically in various domains [60]. In SLAM, the residuals will be small if the measurements are sufficiently consistent with each other, or equivalently, when the realized noise is sufficiently small. It is worth noting that even GN, as an approximation of Newton's method, relies on an explicit small-residual assumption [10]. Therefore, it is sensible to conclude that large residuals will have a negative impact on the convergence rate of both VP and GN. Kaufman's simplification reduces the time per iteration of VP up to 25% [134]. Because of this reduced complexity, Kaufman's method has become the preferred algorithm for solving separable NLS problems [61]. As we will see shortly, in *sparse* separable NLS problems (such as SLAM), Kaufman's modification, in a slightly different form, plays an even more crucial role by enabling us to maintain the sparsity.

Algorithm 1 summarizes a SLAM solver based on an efficient implementation of VP using Kaufman's modification. In the rest of this section we discuss some important details about implementing an efficient version of VP with Kaufman's modification using the QR decomposition of $\tilde{\mathbf{H}}_1$. First we need to compute the *full* QR factorization of $\tilde{\mathbf{H}}_1$,

$$\tilde{\mathbf{H}}_1 = \mathbf{Q}\mathbf{R} = \begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix}\begin{bmatrix}\mathbf{R}_1 \\ \mathbf{0}\end{bmatrix}. \tag{3.16}$$

The sparse structure of $\tilde{\mathbf{H}}_1$ can be exploited in this stage by using optimized software packages such as SuiteSparseQR [34]. Define $d_z \triangleq \dim(\mathbf{z})$ and $d_p \triangleq \dim(\mathbf{p})$. Then $\mathbf{Q}_1 \in \mathbb{R}^{d_z \times d_p}$ and $\mathbf{Q}_2 \in \mathbb{R}^{d_z \times (d_z - d_p)}$ are orthogonal matrices, and $\mathbf{R}_1 \in \mathbb{R}^{d_p \times d_p}$ is an upper triangular matrix. The columns of $\mathbf{Q}_1$ and $\mathbf{Q}_2$ form orthonormal bases for $\mathrm{range}(\tilde{\mathbf{H}}_1)$ and $\mathrm{null}(\tilde{\mathbf{H}}_1^\top)$, respectively. Consequently $\mathbf{P}_{\boldsymbol{\theta}} = \mathbf{Q}_1\mathbf{Q}_1^\top$ and $\mathbf{P}_{\boldsymbol{\theta}}^\perp = \mathbf{Q}_2\mathbf{Q}_2^\top$. The residual vector of VP functional can be simplified using

(3.16),

$$\mathbf{r}_{\text{vp}} = \mathbf{P}_{\boldsymbol{\theta}}^{\perp}(\tilde{\mathbf{z}} - \tilde{\mathbf{H}}_2\boldsymbol{\theta}) = \mathbf{Q}_2\mathbf{Q}_2^{\top}(\tilde{\mathbf{z}} - \tilde{\mathbf{H}}_2\boldsymbol{\theta}). \tag{3.17}$$

To evaluate the reduced NLS cost function $g(\cdot)$ we can use the QR decomposition of $\tilde{\mathbf{H}}_1$ in (3.16) and the fact that the Euclidean norm is invariant under orthogonal transformations. Therefore we have $g(\boldsymbol{\theta}) = \|\mathbf{Q}_2^{\top}(\tilde{\mathbf{z}} - \tilde{\mathbf{H}}_2\boldsymbol{\theta})\|^2$. According to (3.15), the jth column of $\mathbf{J}_{\text{vp}}^{\text{K}}$ is given by

$$[\mathbf{J}_{\text{vp}}^{\text{K}}]_{\cdot,j} = -\mathbf{Q}_2\mathbf{Q}_2^{\top}\left(\frac{\partial\tilde{\mathbf{H}}_1}{\partial\theta_j}\mathbf{p}^{\star}(\boldsymbol{\theta}) + [\tilde{\mathbf{H}}_2]_{\cdot,j}\right). \tag{3.18}$$

Note that for a given $\boldsymbol{\theta}$, $\mathbf{p}^{\star}(\boldsymbol{\theta})$ can be easily computed by solving $\mathbf{R}_1\,\mathbf{p}^{\star}(\boldsymbol{\theta}) = \mathbf{Q}_1^{\top}(\tilde{\mathbf{z}} - \tilde{\mathbf{H}}_2\boldsymbol{\theta})$ by back-substitution. It can be easily verified that multiplying both the residual vector and Jacobian matrix from the left by an orthogonal matrix does not change the GN direction. Hence, using this invariance property we can eliminate $\mathbf{Q}_2$ by multiplying (3.17) and (3.18) by $\mathbf{Q}_2^{\top}$ from the left.

$$\begin{aligned}\mathbf{Q}_2^{\top}\mathbf{r}_{\text{vp}} &= \mathbf{Q}_2^{\top}(\tilde{\mathbf{z}} - \tilde{\mathbf{H}}_2\boldsymbol{\theta})\\ [\mathbf{Q}_2^{\top}\mathbf{J}_{\text{vp}}^{\text{K}}]_{\cdot,j} &= -\mathbf{Q}_2^{\top}\left(\frac{\partial\tilde{\mathbf{H}}_1}{\partial\theta_j}\mathbf{p}^{\star}(\boldsymbol{\theta}) + [\tilde{\mathbf{H}}_2]_{\cdot,j}\right).\end{aligned} \tag{3.19}$$

Note that $\mathbf{Q}_2$ could not be "eliminated" without using Kaufman's simplification.

3.4 Sparse Variable Projection

In the previous section we explained how Algorithm 1 exploits the separable structure of SLAM. At first glance, applying this algorithm to SLAM may appear to be computationally advantageous as it reduces the size of the normal equations from $(2n + n_p)$ to only n_p. However, the normal equations of the reduced problem in SLAM is dense in general. Hence VP, as implemented in Algorithm 1, does not lead to a scalable algorithm as we would need at least $O(n_p^2)$ space and $O(n_p^3)$ time per iteration to solve the resulting *dense* linear system. Therefore it is sensible to ask whether we can exploit separability without giving up the intrinsic sparse structure of SLAM.

Barham and Drane [7] proposed an intuitive algorithm to solve separable NLS problems. Unlike the variable projection algorithm, in their approach the linear variables are not entirely eliminated from the optimization problem. They use the Schur complement in each iteration to solve the normal equations of the original full NLS only for the nonlinear variables. Then, instead of back-substituting the resulting estimate into the normal equations, they exploit the separable structure of the problem

by computing the conditionally-optimal estimate for the linear variables. This process is repeated until convergence. At first glance this intuitive algorithm may seem like a simple heuristic. However, it has been shown that this procedure leads to the same iterations as the Kaufman's algorithm [125].

It is well known that although the original system of normal equations is sparse, the Schur complement in general—and particularly in our case—is dense. However, the original algorithm can be simply amended to maintain sparsity. Instead of computing the Schur complement, we can simply solve the normal equations for both linear and nonlinear variables, and then replace the resulting estimate for the linear ones with the conditionally-optimal values (3.12). By doing so, the sparse structure of the problem is preserved in the process of exploiting separability. In what follows, we explain how this idea can be applied to SLAM.

Retaining Sparsity

Consider the normal equations of the original problem,

$$\tilde{\mathbf{J}}^\top \tilde{\mathbf{J}}\, \delta\mathbf{x}_{(i)} = -\tilde{\mathbf{J}}^\top \tilde{\mathbf{r}}. \tag{3.20}$$

Here $\delta\mathbf{x}_{(i)} \triangleq [\delta\mathbf{p}_{(i)}^\top \;\; \delta\boldsymbol{\theta}_{(i)}^\top]^\top$ denotes the ith GN direction, $\tilde{\mathbf{r}}$ is the normalized residual vector $\tilde{\mathbf{r}} \triangleq \boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{r}$ and $\tilde{\mathbf{J}} \triangleq \frac{\partial}{\partial \mathbf{x}}\tilde{\mathbf{r}}$, both evaluated at $\mathbf{x}_{(i)}$. Let $\mathbb{I} \triangleq \tilde{\mathbf{J}}^\top\tilde{\mathbf{J}}$ be the approximated Hessian. Note that $\tilde{\mathbf{J}}$ can be divided into two blocks, $\tilde{\mathbf{J}} = [\tilde{\mathbf{J}}_p \;\; \tilde{\mathbf{J}}_\theta]$ in which $\tilde{\mathbf{J}}_p \triangleq \frac{\partial}{\partial \mathbf{p}}\tilde{\mathbf{r}}$ and $\tilde{\mathbf{J}}_\theta \triangleq \frac{\partial}{\partial \boldsymbol{\theta}}\tilde{\mathbf{r}}$. Hence, (3.20) can be expressed as,

$$\begin{bmatrix} \mathbb{I}_p & \mathbb{I}_{p,\theta} \\ \mathbb{I}_{p,\theta}^\top & \mathbb{I}_\theta \end{bmatrix} \begin{bmatrix} \delta\mathbf{p}_{(i)} \\ \delta\boldsymbol{\theta}_{(i)} \end{bmatrix} = \begin{bmatrix} -\tilde{\mathbf{J}}_p^\top\tilde{\mathbf{r}} \\ -\tilde{\mathbf{J}}_\theta^\top\tilde{\mathbf{r}} \end{bmatrix}. \tag{3.21}$$

Barham and Drane [7] proposed to eliminate $\delta\mathbf{p}_{(i)}$ from (3.21) using the Schur complement of $\mathbb{I}$ with respect to $\mathbb{I}_p$, often written as $\mathbb{I}/\mathbb{I}_p$:

$$\mathbb{I}/\mathbb{I}_p \triangleq \mathbb{I}_\theta - \mathbb{I}_{p,\theta}^\top \mathbb{I}_p^{-1} \mathbb{I}_{p,\theta}. \tag{3.22}$$

Define

$$\mathbf{d}_\theta \triangleq -\tilde{\mathbf{J}}_\theta^\top\tilde{\mathbf{r}} + \mathbb{I}_{p,\theta}^\top \mathbb{I}_p^{-1} \tilde{\mathbf{J}}_p^\top\tilde{\mathbf{r}}. \tag{3.23}$$

After eliminating $\delta\mathbf{p}_{(i)}$, (3.21) reduces into,

$$(\mathbb{I}/\mathbb{I}_p)\,\delta\boldsymbol{\theta}_{(i)} = \mathbf{d}_\theta. \tag{3.24}$$

Solving (3.24) results in $\delta\boldsymbol{\theta}_{(i)}$. Back-substituting this solution into (3.21) and recovering $\delta\mathbf{p}_{(i)}$ leads to the standard GN direction for the original cost function. Instead, Barham and Drane [7] proposed to pair $\boldsymbol{\theta}_{(i+1)} = \boldsymbol{\theta}_{(i)} + \delta\boldsymbol{\theta}_{(i)}$ with the conditionally-optimal estimate $\mathbf{p}_{(i+1)}$ using (3.12) [7]. The conditionally-optimal estimate for $\mathbf{p}_{(i+1)}$ is $\mathbf{p}^\star(\boldsymbol{\theta}_{(i+1)})$. It can be computed by solving the following *sparse* linear system,

$$(\tilde{\mathbf{H}}_1^\top \tilde{\mathbf{H}}_1)\,\mathbf{p}_{(i+1)} = \tilde{\mathbf{H}}_1^\top \Big(\tilde{\mathbf{z}} - \tilde{\mathbf{H}}_2 \boldsymbol{\theta}_{(i+1)}\Big). \tag{3.25}$$

Note that here $\tilde{\mathbf{H}}_1$ is evaluated at $\boldsymbol{\theta}_{(i+1)}$. Repeating this procedure until convergence leads to a sequence of steps along which the cost function (2.8) has zero gradient with respect to $\mathbf{p}$. This algorithm is summarized in Algorithm 2.

As mentioned earlier, the Schur complement in the *reduced* linear system (3.24) is generally dense. Note that in each iteration of Algorithm 2, the solution of the reduced system (3.24) is identical to the $\delta\boldsymbol{\theta}_{(i)}$ obtained by solving the full system (3.21). Unlike the Schur complement, the full system of normal equations is sparse. Thus instead of eliminating linear variables using the Schur complement, the same outcome can be achieved by solving the sparse full system (3.21), discarding the obtained $\delta\mathbf{p}_{(i)}$ and instead computing the conditionally-optimal $\mathbf{p}_{(i+1)}$ according to (3.25). This algorithm is summarized in Algorithm 3.

It is of utmost importance to note that our proposed algorithm produces (mathematically) identical steps to those of Algorithm 1 and Algorithm 2 (see [125] for the equivalence between Kaufman's method and the algorithm proposed by Barham and Drane [7]). However, unlike those algorithms, Algorithm 3 only requires solving two *sparse* linear systems in each iteration which leads to a crucial computational benefit:

1. The first sparse linear system is the normal equations of the original full NLS problem (3.21). Solving this system results in $\delta\boldsymbol{\theta}_{(i)}$ and $\boldsymbol{\theta}_{(i+1)}$ consequently.

2. In the second sparse linear system, we recover the conditionally-optimal estimate $\mathbf{p}_{(i+1)}$ by solving (3.25). This is where we exploit the separable structure of SLAM.

Our algorithm combines the advantages of both Algorithm 1 and Algorithm 2. On the one hand, the equivalence between Algorithm 3 and Algorithm 1 (through their equivalence to Algorithm 2

[125]) provides a rigorous justification for our approach and connects us to the rich literature on variable projection and the performance of Kaufman's algorithm. On the other hand, the equivalence between Algorithm 3 and Algorithm 2, besides providing an intuitive interpretation for Kaufman's approximation, enables us to preserve the sparse structure of the problem. Algorithm 3 can be easily implemented by a simple modification of existing state-of-the-art back-ends: we only need an additional step to solve (3.25) using e.g., a sparse Cholesky solver. Table 3.1 provides a summarized comparison between these algorithms, sorted by their cost per iteration in descending order. An efficient implementation of our algorithm is discussed in Section 3.7. Figure 3.1 illustrates how our proposed algorithm works.

Remark 5. *As mentioned in Remark 3, our approach can be easily extended to 3D pose-graph and feature-based SLAM with the standard relative pose-pose and pose-point measurement models. Without loss of generality, let us assume the translational and rotational noise components are uncorrelated. With a little abuse of notation, the cost function in these problems can be expressed as,*

$$f(\mathbf{p},\boldsymbol{\theta}) = f_p(\mathbf{p},\boldsymbol{\theta}) + f_\theta(\boldsymbol{\theta}), \tag{3.26}$$

in which $f_p(\mathbf{p},\boldsymbol{\theta}) \triangleq \|\mathbf{z}_p - \mathbf{R}_\theta^\top \mathbf{A}_3^\top \mathbf{p}\|^2_{\boldsymbol{\Sigma}_p^{-1}}$ *and,*

$$f_\theta(\boldsymbol{\theta}) \triangleq \sum_{(i,j)\in\mathcal{E}_p} \|\mathrm{Log}(\mathbf{R}_{z_{ij}}^\top \mathbf{R}_i^\top \mathbf{R}_j)\|^2_{\boldsymbol{\Sigma}_{\theta_{ij}}^{-1}} \tag{3.27}$$

where $\mathrm{Log} : \mathrm{SO}(3) \to \mathbb{R}^3$ *maps a* 3×3 *rotation matrix to the corresponding rotation vector (i.e., axis-angle representation),* $\mathcal{E}_p \subseteq \mathcal{E}$ *is the subset of edges that correspond to pose-pose measurements,* $\mathbf{R}_{z_{ij}}$ *is the observed rotational measurement (corrupted by noise),* $\boldsymbol{\Sigma}_{\theta_{ij}}$ *is the corresponding covariance matrix, and* $\mathbf{R}_i \in \mathrm{SO}(3)$ *is the rotation matrix of the* i*th robot pose. The state-of-the-art sparse back-ends use a minimal parametrization (e.g., rotation vector) for* $\boldsymbol{\theta}$ *in each iteration of the standard iterative schemes such as GN. Let* $\boldsymbol{\theta}_{(k+1)}$ *be the estimate obtained at the* $(k+1)$*th iteration of GN using the update rule* $\boldsymbol{\theta}_{(k+1)} = \boldsymbol{\theta}_{(k)} \boxplus \delta\boldsymbol{\theta}_{(k)}$*. Here* $\boxplus$ *generalizes* $+$ *and can be defined based on, for example, the exponential map for* $\mathrm{SO}(3)$ *[71]. Now given* $\boldsymbol{\theta}_{(k+1)}$*, the conditionally-optimal estimate for* $\mathbf{p}$ *that minimizes* (3.26) *is the solution of,*

$$\tilde{\mathbf{H}}_1^{\circ\top} \tilde{\mathbf{H}}_1^\circ \mathbf{p}_{(k+1)} = \tilde{\mathbf{H}}_1^{\circ\top} \mathbf{z}_p \tag{3.28}$$

where $\tilde{\mathbf{H}}_1^\circ \triangleq \boldsymbol{\Sigma}_p^{-\frac{1}{2}} \mathbf{R}_\theta^\top \mathbf{A}_3^\top$ *in which* $\mathbf{R}_\theta$ *is evaluated at* $\boldsymbol{\theta}_{(k+1)}$*. The proof of Lemma 3.3.1 can be easily extended to show that* $\tilde{\mathbf{H}}_1^\circ$ *is always full rank if the underlying graph is weakly connected.*

Algorithm 2 SLAM solver based on [7]

repeat
  Construct the normal equations (3.21)
  Eliminate $\delta\mathbf{p}_{(i)}$ using the Schur complement (3.24)
  Solve (3.24) to obtain $\delta\boldsymbol{\theta}_{(i)}$
  $\boldsymbol{\theta}_{(i+1)} \leftarrow \boldsymbol{\theta}_{(i)} + \delta\boldsymbol{\theta}_{(i)}$
  $\mathbf{p}_{(i+1)} \leftarrow \mathbf{p}^\star(\boldsymbol{\theta}_{(i+1)})$ according to (3.25)
until convergence

Algorithm 3 Sparse Variable Projection

repeat
  Construct the normal equations (3.21)
  Use the sparse Cholesky solver to solve (3.21)
  $\boldsymbol{\theta}_{(i+1)} \leftarrow \boldsymbol{\theta}_{(i)} + \delta\boldsymbol{\theta}_{(i)}$
  $\mathbf{p}_{(i+1)} \leftarrow \mathbf{p}^\star(\boldsymbol{\theta}_{(i+1)})$ according to (3.25) using the sparse Cholesky solver
until convergence

Table 3.1: A summary of related algorithms sorted by their cost per iteration in descending order.

Algorithm	Separability	Sparsity
Algorithm 1 [86]	✓	✗
Algorithm 2 [7]	✓	✗
Algorithm 3	✓	✓
Full Least Squares	✗	✓

3.5 Projection Gain

Algorithm 3 and the conventional approach can be seen as the two ends of a spectrum of solvers. On one end, conventional methods provide cheap sparse iterations without exploiting the separable structure of the problem. On the other end of this spectrum, Algorithm 3 performs an extra projection step at the end of every iteration, which increases the effectiveness of each iteration at the cost of solving an extra (but smaller) sparse linear system (3.25) per iteration. In the middle of this spectrum we have solvers that benefit from both iterations based on a cost-benefit analysis. Such solvers would benefit simultaneously from the low cost of f-iterations and the extra effectiveness brought by performing a projection step. Our empirical observations indicate that performing projection is crucial in the first few iterations where it also exhibits a bootstrapping effect [73]. After only few iterations and upon getting sufficiently close to a local minimizer, the effectiveness of f-iterations become similar to that of VP iterations. This is where allocating resources to the projection step is not

justified anymore, and one can safely switch back to cheap f-iterations.

To design a switching scheme, first it is necessary to quantify the concept of projection gain. At the ith iteration we have

$$f_i^\star \triangleq f(\mathbf{p}^\star(\boldsymbol{\theta}_{(i)}),\boldsymbol{\theta}_{(i)}) = \min_{\mathbf{p}} f(\mathbf{p},\boldsymbol{\theta}_{(i)}) \leq f(\mathbf{p}_{(i)},\boldsymbol{\theta}_{(i)}) \triangleq f_i^\circ. \tag{3.29}$$

Here $\mathbf{p}_{(i)}$ and $\boldsymbol{\theta}_{(i)}$ specify the solution obtained after performing a GN iteration at iteration i on the full cost function (see the black point connected to e.g., $x_0^{\mathtt{VP}}$ via the red dashed vector in Figure 3.1) and f_i° is the value of (the original) cost function at this point. Similarly, $\mathbf{p}^\star(\boldsymbol{\theta}_{(i)})$ and $\boldsymbol{\theta}_{(i)}$ specify the solution obtained after performing the projection step (see $x_1^{\mathtt{VP}}$ in Figure 3.1) whose cost is denoted by $f_i^\star$. To quantify the projection gain, we define the *relative gain* as

$$\gamma_i \triangleq (f_i^\circ - f_i^\star)/f_i^\circ. \tag{3.30}$$

By definition we have $0 \leq \gamma_i \leq 1$. Note that computing the projection gain only requires an extra evaluation of the cost function f before performing the projection step. At $i = 0$, γ_i is usually close to one (which indicates a high projection gain) unless the initial guess is already close to a (local) solution. Generally speaking, γ_i exhibits a decreasing (but not necessarily monotonic) trend. Upon convergence to a solution, the gain will diminish and therefore $\gamma_i = 0$ (see Figure 3.2). Therefore, a sensible switching heuristic is to perform projections only while γ_i is larger than a threshold, γ_T.

3.6 Maximum A Posteriori Estimation

First note that (2.6) is also a separable NLS problem. Therefore, Algorithm 3 can be easily modified to find the MAP estimate assuming a Gaussian prior over $\mathbf{x}$ is available. Nevertheless, here we address the Bayesian formulation from a slightly different perspective that gives us new insights into the structure of SLAM. The posterior density $p(\mathbf{p},\boldsymbol{\theta}|\mathbf{z})$ can be factored according to

$$p(\mathbf{p},\boldsymbol{\theta}|\mathbf{z}) = p(\boldsymbol{\theta}|\mathbf{z})\, p(\mathbf{p}|\boldsymbol{\theta},\mathbf{z}). \tag{3.31}$$

Using the Bayes rule as shown in Appendix B.1 we have $p(\mathbf{p}|\boldsymbol{\theta},\mathbf{z}) \propto p(\mathbf{z}|\mathbf{x})\, p(\mathbf{p}|\boldsymbol{\theta})$. From the fact that the measurement function is affine in $\mathbf{p}$ it readily follows that a Gaussian prior over $\mathbf{x}$ results in a Gaussian $p(\mathbf{p}|\boldsymbol{\theta},\mathbf{z}) = \mathcal{N}(\mathbf{p};\boldsymbol{\mu}_{\boldsymbol{\theta}}^\circ,\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^\circ)$. To simplify our notation and without losing any generality let us

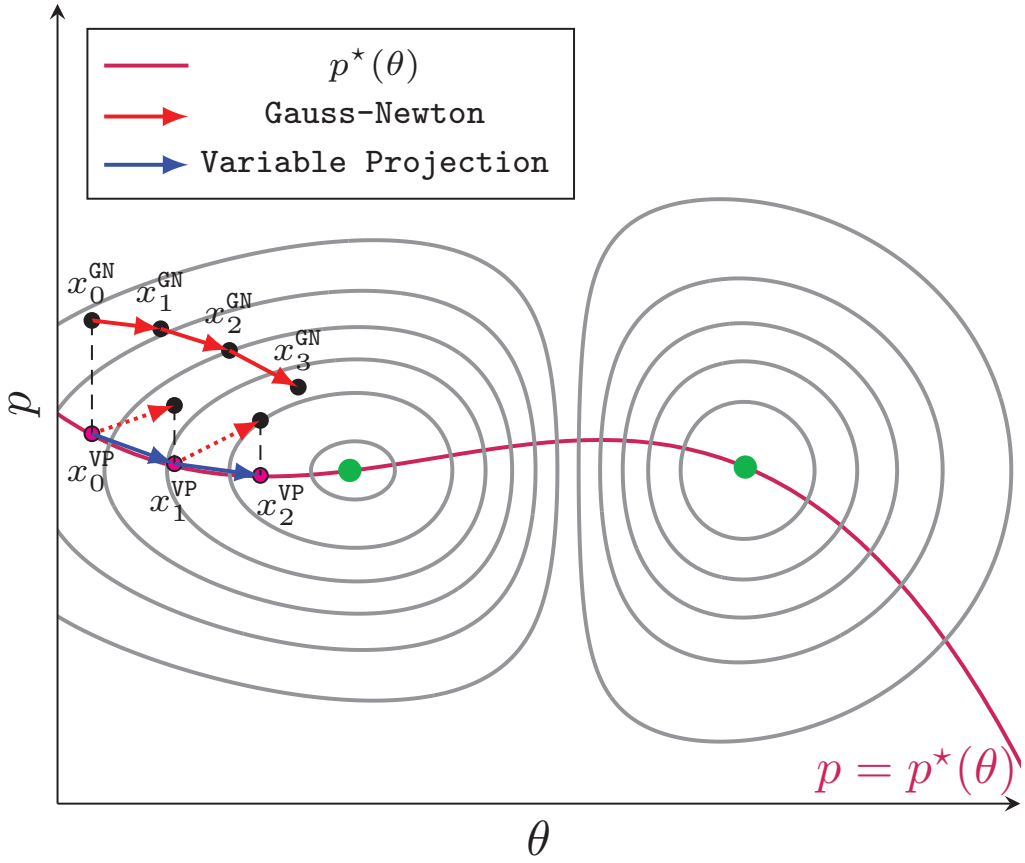


Figure 3.1: This figure illustrates the contour lines of a simplified version of SLAM cost function. The green points show the local minima. The magenta curve is $p^\star(\theta)$; a function that maps any given θ to the corresponding conditionally optimal estimate for p. The blue vector shows the ith step of Algorithm 1—which is identical to the ith step of Algorithm 2 and Algorithm 3. The red vector is the GN step obtained by starting from x_i^{GN}. The dashed red vectors are the intermediate GN steps obtained by starting from x_i^{VP}. Algorithm 3 corrects this intermediate step by projecting the obtained solution back on $p^\star(\theta)$ (dashed line). Note that Algorithm 1 computes the blue vector directly by performing a GN iteration on the *reduced* problem (3.13) with quadratic space and cubic time complexity. However our indirect approach in Algorithm 3 enables us to retain the sparse structure of the original problem.

assume $p(\mathbf{p}|\boldsymbol{\theta}) = p(\mathbf{p})$. As mentioned earlier, this prior is assumed to be Gaussian with mean $\boldsymbol{\mu}_\mathbf{p}$ and covariance $\boldsymbol{\Sigma}_\mathbf{p}$. Then we have,

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}^{\circ} = \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{\circ}\Big(\mathbf{H}_1^\top \boldsymbol{\Sigma}^{-1}(\mathbf{z} - \mathbf{H}_2\boldsymbol{\theta}) + \boldsymbol{\Sigma}_\mathbf{p}^{-1}\boldsymbol{\mu}_\mathbf{p}\Big) \tag{3.32}$$

$$= \boldsymbol{\mu}_\mathbf{p} + \boldsymbol{\Sigma}_\mathbf{p}\mathbf{H}_1^\top \mathbf{S}^{-1}\Big(\mathbf{z} - \mathbf{H}_2\boldsymbol{\theta} - \mathbf{H}_1\boldsymbol{\mu}_\mathbf{p}\Big) \tag{3.33}$$

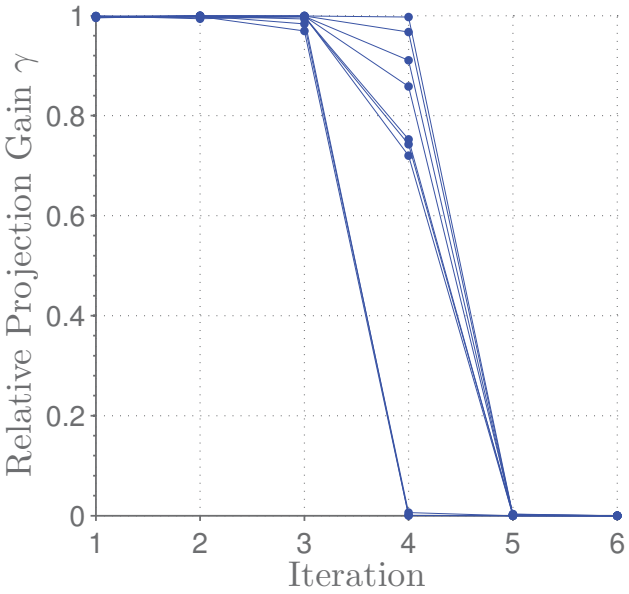


Figure 3.2: Evolution of the projection gain $\gamma_i \triangleq (f_i^\circ - f_i^\star)/f_i^\circ$ in 10 simulated datasets. The projection gain is maximum when γ is close to one. At the 6th iteration $\gamma = 0$ upon convergence to the optimal estimate.

in which we have used the matrix inversion lemma, $\mathbf{S} \triangleq \mathbf{H}_1 \boldsymbol{\Sigma}_\mathbf{p} \mathbf{H}_1^\top + \boldsymbol{\Sigma}$ is the so-called innovation covariance, and

$$\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^\circ = \left(\mathbf{H}_1^\top \boldsymbol{\Sigma}^{-1} \mathbf{H}_1 + \boldsymbol{\Sigma}_\mathbf{p}^{-1} \right)^{-1}. \tag{3.34}$$

Although in general the original posterior distribution $p(\mathbf{p},\boldsymbol{\theta}|\mathbf{z})$ may be far from being Gaussian (e.g., multi-modal, skewed, etc), recovering the optimal $\mathbf{p}$ given $\boldsymbol{\theta}$ reduces to a simple linear-Gaussian estimation problem. Such problems are often called *conditionally* linear-Gaussian. The MAP estimate—by definition—is the maximizer of the posterior distribution (3.31),

$$\mathbf{x}^\star = \arg\max_{\mathbf{p},\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathbf{z})\, p(\mathbf{p}|\boldsymbol{\theta},\mathbf{z}). \tag{3.35}$$

Since $\mathbf{p}$ only appears in the second term, maximizing the above product implies maximizing $p(\mathbf{p}|\boldsymbol{\theta},\mathbf{z})$ with respect to $\mathbf{p}$; i.e.,

$$\mathbf{p}^\star(\boldsymbol{\theta}) = \arg\max_{\mathbf{p}} p(\mathbf{p}|\boldsymbol{\theta},\mathbf{z}) = \boldsymbol{\mu}_{\boldsymbol{\theta}}^\circ. \tag{3.36}$$

As in any Gaussian density, the mean of $p(\mathbf{p}|\boldsymbol{\theta},\mathbf{z})$ is equal to its mode, and therefore $\boldsymbol{\mu}_{\boldsymbol{\theta}}^\circ$ is the solution of (3.36). Maximizing $p(\mathbf{p},\boldsymbol{\theta}|\mathbf{z})$ subject to $\mathbf{p} = \boldsymbol{\mu}_{\boldsymbol{\theta}}^\circ$ is equivalent to a NLS problem that can be solved

as before. After obtaining the MAP estimate for $\boldsymbol{\theta}$, we can instantly recover $\mathbf{p}^\star$ by evaluating (3.36) at $\boldsymbol{\theta}^\star$. Nevertheless, in practice we should use the approach taken in Algorithm 3 in order to retain the sparsity of SLAM.

3.7 Implementation

In this section we outline an efficient implementation of sparse VP that requires minor modification of existing SLAM solvers such as g2o [102]. In each iteration of sparse VP, we need to construct and solve the sparse linear system in (3.25). First, note that there is a close relation between the coefficient matrix in the left hand side of (3.25) and the position-position block ($\mathbb{I}_p$) in the full Hessian (3.21). It can be easily verified that these two terms are the same expression evaluated at different points. The former is evaluated at $\boldsymbol{\theta}_{(i+1)}$, whereas the latter is computed at $\boldsymbol{\theta}_{(i)}$. In other words,

$$\tilde{\mathbf{H}}_1^\top \tilde{\mathbf{H}}_1 = \sum_{k=1}^{|\mathcal{E}|} \frac{\partial \mathbf{r}_k}{\partial \mathbf{p}}^\top \boldsymbol{\Sigma}_k^{-1} \frac{\partial \mathbf{r}_k}{\partial \mathbf{p}}. \tag{3.37}$$

The Jacobians appearing in (3.37) are already available in any conventional Newton-based SLAM solver. Our implementation benefits from this insight by relying on g2o for computing the coefficient matrix of (3.25). Similar to g2o, we exploit the block structure of the coefficient matrix which enables us to process the information terms for edges in parallel. The right-hand side of (3.25) can also be computed using the existing routines for computing the right-hand side of the full normal equations (3.21). Finally, we can use any of the existing sparse linear solvers to solve (3.25) efficiently (CHOLMOD [28] in our case).

Although we do not require any special structure in $\boldsymbol{\Sigma}$, here we point out an interesting property of spherical noise covariance matrices. Let $\boldsymbol{\Sigma}_{p_i}$ denote the covariance matrix of the translational component of the ith measurement. An interesting special case emerges when $\boldsymbol{\Sigma}_{p_i}$ is *spherical* (isotropic), i.e., $\boldsymbol{\Sigma}_{p_i} = \sigma_{p_i}^2 \mathbf{I}_2$. Noting that $\mathbf{R}_\theta$ is orthogonal, it is easy to show that in such SLAM problems, $\tilde{\mathbf{H}}_1^\top \tilde{\mathbf{H}}_1 = \mathbf{L}_w \otimes \mathbf{I}_2$ in which $\mathbf{L}_w$ is the reduced weighted Laplacian matrix of graph $\mathcal{G}$ with the weight function $w : \mathcal{E} \to \mathbb{R}_{>0}$ defined as $w : e_i \mapsto \sigma_{p_i}^{-2}$. A similar structure exists in 3D SLAM problems with spherical noise covariance matrices. Consequently, in such cases the coefficient matrix of (3.25) remains constant and therefore it needs to be factorized only once at the first iteration. After computing the sparse Cholesky factor once, the following VP iterations only need to solve sparse triangular linear systems with different right-hand sides using backward and forward substitutions. As shown in Section 3.8, exploiting this structure significantly reduces the extra cost of VP.

3.8 Experimental Results

We performed experiments on both real and synthetic situations, including six publicly available datasets, in order to evaluate the performance of the proposed algorithm. We have used g2o's implementation of GN [102]. CHOLMOD [28] is used as the linear solver with the approximate minimum degree (AMD) ordering. Our Algorithm 3 (VP)[6] is implemented in C++ as a g2o solver. Our fully integrated code is available at `github.com/kasra/vp-g2o`.[7] An Intel Core i5-2400 CPU operating at 3.1GHz is used for all of the experiments in this chapter. We verified the equivalence between the iterations of Algorithm 1, Algorithm 2 and Algorithm 3 numerically.

We used g2o's 2D simulator to create Manhattan-like pose-graph datasets. This simulator creates a random walk in $\mathbb{R}^2 \times [-\pi,\pi)$ with either 1 meter forward motion or 90° rotation per step. The valid sensor range for scan matching is between 1 and 5 meters within the 135° field of view. In reality, scan matching is an expensive operation. Therefore extracting each and every (potential) loop closure is practically intractable. We imitate this practical limitation by imposing an upper bound on the degree of each vertex in the simulator. For our Monte Carlo analysis we created `Atlas`, a collection of 500 simulated datasets available at `http://github.com/kasra/atlas`. `Atlas` consists of five test suites, each of which is composed of 100 randomly generated pose-graph datasets with 10^4 poses per dataset. Each test suite corresponds to a fixed noise level $\alpha \in \{1,\ldots,5\}$. Noise covariance for each suite is $\mathbf{\Sigma}_\alpha = (0.01\alpha)^2\mathbf{I}$.

3.8.1 Convergence

For each dataset, we computed 50 iterations of different solvers. Solvers are initialized using odometry data. The outcome of each run is one of the following.

(i) **Global Min**: The global minimizer is found within 50 iterations.

(ii) **Local Min**: A local minimizer (other than the global minimizer) is found within 50 iterations.

(iii) **Not Converged**: The solver failed to converge before 50 iterations.

We treat the solution of GN initialized with the ground truth as the global minimizer. We verify if an obtained solution is the global minimizer by comparing it to the solution of Gauss-Newton initialized with the ground truth. If the absolute difference between the cost of the last two iterations is larger than a small threshold, we categorize that case as an instance of **Not Converged**. Similarly, if

[6] In this section VP refers to our proposed algorithm, not to be confused with the original or Kaufman's VP algorithms.
[7] Refer to our project's page on GitHub for instructions.

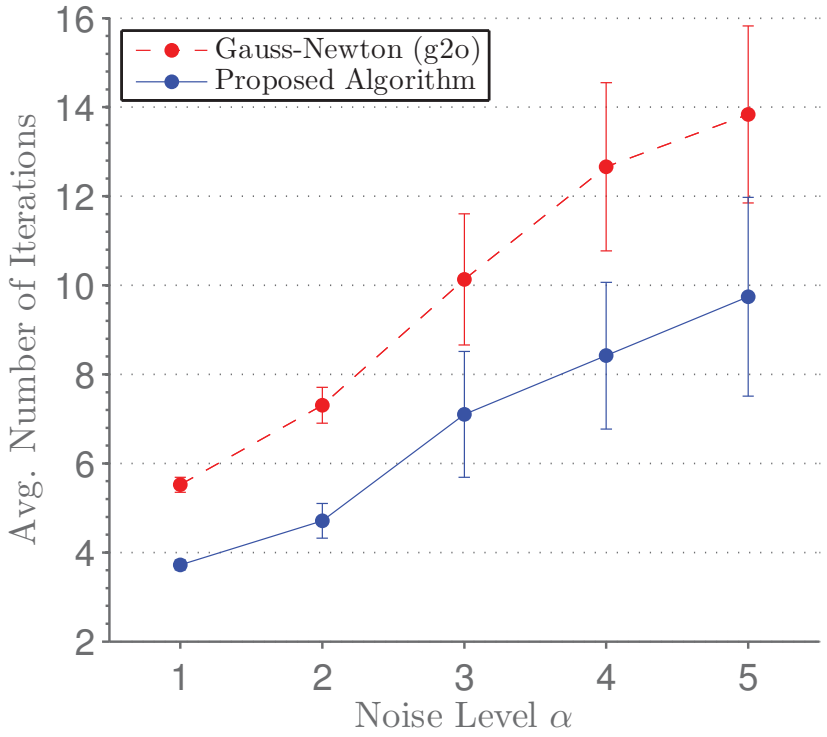


Figure 3.3: Average number of iterations performed to converge to the global minimum (ML estimate) under different noise levels (α) in the `Atlas` datasets. For each noise level, 100 random datasets with 10,000 poses has been generated. The error bars show the 95% confidence interval. The increasing length of error bars is partly due to the decreasing number of successful samples (see Table 3.2). Note that there is a one-to-one correspondence between iterations of Algorithm 3 and those of Algorithm 1 performed on the reduced problem (3.13).

the absolute difference between the final cost and minimum of the NLS cost function is larger than a small threshold, we consider it as an instance of **Local Min**.

Table 3.2 summarizes the outcome under different noise levels. Although in some instances VP outperforms GN, in general the two algorithms exhibit comparable performances in terms of converging to the optimal solution. As expected, both algorithms tend to converge to local minima as α increases; a "good" initial guess is crucial for converging to the optimal solution. Nevertheless, according to Table 3.2 our algorithm significantly outperforms GN in avoiding divergence or extremely slow convergence. Therefore, as reported by other researchers in various fields [61], VP iterations lead to a faster and more reliable convergence than solving the full NLS problem. As any other iterative solver, using VP can lead to a local minimizer other than MLE. This can be avoided by using a "sufficiently good" initial guess [73]. It is worth noting that in the case of converging to local minima, the results obtained by both solvers were generally inaccurate and far from the optimal estimate. Out of the 500 `Atlas` synthetic datasets used in Table 3.2, there are 89 cases for which both GN and VP converge to local minima. In 38 of those instances, GN and VP converge to the *same* local

Table 3.2: Outcome (%) of GN and VP after 50 iterations under different noise levels.

Noise Level	Solver	Global Min	Local Min	Not Converged
$\alpha = 1$	GN	100	0	0
	VP	100	0	0
$\alpha = 2$	GN	91	8	1
	VP	94	6	0
$\alpha = 3$	GN	76	16	8
	VP	78	19	3
$\alpha = 4$	GN	56	36	8
	VP	57	41	2
$\alpha = 5$	GN	37	50	13
	VP	39	60	1

Table 3.3: Outcome (%) of LM and VP-LM after 50 & 100 iterations under different noise levels.

Noise Level	Solver	Global Min		Local Min		Not Converged	
		50 iter.	100 iter.	50 iter.	100 iter.	50 iter.	100 iter.
$\alpha = 1$	LM	53	79	10	7	37	14
	VP-LM	97	97	3	3	0	0
$\alpha = 2$	LM	17	41	22	13	61	46
	VP-LM	90	90	10	10	0	0
$\alpha = 3$	LM	9	18	18	25	73	57
	VP-LM	72	73	27	27	1	0
$\alpha = 4$	LM	1	8	15	24	84	68
	VP-LM	48	48	49	52	3	0
$\alpha = 5$	LM	2	6	16	24	82	0
	VP-LM	32	33	63	66	5	1

Table 3.4: Total run time for 10 random walks with 10^5 poses and non-spherical noise with marginal variances of $\alpha = 1$ noise level. See also Figure 3.5.

#	$\lvert\mathcal{E}\rvert$	GN (sec)	$\text{VP}_{\gamma T=0.2}$ (sec)	Saving vs. GN (%)	VP (sec)	Saving vs. GN (%)
1	344,364	15.14	14.99	1.01	15.53	−2.58
2	343,436	15.99	12.60	21.17	13.00	18.70
3	346,019	31.52	25.93	17.73	26.85	14.82
4	345,249	24.63	20.41	17.15	20.33	17.44
5	343,864	–	91.53	–	95.41	–
6	345,842	29.45	24.35	17.32	24.56	16.59
7	345,864	22.41	22.74	−1.51	23.51	−4.94
8	343,642	18.52	16.68	9.95	17.19	7.21
9	344,730	24.84	18.45	25.72	19.15	22.92
10	343,685	18.31	16.93	7.51	17.76	3.00

minimum. Furthermore, in 74 of those (89) cases, the local minima found by GN and VP are both at least 10% larger than the true minimum. In addition to the results shown in Table 3.2, we performed

another experiment by generating random initial guesses in a neighbourhood of MLE. Random initial guesses were generated by sampling uniformly from the surface of hyper spheres centered at the global minimum with varying radii. As in Table 3.2, we did not observe a statistically significant difference in the tendency of VP and GN for converging to local minima.

Table 3.3 shows the results obtained by Levenberg-Marquardt (LM) and a trust-region version of Algorithm 3 using LM (VP-LM). Note that VP-LM is equivalent to performing LM on the reduced cost function under Kaufman's approximation. For both solvers we use g2o's default settings (e.g., strategy to update the damping parameter). Due to the slow progress of LM, here we report the results after both 50 and 100 iterations. According to Table 3.3, LM exhibits extremely slow convergence while the performance of VP-LM within 50 iterations is comparable to that of GN and VP. Nevertheless, the success rates of GN and VP are slightly higher than VP-LM. It is also remarkable how VP-LM, due to its trust-region strategy, avoids divergence after a sufficient number of iterations.

Figure 3.3 shows the average number of iterations performed to converge to the optimal solution under different noise levels. It clearly indicates that the proposed algorithm can converge to the optimal solution in less number of iterations than GN. To correctly interpret this result, it is crucial to note that there is a one-to-one correspondence between iterations of Algorithm 3 and those of Algorithm 1 performed on the reduced problem (3.13). This observation is consistent with numerous reports from other researchers who apply variable projection to separable NLS problems in other contexts [61].

3.8.2 Run Time

Reducing the number of iterations does not necessarily reduce the total computation time since each VP iteration is more costly than that of GN. In fact for SLAM, original VP [60], Kaufman's approach [86] (Algorithm 1) and Barham and Drane's method [7] (Algorithm 2) are all significantly slower than the state-of-the-art SLAM solvers since they are all incapable of exploiting sparsity. Unlike these algorithms, Algorithm 3 is designed to retain the sparse structure of SLAM. Nevertheless, recall that in each iteration of our algorithm, compared to GN, we have an additional (but smaller) sparse linear system to solve. Therefore each iteration of our algorithm is still slightly more expensive than that of GN. Informally speaking, Figure 3.3 suggests that by exploiting the separable structure of SLAM we can achieve more *effective* iterations at the cost of solving an additional sparse linear system in each iteration.

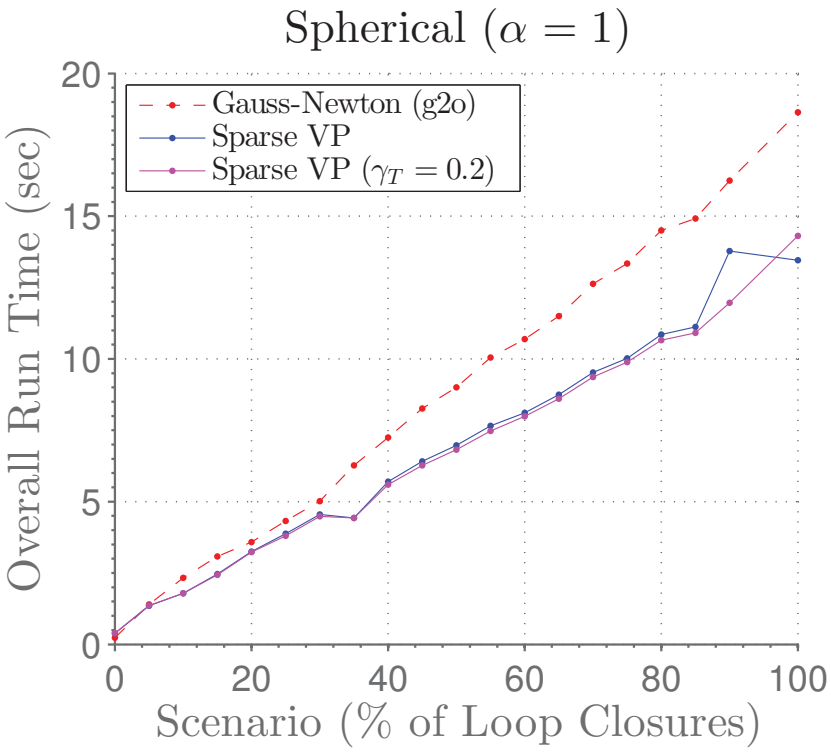
To compare the overall run times, we generated 10 large-scale Manhattan-like random walks,

each with 10^5 poses. Relative measurements are corrupted by additive Gaussian noise with a non-spherical covariance matrix. The marginal variances are equal to those of $\alpha = 1$ noise level.[8] Each of these datasets were solved iteratively using GN, VP and VP with the relative projection gain threshold $\gamma_T = 0.2$ (see Section 3.5). The average run times are listed in Table 3.4 and illustrated in Figure 3.5. In all but one of the datasets, all algorithms converged to the optimal solution. In all of the datasets, except two cases, VP with γ_T exhibits the fastest performance, while vanilla VP outperforms GN in 8 datasets. VP with γ_T outperforms GN because of its more effective iterations on the nonlinear core of SLAM (i.e., projection steps). On the other hand, as expected, it outperforms vanilla VP by avoiding unnecessary costly projection steps when the gain is insignificant.

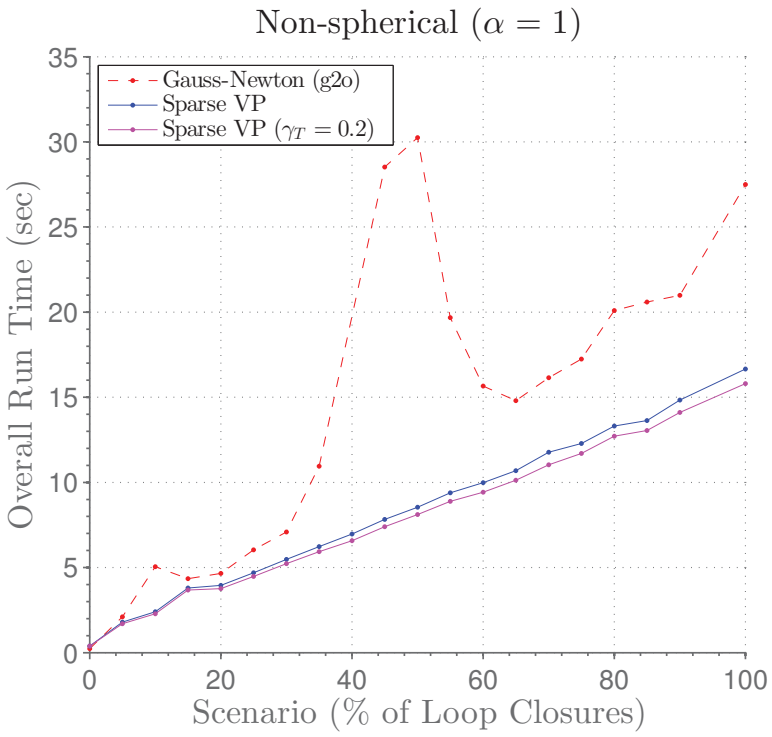
We conducted another experiment to compare the overall run time of VP and GN under varying edge density. For this purpose, first we created a Manhattan-like random walk with 10^5 poses. Two pose-graph datasets were created based on this random walk by simulating noisy measurements for the following noise models: (i) $\alpha = 1$, (ii) non-spherical noise covariance matrix with marginal variances similar to $\alpha = 1$. To study the effect of edge density on the overall run time we created 100 scenarios based on each simulated dataset. The ith scenario contains odometry edges plus i percent of the loop closures of the original simulation (including the ones that were included in the $(i-1)$th scenario). Note that the original simulation is a realistic sparse SLAM problem with a density similar to commonly used benchmarks. Loop closures are selected randomly (i.e., with no particular order) to achieve realistic and balanced scenarios. Figure 3.4 shows the overall run time as a function of loop closure density for each noise model. According to Figure 3.4, in the vast majority of cases, VP has been faster than GN. This shows that in those cases, reducing the number of iterations has paid off the additional cost paid for each iteration. There are few cases where GN is slightly faster than VP. This situation occurs mainly in extremely trivial (sparse) scenarios, in which the initial guess based on odometry is already close to MLE and thus GN can find the solution immediately. Such scenarios are often too sparse to be considered as realistic cases. Once again our results indicate that taking the (expected) projection gain into account is generally beneficial as it helps to avoid unnecessary projection steps.

Datasets with spherical noise covariance matrices possess an additional structure that can be exploited to significantly reduce the cost per iteration of Algorithm 3. Recall that in each iteration of our algorithm we need to solve an additional linear system to recover the conditionally-optimal estimate of linear variables. The cost of this extra step is dominated by the Cholesky factorization

[8] Given the large number of poses, higher noise levels lead to convergence failure (for both GN and VP) and hence are not considered here.



(a) $\alpha = 1$ (spherical).



(b) $\alpha = 1$ (non-spherical).

Figure 3.4: Overall run time for converging to MLE as a function of edge density under different noise models. In Figure 3.4b, GN failed to converge to MLE in one scenario (40). Note that the "100% loop closure density" refers to the 100th scenario in which all of the loop closures of a *realistic* sparse SLAM problem with $|\mathcal{V}| = 10^5$ and $|\mathcal{E}| \approx 342{,}000$ are included—not the complete graph.

of $\tilde{\mathbf{H}}_1^\top \tilde{\mathbf{H}}_1$ in (3.25). According to Section 3.7, this term is constant (i.e., independent of the current estimate) when the noise covariance matrix is spherical. Thus, the Cholesky factorization needs to be done only *once* in such problems (i.e., for the first iteration). In the rest of iterations we only need to solve sparse triangular linear systems by forward and backward substitutions in order to recover the conditionally-optimal $\mathbf{p}$. As illustrated in Figure 3.4, this trick can reduce the cost per iteration and overall run time of our algorithm.

We also used a number of publicly available datasets to evaluate the performance of the proposed algorithm. Table 3.5 provides the number of iterations performed to find the optimal solution, as well as the average of total computation time over 10 runs. The datasets listed in Table 3.5 span the most common forms of SLAM (2D/3D real/synthetic pose-graphs). As expected, VP converges to the optimal estimate in less number of iterations than GN (up to 50%). In most cases VP also outperforms GN in terms of the total computation time (up to 30%). Small datasets and accurate measurements make the SLAM problem less challenging in terms of convergence [20]. In such cases, the computational benefits of exploiting the separable structure of SLAM can be less than some more challenging scenarios. This conclusion is consistent with what we saw earlier in Figure 3.4.

For the datasets listed in Table 3.5, both algorithms are able to converge to MLE starting from the odometry initial guess. This is true for most of the existing real datasets, although as seen in Table 3.2 for synthetic datasets, a bad initial guess can cause VP and GN to converge to (different) local minimizers of the cost function. For example this is the case for the Victoria Park dataset. Starting from the same initial guess and after 30 iterations, VP and GN achieve $f_{\text{VP}} = 13{,}659$ and $f_{\text{GN}} = 30{,}611$, respectively. Similar to what we observed before in our Monte Carlo simulations, even without using a line search, the objective value resulting from VP directions follows a stable decreasing trend, while in the first few iterations, GN steps cause the cost function to increase.

3.9 Discussion

In this chapter, we mainly considered batch solvers. Incrementally solving SLAM [84, 85, 131] is more suitable for online applications. The separable structure of SLAM is preserved in incremental formulation of SLAM and can be exploited by the same principles and techniques introduced in this chapter. As seen earlier, our results indicate that exploiting separability, through increasing the convergence rate, is mostly beneficial if the initial guess is not already "too close" to the solution. One advantage of incrementally solving SLAM is that one can use the ML estimate using the data

Table 3.5: Summary of results for some of the publicly available real and synthetic datasets. In all cases both algorithms converge to the maximum likelihood estimate.

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	Solver	# Iter.	Time (s)
City10K	10,000	20,678	GN	7	0.41068
			VP	4	0.31617
Manhattan	3,500	5,598	GN	6	0.07907
			VP	4	0.07200
Intel	943	1,837	GN	3	0.01463
			VP	2	0.01607
UTM Downtown	14,549	16,365	GN	10	0.24754
			VP	4	0.17719
Sphere2500	2,500	9,799	GN	5	0.96145
			VP	4	0.82080
New College	52,480	52,577	GN	8	2.19560
			VP	6	2.00967

collected up to time $t-1$ for finding the ML estimate at time t. This initial guess is usually good and therefore exploiting separability may not be useful in such cases. An important exception is when the most recent measurements (measurements collected at time t) lead to a significant update of robot trajectory. For instance, this situation could arise if, for a period of time, the robot has only access to noisy odometry data (dead reckoning) and small loop-closures, and then suddenly closes a larger loop. Incremental solvers that are capable of exploiting separability (using the solver proposed in this chapter) can therefore benefit from its faster convergence in such scenarios. Recall that if the initial guess is already close to the solution, the projection gain γ will be small. Therefore, immediately after the first projection step and by computing the projection gain, the proposed switching scheme in Section 3.5 can automatically detect whether or not exploiting separability is beneficial. In the worst case, after performing a single projection step, our algorithm may decide not to exploit separability by switching to Gauss-Newton steps. However, if the projection gain happens to be significant (e.g., in the case of closing larger loops), an incremental solver that is capable of exploiting separability will exhibit faster convergence.

3.10 Summary

In this chapter, we proposed a scalable and efficient algorithm to take advantage of the separable structure of SLAM. It was shown that by exploiting this structure, we can achieve faster and more reliable convergence than the state-of-the-art solvers. A key contribution of this work comes from establishing the link to the rich literature on separable NLS problems. In particular, recognizing the

equivalence between Algorithm 1 and Algorithm 2 was the missing link that enabled us to retain sparsity while exploiting the separable structure through Algorithm 3. This link also provides a firm theoretical justification for the proposed algorithm. Moreover, we demonstrated how one can avoid performing insignificant projections by taking the projection gain into consideration.

The proposed algorithm can be applied to the most common forms of SLAM (2D/3D feature-based and pose-graphs) without any restrictive assumption on the structure of the noise covariance matrix. Our algorithm is not limited to a particular type of NLS solver and the benefits it brings along are orthogonal to those of other possible improvements such as a more efficient implementation (of e.g., GN) or using different Newton-based solvers, trust-region or line search techniques. As an advantage, our final algorithm can be easily adopted by the existing back-ends (e.g., LM, Powell's Dog-leg [131], etc) without any major modification. By stripping down SLAM to its nonlinear core and recovering the conditionally-optimal estimate for linear variables, our approach yields more effective and reliable iterations than solving the full NLS problem. The number of iterations required for solving the reduced problem (3.13) was shown to be less than that of the full NLS problem. Exploiting separability is especially beneficial when GN (or other Newton-based solver) iterations are relatively costly and/or when it takes more than a few iterations to solve the full NLS problem. Datasets with relatively high measurement noise and bad initial guess are among those cases.

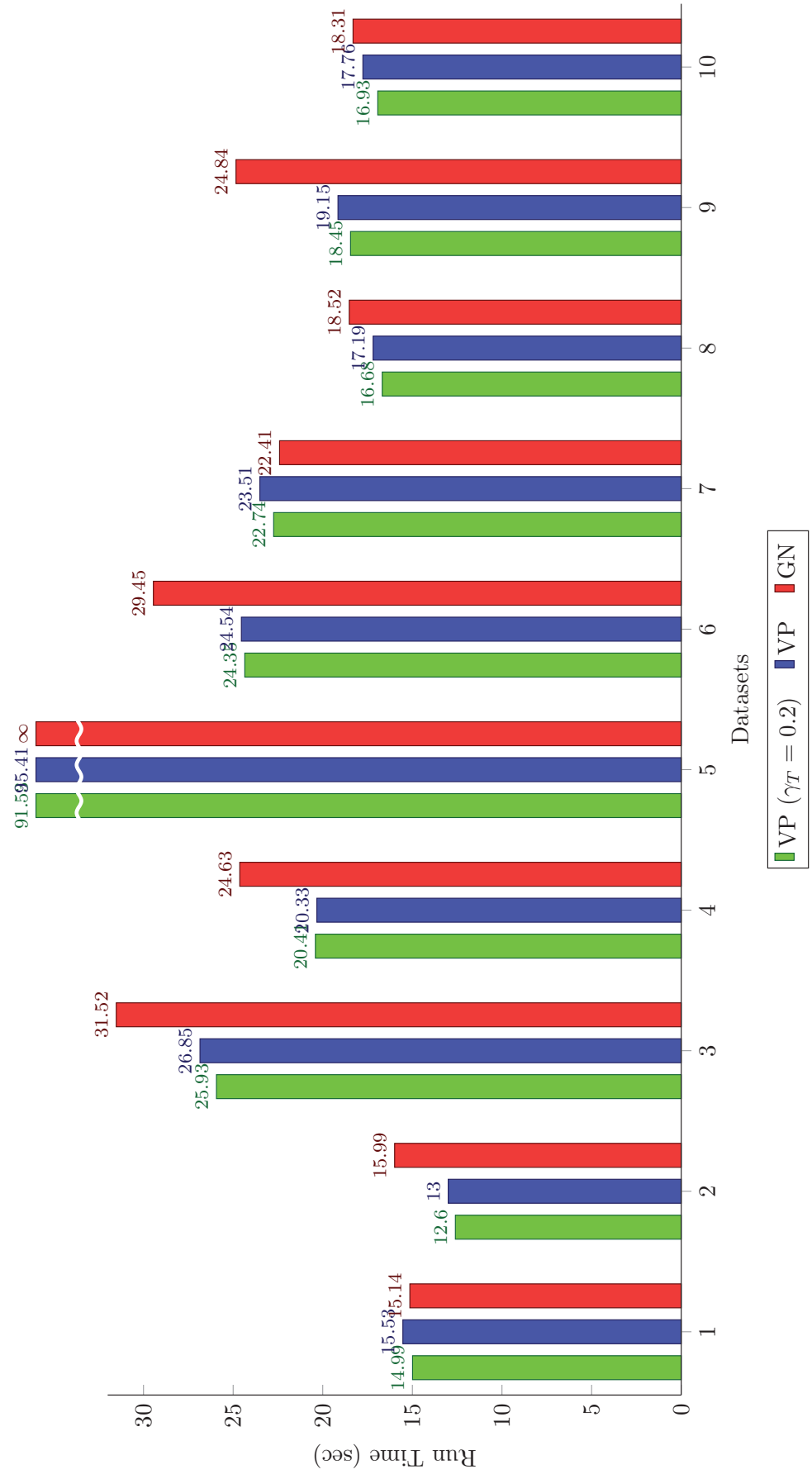


Figure 3.5: An illustration of the run times listed in Table 3.4.

Part II

SLAM: Estimation over Graphs

CHAPTER 4

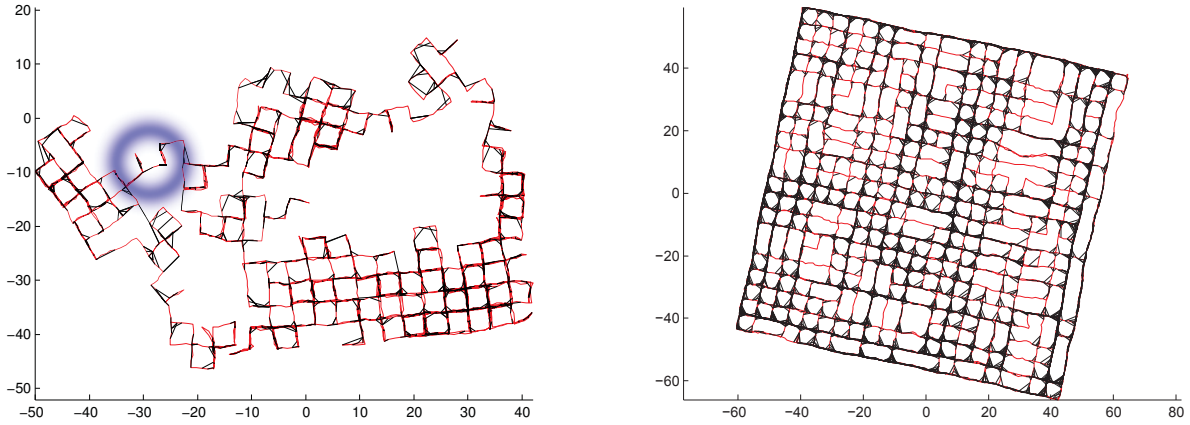
Estimation over Graphs

Parameter estimation problems are often characterized by their measurement models. In robotics, a measurement model usually corresponds to a probabilistic model for a specific sensor. These models are normally described by (i) a nominal noise-free measurement function, and (ii) a probability distribution according to which random noise corrupts the nominal measurement. As we saw earlier, any instance of SLAM can be represented by a graph that encodes "who's observing whom". The topology of the graphical representation of SLAM is another axis along which a particular instance of SLAM can be characterized.

The graphical structure of SLAM influences the estimation error covariance of the maximum likelihood estimator. This fact is illustrated in Figure 4.1. This figure shows the maximum likelihood estimates for two synthetic pose-graph SLAM datasets. These datasets are regenerated with identical measurement functions and noise models. In both datasets, the robot performs a random walk in $\mathbb{R}^2 \times [-\pi, \pi)$ with either forward motion or 90° rotation per step. It is evident that the final estimation errors in Figure 4.1b are smaller than the estimation errors in Figure 4.1a. To be more precise, the estimation error covariance in the City10K is "smaller" than that of Manhattan. Intuitively, we can conclude that the graphical structure of City10K makes it more *resilient* to noise than Manhattan. Graphical representation of SLAM, despite its simplicity, can be highly informative. Our goal in this chapter is to reveal and analyze the impact of the topology of the graphical representation of SLAM and similar problems on the estimation error covariance of the maximum likelihood estimator.

Outline

In Section 4.1, we mathematically formulate several estimation-over-graph (EoG) problems. In Section 4.2, we present our main results on the influence of different notions of graph connectivity on



(a) Manhattan [121] **(b)** City10K [83]

Figure 4.1: Maximum likelihood estimates for two datasets with identical noise models. The bridges highlighted in the Manhattan dataset are good examples of its weak connectivity.

the quality of estimation. In particular, we consider average degree, algebraic connectivity and tree connectivity. Finally, in Section 4.3, we empirically validate our theoretical results using real and synthetic datasets.

4.1 Estimation-over-Graph Problems

In this section, we mathematically formulate several classes of EoG problems.

4.1.1 Diff-Net

Diff-Net (difference-network) is one of the simplest classes of linear-Gaussian EoG problems. Diff-Net is motivated by the time-synchronization problem in sensor networks [8]. Let $\{\mathbf{x}_i\}_{i=1}^n$ represent the unknown states and $\{\mathbf{z}_i\}_{i=1}^m$ represent the pairwise noisy measurements. In Diff-Net, the pairwise measurement between $\mathbf{x}_u$ and $\mathbf{x}_v$ (in $\mathbb{R}^d$ for $d \in \mathbb{N}$) is generated according to $\mathbf{z}_{uv} = \mathbf{x}_u - \mathbf{x}_v + \boldsymbol{\epsilon}_{uv}$ in which $\boldsymbol{\epsilon}_{uv}$ is a zero-mean Gaussian noise.

Assumption 4.1.1 (Diff-Net Noise)**.** *Let $\boldsymbol{\epsilon}_i$ be the noise corrupting the ith measurement. We assume that,*

1. $\mathbb{C}\text{ov}[\boldsymbol{\epsilon}_i,\boldsymbol{\epsilon}_j] = \mathbf{0}_{d\times d}$ *if and only if* $i \neq j$.
2. $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0},\sigma_i^2\mathbf{I}_d)$.

A Diff-Net with n variables can be naturally represented by the graph $\mathcal{G} = ([n],\mathcal{E})$:

- Variable $\mathbf{x}_i$ is represented by the vertex $i \in [n]$.

- Relative measurement $\mathbf{z}_{ij}$ is represented by the edge $\{i,j\} \in \mathcal{E}$.
- Edges are weighted by the precision (inverse of variance) of the corresponding measurements; i.e.,

$$w : \mathcal{E} \to \mathbb{R}_{>0} \tag{4.1}$$

$$e_k \mapsto \sigma_k^{-2}. \tag{4.2}$$

As we saw earlier in Chapter 3, due to the relative nature of measurements, we need to anchor a vertex. Let $\mathbf{A}$ be the reduced incidence matrix of $\mathcal{G}$. Let $\mathbf{x}$ and $\mathbf{z}$ denote the stacked vector of states (after anchoring an arbitrary vertex) and measurements, respectively. The measurement model can be written as,

$$\mathbf{z} = (\mathbf{A} \otimes \mathbf{I}_d)^\top \mathbf{x} + \boldsymbol{\epsilon}, \tag{4.3}$$

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 $\mathbf{W} \triangleq \operatorname{diag}\Big(w(e_1), \ldots, w(e_m)\Big)$.

Proposition 4.1.1. *The Fisher information matrix in Diff-Net is given by $\mathbb{I} = \mathbf{L}_w \otimes \mathbf{I}_d$ in which $\mathbf{L}_w$ is the reduced weighted Laplacian matrix of $\mathcal{G}$.*

Proof. Let $\mathbf{H} \triangleq (\mathbf{A} \otimes \mathbf{I}_d)^\top$ denote the measurement function in Diff-Net. Plugging $p(\mathbf{z}; \mathbf{x}) = \mathcal{N}(\mathbf{z}; \mathbf{H}\mathbf{x}, \boldsymbol{\Sigma})$ into the definition of the Fisher information matrix (A.7) results in

$$\mathbb{I} = \mathbf{H}^\top \boldsymbol{\Sigma}^{-1} \mathbf{H} \tag{4.4}$$

$$= (\mathbf{A} \otimes \mathbf{I}_d)(\mathbf{W} \otimes \mathbf{I}_d)(\mathbf{A} \otimes \mathbf{I}_d)^\top \tag{4.5}$$

$$= (\mathbf{A}\mathbf{W}\mathbf{A}^\top) \otimes \mathbf{I}_d \tag{4.6}$$

$$= \mathbf{L}_w \otimes \mathbf{I}_d. \tag{4.7}$$

□

Remark 6. *The maximum likelihood estimator in Diff-Net, as a linear-Gaussian estimation problem, is unbiased and efficient.*

4.1.2 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. [44][45] proposed one of the early SLAM algorithms based on this model. The goal in Compass-SLAM is to estimate the robot 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, for which the globally-optimal solution can be computed easily by solving a linear least squares problem.

Remark 7. *The maximum likelihood 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 Diff-Nets. 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.,

$$\mathbf{R} \triangleq \mathrm{diag}\Big(\mathbf{R}_1,\ldots,\mathbf{R}_m\Big). \tag{4.8}$$

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

$$\mathbf{z} = \mathbf{R}^\top(\mathbf{A}\otimes\mathbf{I}_d)^\top\mathbf{p} + \boldsymbol{\epsilon} \tag{4.9}$$

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 4.1.2 (Compass-SLAM Noise)**.** *Let $\boldsymbol{\epsilon}_i$ be the noise affecting the ith measurement. We assume that,*

1. $\mathbb{C}\mathrm{ov}[\boldsymbol{\epsilon}_i,\boldsymbol{\epsilon}_j] = \mathbf{0}_{d\times d}$ *if and only if* $i \neq j$.
2. $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0},\sigma_i^2\mathbf{I}_d)$.

Proposition 4.1.2. *The Fisher information matrix 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 $\mathcal{G}$.*

Proof. The proof is similar to that of Proposition 4.1.1. 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 Fisher information matrix (A.7) results in

$$\mathbb{I} = \mathbf{H}^\top \boldsymbol{\Sigma}^{-1}\mathbf{H} \tag{4.10}$$

$$= (\mathbf{A}\otimes\mathbf{I}_d)\mathbf{R}\boldsymbol{\Sigma}^{-1}\mathbf{R}^\top(\mathbf{A}\otimes\mathbf{I}_d)^\top \tag{4.11}$$

$$= (\mathbf{A}\otimes\mathbf{I}_d)(\mathbf{W}\otimes\mathbf{I}_d)(\mathbf{A}\otimes\mathbf{I}_d)^\top \tag{4.12}$$

$$= (\mathbf{A}\mathbf{W}\mathbf{A}^\top)\otimes\mathbf{I}_d \tag{4.13}$$

$$= \mathbf{L}_w\otimes\mathbf{I}_d. \tag{4.14}$$

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

4.1.3 SLAM

This section is concerned mainly with the 2D pose-graph SLAM problem with *relative-pose* measurements.[1] Similar to Chapter 3, the state vector is $\mathbf{x} \triangleq [\mathbf{p}^\top\ \boldsymbol{\theta}^\top]^\top$, and $\mathbf{z} = [\mathbf{z}_p^\top\ \mathbf{z}_\theta^\top]^\top$ is the stacked vector of translational and rotational measurements. The measurement model is defined in Section 3.2.2. According to this model,

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \boldsymbol{\epsilon}, \tag{4.15}$$

in which $\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma})$. The measurement function $\mathbf{h}$ is given by,

$$\mathbf{h}(\mathbf{x}) \triangleq \begin{bmatrix}\mathbf{h}_p(\mathbf{x})\\ \mathbf{h}_\theta(\boldsymbol{\theta})\end{bmatrix} \tag{4.16}$$

where $\mathbf{h}_p(\mathbf{x}) \triangleq \mathbf{R}^\top(\mathbf{A}\otimes\mathbf{I}_2)^\top\mathbf{p}$ and $\mathbf{h}_\theta(\boldsymbol{\theta}) \triangleq \mathbf{A}^\top\boldsymbol{\theta}$. In this section, we make an extra assumption regarding the structure of the noise covariance matrix.

Assumption 4.1.3 (Block-Isotropic Noise)**.** *We assume the noise covariance matrix* $\boldsymbol{\Sigma}$ *can be written as* $\boldsymbol{\Sigma} = \mathrm{diag}(\boldsymbol{\Sigma}_p,\boldsymbol{\Sigma}_\theta)$*, where*

$$\boldsymbol{\Sigma}_p = \mathrm{diag}\left(\sigma_{p_1}^2\mathbf{I}_2,\ldots,\sigma_{p_m}^2\mathbf{I}_2\right) \tag{4.17}$$

$$\boldsymbol{\Sigma}_\theta = \mathrm{diag}\left(\sigma_{\theta_1}^2,\ldots,\sigma_{\theta_m}^2\right). \tag{4.18}$$

[1] Extending our models/results to 2D feature-based SLAM is straightforward as noted in Chapter 3.

According to (A.7), for $p(\mathbf{z};\mathbf{x}) = \mathcal{N}(\mathbf{z};\mathbf{h}(\mathbf{x}),\boldsymbol{\Sigma})$ the Fisher information matrix is given by

$$\mathbb{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})^\top \boldsymbol{\Sigma}^{-1}\mathbf{J}(\mathbf{x}), \tag{4.19}$$

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:

$$\mathbf{J}(\mathbf{x}) \triangleq \frac{\partial\mathbf{h}(\mathbf{x})}{\partial\mathbf{x}} = \begin{bmatrix} \mathbf{J}_p^p & \mathbf{J}_\theta^p \\ \mathbf{J}_p^\theta & \mathbf{J}_\theta^\theta \end{bmatrix} \tag{4.20}$$

$$\mathbf{J}_p^p \triangleq \frac{\partial\mathbf{h}_p}{\partial\mathbf{p}} = \mathbf{R}^\top(\mathbf{A}\otimes\mathbf{I}_2)^\top \tag{4.21}$$

$$\mathbf{J}_\theta^p \triangleq \frac{\partial\mathbf{h}_p}{\partial\boldsymbol{\theta}} = \widetilde{\mathbf{R}}\boldsymbol{\Delta} \tag{4.22}$$

$$\mathbf{J}_p^\theta \triangleq \frac{\partial\mathbf{h}_\theta}{\partial\mathbf{p}} = \mathbf{0} \tag{4.23}$$

$$\mathbf{J}_\theta^\theta \triangleq \frac{\partial\mathbf{h}_\theta}{\partial\boldsymbol{\theta}} = \mathbf{A}^\top. \tag{4.24}$$

The only new terms are $\widetilde{\mathbf{R}}$ and $\boldsymbol{\Delta}$:

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

$$\boldsymbol{\Gamma} \triangleq \mathbf{I}_m \otimes \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{4.25}$$

  It is easy to verify that $\boldsymbol{\Gamma}^\top\boldsymbol{\Gamma} = \mathbf{R}^\top\mathbf{R} = \widetilde{\mathbf{R}}^\top\widetilde{\mathbf{R}} = \mathbf{I}$.

- $\boldsymbol{\Delta} \in \mathbb{R}^{2m\times n}$ has the following structure. Suppose in the kth measurement, the i_kth node has observed the j_kth node. Then, for each $e_k \in \mathcal{E}$, there is a 2×1 block in $\boldsymbol{\Delta}$ that contains

$$(\boldsymbol{\Delta})_{2k-1:2k,i_k} = \mathbf{p}_{j_k} - \mathbf{p}_{i_k}. \tag{4.26}$$

  The remaining elements in $\boldsymbol{\Delta}$ are all zero. As noted in [20], $\boldsymbol{\Delta}^\top\boldsymbol{\Delta}$ is a diagonal matrix with an interesting structure. $(\boldsymbol{\Delta}^\top\boldsymbol{\Delta})_{i,i}$ is equal to the sum of squared distances between the ith robot pose, and every node observed by it,

$$(\boldsymbol{\Delta}^\top\boldsymbol{\Delta})_{i,i} = \sum_{j\in\mathcal{S}(i)} \|\mathbf{p}_i - \mathbf{p}_j\|^2. \tag{4.27}$$

  Here $\mathcal{S}(v)$ is the set of nodes observed by $v \in \mathcal{V}$.

Now we can compute the Fisher information matrix $\mathbb{I}(\mathbf{x})$:

$$\mathbb{I}(\mathbf{x}) = \begin{bmatrix} \mathbf{L}_{w_p} \otimes \mathbf{I}_2 & (\mathbf{A}_{w_p} \otimes \mathbf{I}_2)\mathbf{\Gamma}\mathbf{\Delta}_{w_p} \\ *^\top & \mathbf{L}_{w_\theta} + \mathbf{\Delta}_{w_p}^\top \mathbf{\Delta}_{w_p} \end{bmatrix}, \tag{4.28}$$

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.
- $\mathbf{A}_{w_p} \triangleq \mathbf{A}\mathbf{W}_p^{\frac{1}{2}}$ is the reduced weighted incidence matrix of $\mathcal{G}$ when edges are weighted by w_p defined above.
- $\mathbf{\Delta}_{w_p} \triangleq \mathbf{W}_p^{\frac{1}{2}}\mathbf{\Delta}$. Similar to (4.27), $\mathbf{\Delta}_{w_p}^\top \mathbf{\Delta}_{w_p}$ is a diagonal matrix and we have,

$$(\mathbf{\Delta}_{w_p}^\top \mathbf{\Delta}_{w_p})_{i,i} = \sum_{j \in \mathcal{S}(i)} w_p(i,j) \|\mathbf{p}_i - \mathbf{p}_j\|^2. \tag{4.29}$$

4.2 Main Results

4.2.1 Average Degree $\leftrightarrow$ Log-Likelihood Ratio

In this section, we will demonstrate that the ratio between the value of log-likelihood cost function evaluated at the ground truth and the maximum likelihood estimate can be accurately approximated by a simple function that only depends on the structure of the underlying graph. To simplify our notation, let $\mathbf{x}^\star$ and $\mathbf{x}^\circ$ be the maximum likelihood estimate and the true value of $\mathbf{x}$, respectively. Olson and Kaess [123] looked at 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 4.2 illustrates these values in a toy example. Define $\gamma \triangleq f^\star / f^\circ$. It is easy to see that,

1. $0 \leq \gamma \leq 1$, and
2. $\gamma \to 1^-$ as $\mathbf{x}^\star \to \mathbf{x}^\circ$.

Hence, $\gamma \approx 1$ ($\gamma \approx 0$, respectively), indicates that the estimation error is relatively small (large, respectively), and therefore $\mathbf{x}^\star$ is (is not, respectively) a *reliable* estimate. Using Monte Carlo simu-

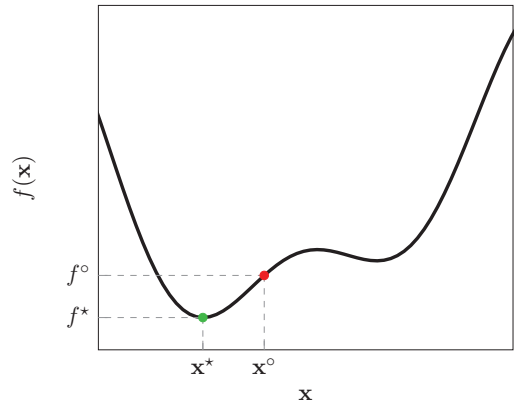


Figure 4.2: A toy example. The function drawn in black represents the negative log-likelihood objective function. The maximum likelihood estimate $\mathbf{x}^\star$ and the ground truth $\mathbf{x}^\circ$, together with their objective values, are specified in the figure.

lations, Olson and Kaess [123] empirically observed that as the average node degree of graph, i.e., $\bar{d} \triangleq \frac{1}{n}\sum_{i=1}^{n} \deg(i)$ increases, γ on average approaches 1 (see Figure 5 in [123]). According to their interpretation, γ is a "coarse measure of *overfitting*" [123]. We repeated their experiment and observed the same behavior. The blue points in Figure 4.3 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 [123]). 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 4.2.1 (Additive Gaussin 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 given in Assumption 4.2.1 can be written as $f(\mathbf{x}) = \|\mathbf{z} - \mathbf{h}(\mathbf{x})\|^2_{\boldsymbol{\Sigma}^{-1}}$. To compute γ, we need to compute both $f^\star$ and f°. Let us begin with 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 consequently $\boldsymbol{\epsilon}$. The following proposition gives the distribution of f°.

Proposition 4.2.1. *Under the Assumption 4.2.1,* $f^\circ \sim \chi^2_{\nu^\circ}$ *in which* $\nu^\circ \triangleq \dim(\mathbf{z})$.

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

[2]Without loss of generality we can assume the noise is zero-mean. Note that here we do not make any other assumptions regarding the structure of the noise covariance matrix $\boldsymbol{\Sigma}$.

According to Proposition 4.2.1, f° follows a $\chi^2_{\nu^\circ}$ distribution with $\dim(\mathbf{z})$ degrees of freedom. Now let us examine the behaviour 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 proposition provides the distribution of $f^\star$ when the measurement function is affine in $\mathbf{x}$.

Proposition 4.2.2. *Under the assumptions below, $f^\star \sim \chi^2_{\nu^\star}$ where $\nu^\star \triangleq \dim(\mathbf{z}) - \dim(\mathbf{x})$.*

1. *Assumption 4.2.1.*
2. *Affine measurement function $\mathbf{h}(\mathbf{x}) = \mathbf{H}\mathbf{x} + \mathbf{c}$ where $\mathbf{H}$ is full column rank.*

Proof. Define $\tilde{\mathbf{H}} \triangleq \boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{H}$, $\tilde{\mathbf{z}} \triangleq \boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{z}$, $\tilde{\mathbf{c}} \triangleq \boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{c}$ and $\tilde{\boldsymbol{\epsilon}} \triangleq \boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$. The maximum likelihood estimate is given by,

$$\mathbf{x}^\star = \underbrace{(\tilde{\mathbf{H}}^\top\tilde{\mathbf{H}})^{-1}\tilde{\mathbf{H}}^\top}_{\tilde{\mathbf{H}}^\dagger}(\tilde{\mathbf{z}} - \tilde{\mathbf{c}}), \tag{4.30}$$

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 $\mathbf{x} = \mathbf{x}^\star$,

$$\begin{aligned}
f(\mathbf{x}^\star) &= \|\mathbf{z} - \mathbf{H}\mathbf{x}^\star - \mathbf{c}\|^2_{\boldsymbol{\Sigma}^{-1}} && (4.31)\\
&= \|\tilde{\mathbf{z}} - \tilde{\mathbf{H}}\mathbf{x}^\star - \tilde{\mathbf{c}}\|^2 && (4.32)\\
&= \|\mathbf{z} - \tilde{\mathbf{c}} - \tilde{\mathbf{H}}\tilde{\mathbf{H}}^\dagger(\mathbf{z} - \tilde{\mathbf{c}})\|^2 && (4.33)\\
&= \|(\mathbf{I} - \tilde{\mathbf{H}}\tilde{\mathbf{H}}^\dagger)(\tilde{\mathbf{z}} - \tilde{\mathbf{c}})\|^2 && (4.34)\\
&= \|(\mathbf{I} - \tilde{\mathbf{H}}\tilde{\mathbf{H}}^\dagger)(\tilde{\mathbf{H}}\mathbf{x}^\circ + \tilde{\mathbf{c}} + \tilde{\boldsymbol{\epsilon}} - \tilde{\mathbf{c}})\|^2 && (4.35)\\
&= \|(\mathbf{I} - \tilde{\mathbf{H}}\tilde{\mathbf{H}}^\dagger)(\tilde{\mathbf{H}}\mathbf{x}^\circ + \tilde{\boldsymbol{\epsilon}})\|^2 && (4.36)\\
&= \|(\mathbf{I} - \tilde{\mathbf{H}}\tilde{\mathbf{H}}^\dagger)\tilde{\boldsymbol{\epsilon}}\|^2. && (4.37)
\end{aligned}$$

Now note that $\mathbf{I} - \tilde{\mathbf{H}}\tilde{\mathbf{H}}^\dagger$ is the orthogonal projection onto the nullspace of $\tilde{\mathbf{H}}^\top$ with dimension $r \triangleq \dim(\mathbf{z}) - \dim(\mathbf{x})$. Let $\mathbf{u}_1, \ldots, \mathbf{u}_r$ be an orthonormal basis for the nullspace of $\tilde{\mathbf{H}}^\top$. Therefore $(\mathbf{I} - \tilde{\mathbf{H}}\tilde{\mathbf{H}}^\dagger)\tilde{\boldsymbol{\epsilon}} = \sum_{i=1}^r (\mathbf{u}_i^\top\tilde{\boldsymbol{\epsilon}})\,\mathbf{u}_i$. Now note that,

$$\begin{aligned}
\|(\mathbf{I} - \tilde{\mathbf{H}}\tilde{\mathbf{H}}^\dagger)\tilde{\boldsymbol{\epsilon}}\|^2 &= \tilde{\boldsymbol{\epsilon}}^\top(\mathbf{I} - \tilde{\mathbf{H}}\tilde{\mathbf{H}}^\dagger)\tilde{\boldsymbol{\epsilon}} && (4.38)\\
&= \tilde{\boldsymbol{\epsilon}}^\top\sum_{i=1}^r(\mathbf{u}_i^\top\tilde{\boldsymbol{\epsilon}})\,\mathbf{u}_i && (4.39)\\
&= \sum_{i=1}^r(\mathbf{u}_i^\top\tilde{\boldsymbol{\epsilon}})^2. && (4.40)
\end{aligned}$$

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

Proposition 4.2.2 describes the distribution of $f^\star$ for affine measurement functions corrupted by additive Gaussian noise (Assumption 4.2.1). Unfortunately, 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 $\mathbf{x} = \mathbf{x}^\circ$, i.e.,

$$\mathbf{h}(\mathbf{x}) \approx \mathbf{h}(\mathbf{x}^\circ) + \mathbf{H}_\circ(\mathbf{x} - \mathbf{x}^\circ) \tag{4.41}$$

$$= \mathbf{H}_\circ \mathbf{x} + \underbrace{\mathbf{h}(\mathbf{x}^\circ) - \mathbf{H}_\circ \mathbf{x}^\circ}_{\mathbf{c}_\circ} \tag{4.42}$$

in which $\mathbf{H}_\circ$ is the Jacobian matrix of $\mathbf{h}(\mathbf{x})$ evaluated at the true value of $\mathbf{x}$, i.e.,

$$\mathbf{H}_\circ \triangleq \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}^\circ}. \tag{4.43}$$

Therefore, if the linearization error $\|\mathbf{h}(\mathbf{x}^\star) - \mathbf{H}_\circ \mathbf{x}^\star - \mathbf{c}_\circ\|$ is negligible, we can still rely on Proposition 4.2.2 (as an approximation) for nonlinear measurement functions. Note that this linearization error depends on the estimation error $\|\mathbf{x}^\star - \mathbf{x}^\circ\|$. Estimation error will be small when the measurements are precise (e.g., when $\mathrm{diag}(\boldsymbol{\Sigma})$ is "small") and/or when there are a sufficiently large number of measurements ($m \gg n$). Thus in such cases, it is reasonable to use Proposition 4.2.2 with caution even for nonlinear measurement functions. According to Proposition 4.2.1 and Proposition 4.2.2, the distribution of f° and $f^\star$ ultimately depends on the size of $\mathbf{z}$ and $\mathbf{x}$. For example, in 2D pose-graphs we have $\nu^\circ = 3m$ and $\nu^\star = 3(m-n)$. $\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma})}[\gamma]$ can be approximated by,

$$\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma})}[\gamma] \approx \frac{\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma})}[f^\star]}{\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma})}[f^\circ]} \tag{4.44}$$

$$\approx \frac{\nu^\star}{\nu^\circ} \tag{4.45}$$

$$= 1 - \frac{n}{m}. \tag{4.46}$$

This naive approximation in (4.44) can be justified by the first-order Taylor expansion of γ (i.e., ratio) at $f^\star = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma})}[f^\star] \approx \nu^\star$ and $f^\circ = \mathbb{E}_{\boldsymbol{\epsilon}\sim(\mathbf{0},\boldsymbol{\Sigma})}[f^\circ] = \nu^\circ$. Note that in general, approximating the expected value of a ratio by the ratio of expected values does not result in a reasonable estimate. Nevertheless, as we will see shortly in Figure 4.3, this approximation is good enough for explaining

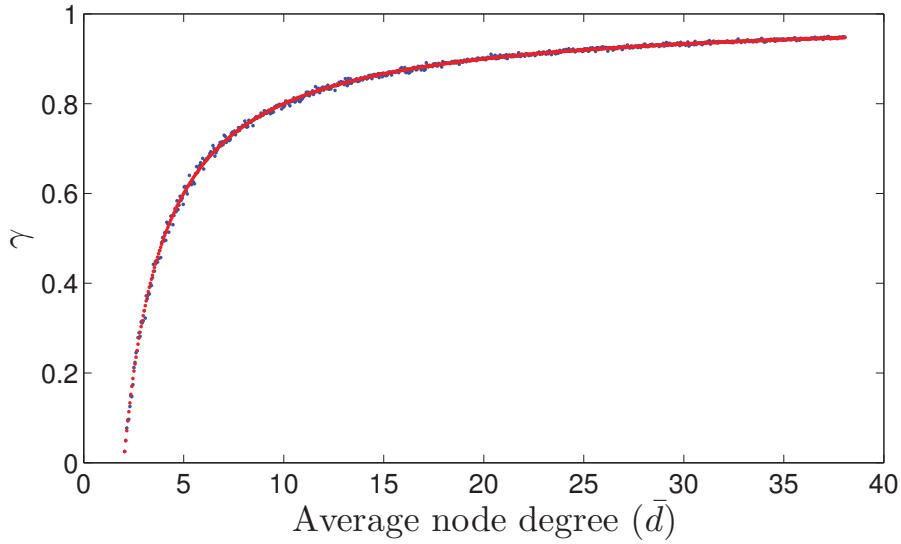


Figure 4.3: Average of γ in 50 Monte Carlo simulations for different average node degrees (blue). Estimated value of $\mathbb{E}[\gamma] \approx 1 - \frac{2}{\bar{d}}$ (red).

the aforementioned empirical observations.

Using the handshaking lemma (Lemma A.4.1) we can express (4.46) in terms of the average degree of graph; i.e.,

$$\mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})}[\gamma] \approx 1 - 2/\bar{d}\,. \tag{4.47}$$

Equation (4.47) 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 (4.47), as the average degree increases, γ in expectation approaches 1, which indicates an accurate maximum likelihood estimate. The red points in Figure 4.3 are drawn according to (4.47). According to this figure, (4.47) is consistent with our and Olson and Kaess [123]'s empirical observations regarding the average value of γ (blue points in Figure 4.3).

As noted before, according to Lemma A.4.1 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 variables (e.g., fixed number of poses), maximizing the average degree is equivalent to merely gathering as many measurements as possible. In other words, the average degree, as a connectivity measure, is insensitive to subtle topological differences between graphs with the same number of edges (per vertices). Hence, although (4.47) 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 measure-

ments (or even different measurement precisions). Nevertheless, (4.47) is still an interesting result as it provides insights into how making new observations ultimately leads to a more precise estimate.

4.2.2 Algebraic Connectivity $\leftrightarrow$ Worst-Case Error Variance

Now we present another result on how in Diff-Nets and Compass-SLAM, a "well-connected" graph is necessary for achieving reliable estimates. The following remark sets the stage for our main result.

Remark 8. *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 $\mathbf{u} \in \mathbb{R}^n$:*

$$\begin{aligned} \lambda_{\max} &\triangleq \lambda_n(\mathbb{C}\mathrm{ov}[\mathbf{e}]) && (4.48)\\ &= \max_{\|\mathbf{u}\|=1} \mathbf{u}^\top \mathbb{C}\mathrm{ov}[\mathbf{e}]\, \mathbf{u} && (4.49)\\ &= \max_{\|\mathbf{u}\|=1} \mathbf{u}^\top \mathbb{E}\big[\, \mathbf{e}\, \mathbf{e}^\top \big]\, \mathbf{u} && (4.50)\\ &= \max_{\|\mathbf{u}\|=1} \mathbb{E}\big[(\mathbf{u}^\top \mathbf{e})^2\big]. && (4.51) \end{aligned}$$

   Note that (4.51) *is the worst case variance of error among all unit directions.*

2. *Geometrically speaking, if $\mathbb{C}\mathrm{ov}[\mathbf{e}]$ is the covariance matrix of a Gaussian distribution, the "hyper-diameter" of the uncertainty hyper-ellipsoid is specified by $\sqrt{\lambda_{\max}}$.*

In the optimal experimental design literature, the design that minimizes $\lambda_{\max}$ of the estimation error covariance matrix is known as the E-optimal (extreme or eigenvalue) design [130]. As introduced in Section A.4, for any simple connected undirected graph $\mathcal{G}$, the second-largest eigenvalue of the (weighted) Laplacian matrix $\mathbf{L}_\circ$, i.e., $\lambda_2(\mathbf{L}_\circ) > 0$, is known as the *algebraic connectivity* of $\mathcal{G}$. Theorem 4.2.3 reveals the connection between the algebraic connectivity of a graph and the worst case estimation error variance—as a measure of estimation reliability—in Diff-Net and Compass-SLAM.

Theorem 4.2.3 (Algebraic Connectivity and Worst Case Error Variance)**.** *Let $\mathbb{C}\mathrm{ov}[\mathbf{x}^\star]$ be the estimation error covariance matrix of the maximum likelihood estimator in Diff-Net and Compass-SLAM. Let $\mathbf{L}_{\circ w}$ be the corresponding weighted Laplacian matrix. Then we have,*

$$\lambda_{\max}(\mathbb{C}\mathrm{ov}[\mathbf{x}^\star]) \geq \lambda_2^{-1}(\mathbf{L}_{\circ w}). \tag{4.52}$$

Proof. Recall that the Fisher information matrix in the d-dimensional Diff-Net and 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 [59, Theorem 9.1.1], $0 < \lambda_1(\mathbf{L}) \leq \lambda_2(\mathbf{L}_{\circ w})$. Now note that,

$$\lambda_{\max}(\mathbb{C}\mathrm{ov}[\mathbf{x}^\star]) = \lambda_1(\mathbb{I})^{-1} \tag{4.53}$$

$$= \lambda_1(\mathbf{L}_w \otimes \mathbf{I}_d)^{-1} \tag{4.54}$$

$$= \lambda_1(\mathbf{L}_w)^{-1} \tag{4.55}$$

$$\geq \lambda_2(\mathbf{L}_{\circ w})^{-1}. \tag{4.56}$$

□

According to Theorem 4.2.3, a necessary condition for having a sufficiently small worst-case estimation error variance in Diff-Net and Compass-SLAM is to have a sufficiently large algebraic connectivity. Note that $\mathbf{L}_w$ also depends on the edge weights. Therefore, as expected, scaling the information content (precision) of every edge by a factor of α, scales the (weighted) algebraic connectivity by the same factor (e.g., $\alpha\lambda_2$ where λ_2 is the original algebraic connectivity). Finally, recall that in linear-Gaussian problems, the Cramér-Rao bound is achievable. Hence, (4.52) can also be interpreted as a bound on the achievable E-criterion.

4.2.3 Tree Connectivity ↔ Volume of Uncertainty Ellipsoids

In this section, we demonstrate that the weighted number of spanning trees—as a measure of graph connectivity, see Section A.4—has a significant impact on the determinant of the estimation error covariance of the maximum likelihood estimator in several EoG problems, including SLAM. The D-optimality criterion [130] (D-criterion for short) can be interpreted as a scalar measure of the uncertainty encoded in a covariance matrix. For instance, in multivariate Gaussian distributions, the square root of the determinant of the covariance matrix is proportional to the hyper-volume of the confidence hyper-ellipsoids. Moreover, from an information-theoretic standpoint, the log-determinant of the covariance matrix of a Gaussian distribution is proportional to its differential entropy up to an additive constant.

Before presenting our main results, we point out few numerical tricks that are essential 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, instead of directly working with the weighted number of spanning trees $t(\mathcal{G}) = \det \mathbf{L}_w$ (Theorem A.4.6), we often use the *weighted tree-connectivity* as defined below.

Definition 4.2.1 (Tree-Connectivity)**.** The *tree-connectivity* of graph $\mathcal{G}$ is formally defined as

$$\tau(\mathcal{G}) \triangleq \begin{cases} \log t(\mathcal{G}) & \text{if } \mathcal{G} \text{ is connected,} \\ 0 & \text{otherwise.} \end{cases} \tag{4.57}$$

Similarly, for weighted graphs weighted by a positive weight function $w : \mathcal{E}(\mathcal{G}) \to \mathbb{R}_{>0}$, the *weighted tree-connectivity* is defined as

$$\tau_w(\mathcal{G}) \triangleq \begin{cases} \log t_w(\mathcal{G}) & \text{if } \mathcal{G} \text{ is connected,} \\ 0 & \text{otherwise.} \end{cases} \tag{4.58}$$

In SLAM and many other real-world EoGs, the Laplacian matrix and, consequently, the Fisher information matrix are sparse. Algorithm 4 outlines an efficient procedure for computing the log-determinant of any sparse positive-definite matrix. It is important to note that, unlike the Fisher information matrix, the covariance matrix in SLAM is generally dense. Therefore, to exploit the sparse structure of the problem, we always compute $\log\det \mathbb{C}\mathrm{ov}[\mathbf{x}^\star]$ indirectly via $\log\det \mathbb{I}(\mathbf{x})$:

$$\log\det \mathbb{C}\mathrm{ov}[\mathbf{x}^\star] \approx \log\det \mathbb{I}(\mathbf{x}^\star)^{-1} \tag{4.59}$$

$$= -\log\det \mathbb{I}(\mathbf{x}^\star). \tag{4.60}$$

For dense graphs and in the worst case, computing the $\log\det$ requires $O(n^3)$ time—where n is the number of vertices—while in many practical sparse scenarios that arise in the context of robotics and sensor networks, Algorithm 4 performs much faster given a sufficiently good fill-reducing permutation. Now we are ready to present our main results.

Theorem 4.2.4 (D-optimal Diff-Net and Compass-SLAM)**.** *In the d-dimensional Diff-Net and Compass-*

Algorithm 4 $\log\det(\mathbf{S})$ for a sparse symmetric $\mathbf{S} \succ \mathbf{0}$

```
function LogDet(S)
    // Choose a fill-reducing permutation heuristic P
    P ← COLAMD(S)                    ▷ e.g., column approximate minimum degree
    // Compute the sparse Cholesky factor C s.t. S = CC^⊤
    C ← SparseCholesky(PSP^⊤)
    return 2 ∑_i log C_{i,i}
end function
```

SLAM (i.e., $\mathbf{x}_i \in \mathbb{R}^d$) we have,

$$\log\det(\mathbb{C}\mathrm{ov}[\mathbf{x}^\star]) = -d\,\tau_w(\mathcal{G}). \tag{4.61}$$

Here the weights are equal to the precision of the corresponding measurements (see Section 4.1).

Proof. Based on Propositions 4.1.1 and 4.1.2 and the weighted matrix-tree theorem (Theorem A.4.6) we have,

$$\begin{aligned}
\log\det(\mathbb{C}\mathrm{ov}[\mathbf{x}^\star]) &= -\log\det(\mathbb{I}) && (4.62)\\
&= -\log\det(\mathbf{L}_w \otimes \mathbf{I}_d) && (4.63)\\
&= -d\,\tau_w(\mathcal{G}). && (4.64)
\end{aligned}$$

□

Theorem 4.2.4 ensures that in Diff-Net and Compass-SLAM, under the specified assumptions, the following (equivalent) objectives are all achieved by maximizing the weighted number of spanning trees in the underlying graph:

1. Minimizing the differential entropy of $\mathbf{x}^\star$.
2. Minimizing the hyper-volume of uncertainty hyper-ellipsoid.
3. Maximizing the D-criterion.

The graph with the maximum (weighted) number of spanning trees among a family of graphs has been referred to as *t-optimal*. Theorem 4.2.4 states that D-optimality and t-optimality are equivalent under the aforementioned assumptions. Now we extend Theorem 4.2.4 to the SLAM problem defined in Section 4.1.3. The following lemma readily follows from applying Schur's determinant formula (Lemma A.1.1) and matrix-tree theorem (Theorem A.4.6) on (4.28).

Lemma 4.2.1. *For the SLAM problem defined in Section 4.1.3 we have,*

$$\log\det \mathbb{I}(\mathbf{x}) = 2\cdot\tau_{w_p}(\mathcal{G}) + \log\det\Big(\mathbf{L}_{w_\theta} + \boldsymbol{\Delta}_{w_p}^\top \mathbf{P}_{w_p}^\perp \boldsymbol{\Delta}_{w_p}\Big), \tag{4.65}$$

in which $\mathbf{P}_{w_p}^\perp$ *is defined as,*

$$\mathbf{P}_{w_p}^\perp \triangleq \mathbf{I} - \boldsymbol{\Gamma}^\top\Big(\mathbf{A}_{w_p}^\top \mathbf{L}_{w_p}^{-1}\mathbf{A}_{w_p}\otimes \mathbf{I}_2\Big)\boldsymbol{\Gamma}. \tag{4.66}$$

Now we use Lemma 4.66 to bound $\log\det\mathbb{I}(\mathbf{x})$.

Proposition 4.2.5. *For the SLAM problem defined in Section 4.1.3 we have*

$$\mathcal{L} \leq \log\det\mathbb{I}(\mathbf{x}) \leq \mathcal{U}, \tag{4.67}$$

in which $\mathcal{L}$ *and* $\mathcal{U}$ *are defined below,*

$$\mathcal{L} \triangleq 2\cdot\tau_{w_p}(\mathcal{G}) + \tau_{w_\theta}(\mathcal{G}) \tag{4.68}$$

$$\mathcal{U} \triangleq 2\cdot\tau_{w_p}(\mathcal{G}) + \sum_{i=1}^{n}\log\Big(\lambda_i(\mathbf{L}_{w_\theta}) + \|\boldsymbol{\Delta}_{w_p}^\top\boldsymbol{\Delta}_{w_p}\|_\infty\Big). \tag{4.69}$$

Finally, Theorem 4.2.6 shows that when $\|\boldsymbol{\Delta}_{w_p}^\top\boldsymbol{\Delta}_{w_p}\|_\infty \to 0$, the value of $\log\det\mathbb{I}(\mathbf{x})$ only depends on the weighted tree-connectivity of the underlying graph.

Theorem 4.2.6. *Define* $\delta \triangleq \|\boldsymbol{\Delta}_{w_p}^\top\boldsymbol{\Delta}_{w_p}\|_\infty$. *For the SLAM problem defined in Section 4.1.3 we have,*

$$\lim_{\delta\to 0^+}\log\det\mathbb{I}(\mathbf{x}) = 2\cdot\tau_{w_p}(\mathcal{G}) + \tau_{w_\theta}(\mathcal{G}). \tag{4.70}$$

From (4.29) we have,

$$\delta = \max\left\{d_i : d_i = \sum_{j\in\mathcal{S}(i)} w_p(i,j)\|\mathbf{p}_i - \mathbf{p}_j\|^2 \quad \forall i\in[n]\right\}. \tag{4.71}$$

This parameter depends on the outdegree of the nodes through $|\mathcal{S}(i)|$, the sensing range through $\|\mathbf{p}_i - \mathbf{p}_j\|^2$, and the precision of the translational measurements through w_p. In the following corollary, we consider a special case in which rotational and translation measurements are corrupted by isotropic noise.

Corollary 4.2.7. *Suppose the covariance matrix for rotational and translational measurements is isotropic,*

i.e., $\mathbf{\Sigma}_p = \sigma_p^2 \cdot \mathbf{I}$ *and* $\mathbf{\Sigma}_\theta = \sigma_\theta^2 \cdot \mathbf{I}$*. Let* $\mathbb{I}(\mathbf{x}|\mathcal{T})$ *be the Fisher information matrix associated to an arbitrary spanning tree, e.g., the odometry subgraph. Then,*

$$\lim_{\delta \to 0} \log\det \mathbb{I}(\mathbf{x}) - \log\det \mathbb{I}(\mathbf{x}|\mathcal{T}) = 3 \cdot \tau(\mathcal{G}). \tag{4.72}$$

A natural choice for $\mathcal{T}$ is the odometry spanning tree $\mathcal{T}_{\text{odo}}$. In this case, $\Delta_{\text{inf}}(\mathcal{G}) \triangleq \log\det \mathbb{I}(\mathbf{x}) - \log\det \mathbb{I}(\mathbf{x}|\mathcal{T}_{\text{odo}})$ can be interpreted as the information gained by closing loops as compared with the dead reckoning scenario. According to Corollary 4.2.7, $\Delta_{\text{inf}}(\mathcal{G})$ will be proportional to the tree-connectivity of graph when δ is sufficiently small. The following corollary directly follows from Corollary 4.2.7 and Cayley's formula (see Theorem A.4.5).

Corollary 4.2.8. *In the SLAM problem defined in Section 4.1.3 with isotropic noise we have,*

$$\lim_{\delta \to 0^+} \Delta_{\text{inf}}(\mathcal{K}_n) = 3 \cdot (n-2) \cdot \log(n). \tag{4.73}$$

Theorems 4.2.4 and 4.2.6 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. One way of comparing the tree-connectivity of graphs with a different number of vertices is to somehow normalize the absolute tree-connectivity by the number of spanning trees.

Definition 4.2.2 (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 approach, 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 A.4.5).

Corollary 4.2.9. *Let* $\mathcal{G}$ *be a simple undirected graph with* n *vertices. The following statements hold regarding the normalized tree-connectivity of* $\mathcal{G}$*.*

(i)

$$\bar{\tau}(\mathcal{G}) = \frac{\tau(\mathcal{G})}{(n-2)\log(n)}. \tag{4.74}$$

(ii) $0 \leq \bar{\tau}(\mathcal{G}) \leq 1$.

(iii) $\bar{\tau}(\mathcal{G}) = 0$ *if and only if* $\mathcal{G}$ *is not connected.*

(iv) $\bar{\tau}(\mathcal{G}) = 1$ *if and only if* $\mathcal{G}$ *is the complete graph.*

Corollary 4.2.10 follows from Corollary 4.2.7.

Corollary 4.2.10. *In the SLAM problem defined in Section 4.1.3 with isotropic noise we have,*

$$\lim_{\delta \to 0^+} \frac{\Delta_{\text{inf}}(\mathcal{G})}{\Delta_{\text{inf}}(\mathcal{K})} = \bar{\tau}(\mathcal{G}). \tag{4.75}$$

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. However, this would require making explicit assumptions regarding the weights of the missing edges.

4.3 Experimental Results

We conducted several experiments on both real and synthetic SLAM datasets. These experiments are specifically designed to:

1. Evaluate the tree-connectivity of some of the publicly available real and synthetic SLAM benchmarks.
2. Empirically validate Theorem 4.2.6.
3. Assess the sensitivity of the asymptotic result provided by Theorem 4.2.6 with respect to the value of δ.

To test Theorem 4.2.6 numerically, we use the relative error (RE) defined as

$$\text{RE} \triangleq \left| \frac{\log\det \mathbb{I}(\mathbf{x}) - \mathcal{L}}{\log\det \mathbb{I}(\mathbf{x})} \right|, \tag{4.76}$$

Table 4.1: 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	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

where

$$\mathcal{L} = \lim_{\delta \to 0^+} \log\det \mathbb{I}(\mathbf{x}) \triangleq 2\tau_{w_p}(\mathcal{G}) + \tau_{w_\theta}(\mathcal{G}). \tag{4.77}$$

The datasets used in this section 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 4.2.6 and isotropic for Corollary 4.2.7). Note that RE, through the Fisher information matrix, depends on $\mathbf{x}$. The inverse of the Fisher information matrix evaluated at the ground truth results in the Cramér-Rao lower bound (see Appendix A). Moreover, the covariance matrix of the maximum likelihood estimator is usually approximated by computing the inverse of the Fisher information matrix at the maximum likelihood estimate. Among the datasets used in this section, Manhattan [121] (a synthetic 2D pose-graph dataset) is the only one for which the ground truth is publicly available. Therefore, RE in other datasets is inevitably 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. [98].

Results

The normalized tree-connectivity for several publicly available datasets is shown in Table 4.1. The entries in Table 4.1 are sorted (in descending order) based on the normalized tree-connectivity $\bar{\tau}(\mathcal{G})$. First, note that the relative error (RE) is typically small, except in the case of Lincoln dataset. A small RE indicates that $\log\det \mathbb{I}(\mathbf{x})$ is already close to its asymptotic value predicted by Theorem 4.2.6. Hence, this empirical observation suggests that in these datasets, Theorem 4.2.6 is not sensitive to the value of δ (e.g., in Manhattan $\delta \approx 1296.91$). In such cases, the log-determinant of the Fisher

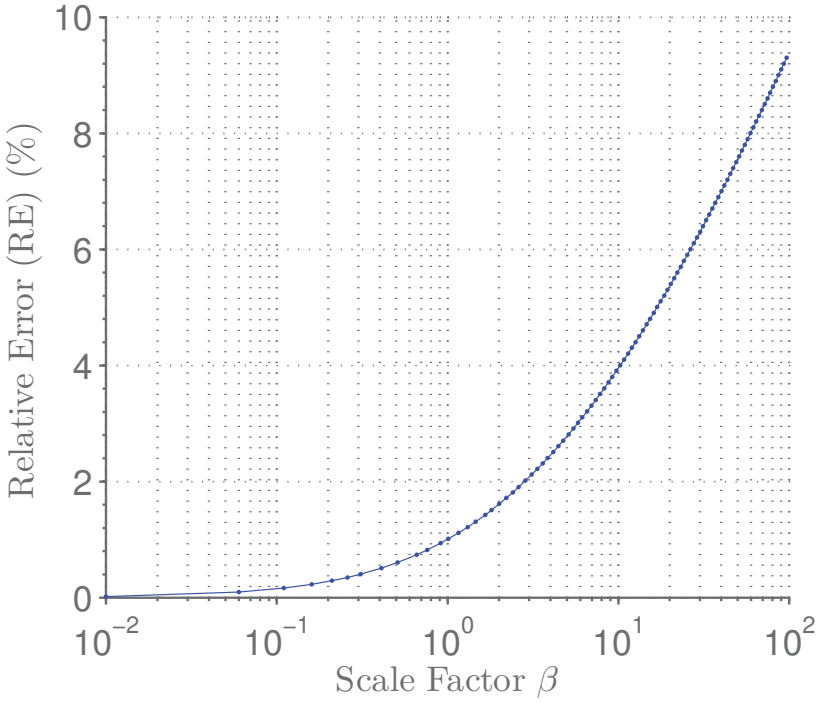


Figure 4.4: RE evaluated at the ground truth as a function of the scale parameter β for Manhattan. Here $\delta = \beta\delta_{\text{orig}}$ in which $\delta_{\text{orig}} \approx 1296.91$ is the value of δ in Manhattan dataset (see Table 4.1). This can be done by scaling either σ_p^2 or $\mathbf{p}$. Note the logarithmic scale of the horizontal axis.

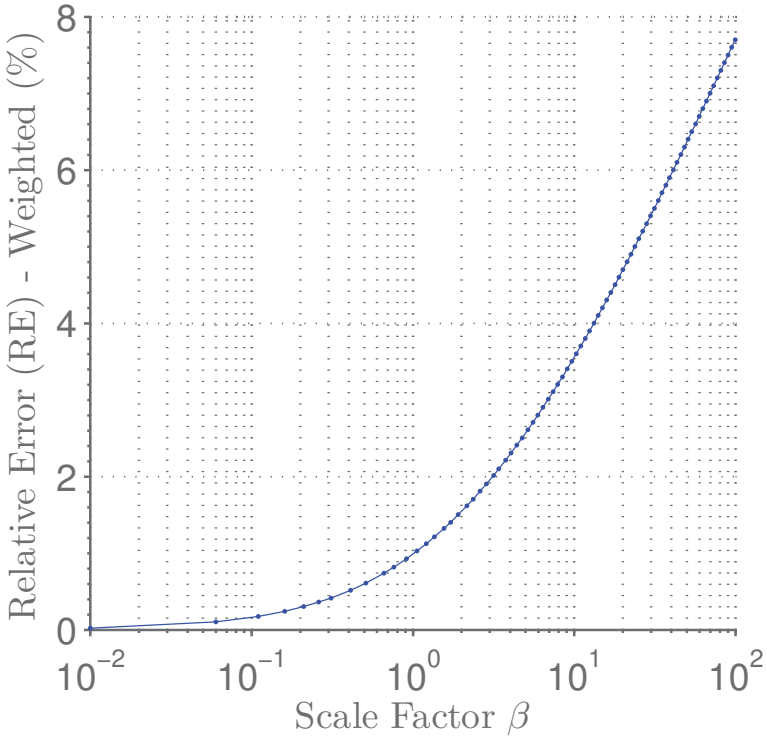
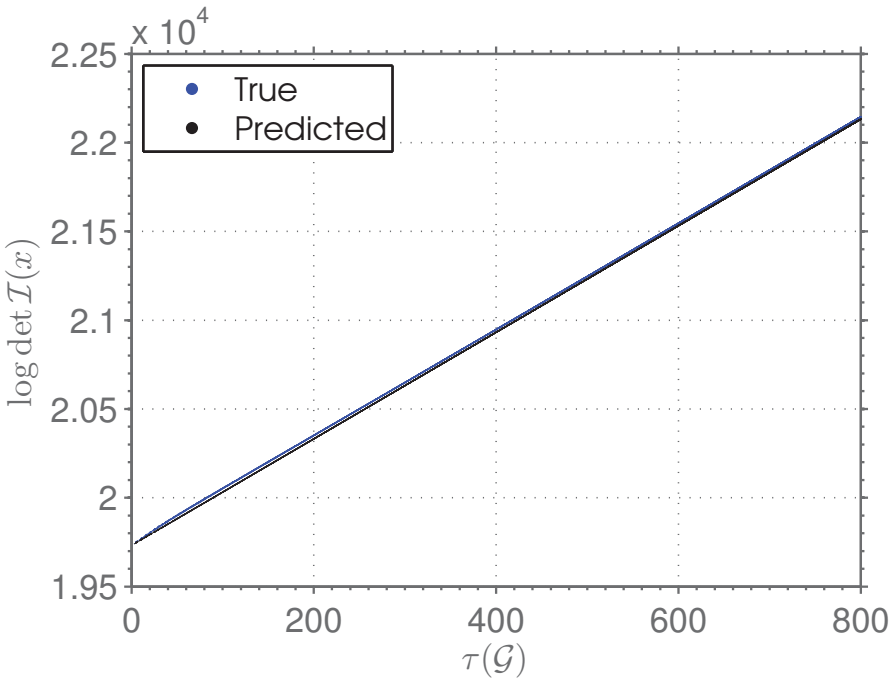
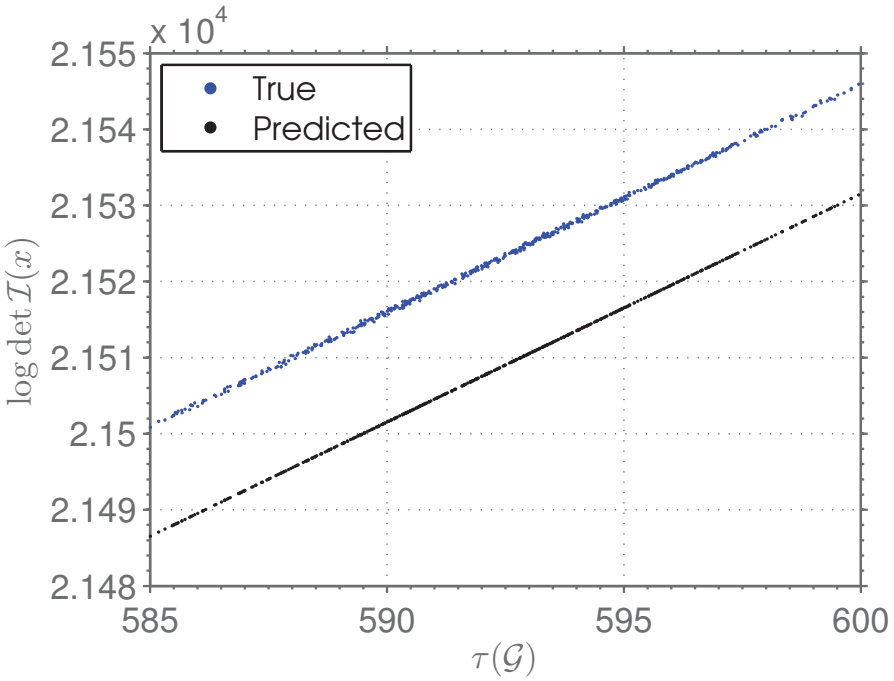


Figure 4.5: RE evaluated at the ground truth as a function of scale parameter β for Manhattan. Here $\delta = \beta\delta_0$ in which $\delta_0 \approx 6.21 \times 10^4$. In this experiment, the edges have different noise variances. Note the logarithmic scale of the horizontal axis.



(a) Overview.



(b) Zoomed region.

Figure 4.6: $\log\det \mathbb{I}(\mathbf{x})$ 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 maximum likelihood estimate of the original dataset. The prediction is based on the limit value provided by Theorem 4.2.6.

information matrix 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 Gauss-Newton is subject to local minima. The large RE in the case of Lincoln dataset (highlighted in red) is partially

due to the fact that Gauss-Newton has failed to converge to the true maximum likelihood estimate. By contrast, estimating the D-criterion based on Theorem 4.2.6 is naturally robust to such convergence errors.

Notice that in Table 4.1, 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 [123]—is too simple to capture the structural differences between two graphs with the same number of measurements per variables; see Lemma A.4.1.

Our next experiment is based on the Manhattan dataset. The Fisher information matrix in this case is computed at the ground truth. Figure 4.4 depicts 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 4.4 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 (see Table 4.1). As illustrated in Figure 4.4, the log-determinant of the Fisher information matrix approaches the limit value predicted by Theorem 4.2.6 as $\delta \to 0^+$ (see also the special case in Corollary 4.2.7).

We 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 4.5, we scale δ according to $\beta\delta_{\text{orig}}$ (this time $\delta_{\text{orig}} \approx 6.21 \times 10^4$ due to random perturbations). It is clear based on our results that when δ is relatively small, Theorem 4.2.6 provides a reasonable estimate for the log-determinant of the Fisher information matrix.

Finally, Figure 4.6 shows $\log\det \mathbb{I}(\mathbf{x})$ 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 $\mathcal{L}$. Figure 4.6 indicates that in this case, tree-connectivity almost entirely characterizes the D-criterion as predicted by our theoretical results.

4.4 Summary

In this chapter, we studied the impact of several graph-connectivity measures in Diff-Nets, Compass-SLAM and 2D pose-graph SLAM on the estimation error covariance matrix associated to the maximum likelihood estimator. In particular, we presented a theoretical justification for the empirical

observations made in [123] about the average degree in the graph. We established a connection between the E-optimality criterion in Diff-Nets and Compass-SLAM and the algebraic connectivity of the underlying graph. Finally, we proved that the weighted number of spanning trees is closely related to the D-optimality criterion. Our work demonstrates that the graphical representation of SLAM offers a compact, yet rich representation that can be used to predict the quality of estimation without knowing specific details about the geometry of the scene (i.e., without the need for solving the original optimization problem).

CHAPTER 5

Synthesis of Near-Tree-Optimal Graphs

In Chapter 4, we demonstrated that the graphical structure of SLAM can be used to reason about the estimation error covariance without a detailed knowledge about the geometry of the scene (i.e., without solving the underlying estimation/optimization problem). In particular, we proved that the weighted number of spanning trees is closely related to the determinant of the Fisher information matrix in several EoG problems, including SLAM. This result suggests that, under certain conditions, D-optimal SLAM problems can be constructed by designing SLAM problems whose graphical representations are t-optimal. In this chapter, we tackle the combinatorial optimization problem of designing graphs with the maximum weighted number of spanning trees under several types of constraints. The problem of designing *sparse graphs* with the maximum weighted number of spanning trees also arises in several other contexts across different domains in science and engineering. We investigate some of these applications in Section 5.7.

Outline

In the next section, we point out a few basic results needed for formulating the problem of t-optimal graph synthesis. Section 5.2 presents a formal definition of this problem, motivated by the measurement selection problem in SLAM. In Section 5.3, we develop our approximation algorithms and provide an analysis. In Section 5.4, we extend our algorithms and their analyses to the dual problem. Matroid constraints are discussed in Section 5.5. In Section 5.6, we extend our original problem definition to the case of disconnected base graphs. Section 5.7 shows how our graph synthesis framework can be used in several applications, including the measurement selection problem in SLAM. Finally, in Section 5.8, we evaluate the performance of our algorithms on both synthetic and real graphs.

5.1 Maximizing Tree-Connectivity

We begin by simplifying our notation. 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 $|\mathcal{V}(\mathcal{G})| = |\mathcal{V}(\mathcal{H})|$. Now suppose $\mathbb{G} \subseteq \mathbb{G}_n$ and a positive weight function $w : \mathcal{E}(\mathcal{K}_n) \to \mathbb{R}_{>0}$ are given. In the most general case, we are interested in finding t-optimal graphs with respect to $\mathbb{G}$ and w, i.e.,

$$\underset{\mathcal{G}\in\mathbb{G}\subseteq\mathbb{G}_n}{\text{maximize}} \quad t_w(\mathcal{G}). \tag{5.1}$$

When all edges have equal weights,

$$w(u,v) = w(p,q), \qquad \forall u,v,p,q \in [n], \tag{5.2}$$

$t_w(\mathbb{G})$ in (5.1) 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 (5.1) are assumed to be given. Hence, the decision variables are in fact the 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, (5.1) can be rewritten as,

$$\underset{\mathcal{E}\in\mathbb{E}}{\text{maximize}} \quad t_w([n],\mathcal{E}). \tag{5.3}$$

Lemma 5.1.1. *The set of optimal solutions of* (5.1) *is invariant under scaling w by any constant $\alpha > 0$.*

Proof. 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, $t_w(\mathcal{G}) \geq t_w(\mathcal{H}) \Leftrightarrow t_{w_\alpha}(\mathcal{G}) \geq t_{w_\alpha}(\mathcal{H})$ for any $\mathcal{G}$ and $\mathcal{H}$ with n vertices. □

Define $w_\perp \triangleq \min\, w(u,v)$. If $w_\perp < 1$, according to Lemma 5.1.1 we can scale every weight by any $\alpha \geq w_\perp^{-1}$ without affecting the t-optimal topologies in (5.1). Therefore, without losing any generality we can assume the following.

Assumption 5.1.1. $w(u,v) \geq 1$ *for all vertices u and v.*

For clarity, let us define the weighted number of spanning trees and weighted tree-connectivity as functions of the edge set of graph.

Definition 5.1.1. For any $n \geq 2$ and $w : \mathcal{E}(\mathcal{K}_n) \to \mathbb{R}_{\geq 1}$ define,

$$\begin{aligned} t_{n,w} : \mathcal{E}(\mathcal{K}_n) &\to \mathbb{R}_{\geq 0} \\ \mathcal{E} &\mapsto t_w([n],\mathcal{E}). \end{aligned} \tag{5.4}$$

Similarly, we define

$$\begin{aligned} \tau_{n,w} : \mathcal{E}(\mathcal{K}_n) &\to \mathbb{R}_{\geq 0} \\ \mathcal{E} &\mapsto \tau_w([n],\mathcal{E}). \end{aligned} \tag{5.5}$$

Consequently, (5.3) can be rewritten as,

$$\underset{\mathcal{E}\in\mathbb{E}}{\text{maximize}} \quad t_{n,w}(\mathcal{E}), \tag{5.6}$$

or, equivalently, since $\log$ is monotonic,

$$\underset{\mathcal{E}\in\mathbb{E}}{\text{maximize}} \quad \tau_{n,w}(\mathcal{E}). \tag{5.7}$$

It will become clear soon why using $\tau_{n,w}$ is preferred over $t_{n,w}$. Notice that (5.3)-(5.7) represent the most general case of finding t-optimal graphs. 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,

$$\begin{aligned} \underset{\mathcal{E}\subseteq\mathcal{E}(\mathcal{K}_n)}{\text{maximize}} \quad & \tau_{n,w}(\mathcal{E}) \\ \text{subject to} \quad & |\mathcal{E}| = m. \end{aligned} \tag{5.8}$$

The cardinality constraint enforces the desired level of sparsity. In many cases, sparsity of the graph is often a crucial factor in determining the amount of resources needed for solving the problems that arise over graph structures and networks. Particularly in EoG problems such as SLAM, sparsity of the graph determines the computational cost of solving the resulting inference problem associated with ML/MAP estimation.

5.1.1 Characterizing t-Optimal Graphs in $\mathbb{G}_{n,m}$

The problem of characterizing graphs in $\mathbb{G}_{n,m}$ with the maximum number of spanning trees has remained open, and is solved only for special cases; e.g., for specific ranges of m (with respect to n),

such as when $n-1 \leq m \leq n+3$ (almost-tree graphs) and $\binom{n}{2} - n/2 \leq m \leq \binom{n}{2}$ (almost-complete graphs). Another major result is due to Cheng [29] who proved that the family of regular complete multipartite graphs are t-optimal among all graphs with the same number of vertices and edges. These results can be found in [29, 89, 126, 137, 153]. See also [13] for a recent survey of known results.

Unfortunately the span of these special cases is too narrow for the types of graphs that typically arise in SLAM and many other problems. Furthermore, in many problems, the $\mathbb{G}_{n,m}$ constraint alone is insufficient for characterizing the true set of "feasible" graphs and cannot capture implicit practical constraints. Finally, these results do not cover the case of weighted graphs or parallel edges which are essential for representing, e.g., EoG problems such as SLAM.

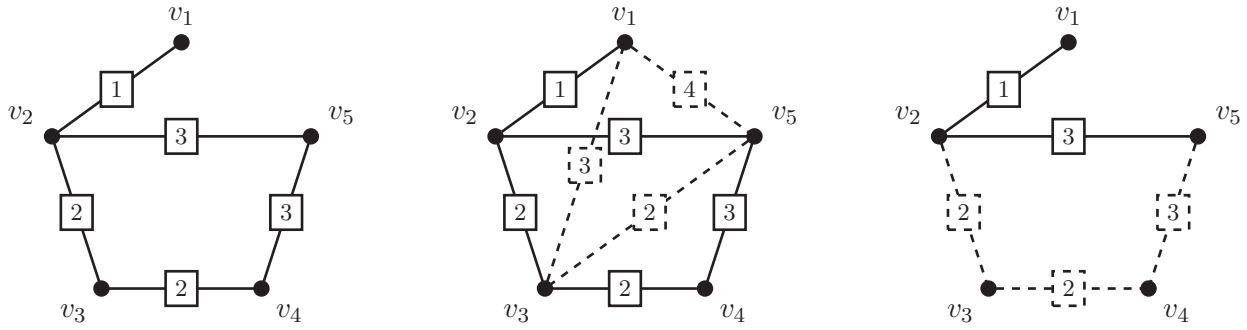
5.2 The Edge Selection Problem

Due to the drawbacks of the $\mathbb{G}_{n,m}$ constraint, in this section we study several special scenarios of (5.3) that emerge from designing reliable (i.e., near-D-optimal) EoG problems such as SLAM. An important special case of (5.3) is the following. Suppose a *connected* base graph and a positive weight function over the set of all possible edges are given. Informally speaking, the edge selection problem (ESP) is a combinatorial optimization problem whose goal is to find a t-optimal graph with respect to the set of graphs whose edge sets are at most k elements different from that of the base graph. A precise definition of ESP is given below.

Problem 5.2.1 (k-ESP)**.** Any instance of k-ESP is specified by the following:

- A set of $n \geq 2$ vertices $\mathcal{V} = [n]$,
- A planning horizon: $k \in \mathbb{N}$,
- A base graph: a *connected* graph $\mathcal{G}_{\text{init}} = (\mathcal{V}, \mathcal{E}_{\text{init}})$,
- A weight function: $w : \mathcal{E}(\mathcal{K}_n) \to \mathbb{R}_{\geq 1}$,
- A candidate set, i.e., a set of $c \geq k$ candidate edges that can be either $\mathcal{C}^+ \subseteq \mathcal{E}(\mathcal{K}_n) \setminus \mathcal{E}_{\text{init}}$ or $\mathcal{C}^- \subseteq \mathcal{E}_{\text{init}}$.

Two variants of k-ESP are defined below.



(a) $\mathcal{G}_{\text{init}}$. **(b)** k-ESP$^+$. **(c)** k-ESP$^-$.

Figure 5.1: Examples of k-ESP. The candidate edges are drawn with the dashed lines.

1. k-ESP$^+$:

$$\begin{aligned} &\underset{\mathcal{E}\subseteq\mathcal{C}^+}{\text{maximize}} && \tau_{n,w}(\mathcal{E}_{\text{init}} \cup \mathcal{E}) \\ &\text{subject to} && |\mathcal{E}| = k. \end{aligned} \tag{5.9}$$

2. k-ESP$^-$:

$$\begin{aligned} &\underset{\mathcal{E}\subseteq\mathcal{C}^-}{\text{maximize}} && \tau_{n,w}(\mathcal{E}_{\text{init}} \setminus \mathcal{E}) \\ &\text{subject to} && |\mathcal{E}| = k. \end{aligned} \tag{5.10}$$

Example 5.2.1 (k-ESP)**.** Figure 5.1 illustrates instances of k-ESP$^+$ and k-ESP$^-$. The set of candidate edges $\mathcal{C}^+ = \big\{\{1,3\},\{3,5\},\{1,5\}\big\}$ and $\mathcal{C}^- = \big\{\{2,3\},\{3,4\},\{4,5\}\big\} \subseteq \mathcal{E}_{\text{init}}$ are shown with the dashed edges. The number written on each edge is the weight assigned to that edge.

The notion of "base graph" ($\mathcal{G}_{\text{init}}$) in the definition of k-ESP can be seen as a design constraint: in k-ESP$^+$, the base graph must be a spanning subgraph of the optimal graph, and conversely in k-ESP$^-$, the optimal graph has to be a spanning subgraph of the base graph. At first glance, the $\mathbb{G}_{n,m}$ may seem as a special case of k-ESP$^+$ with $\mathcal{E}_{\text{init}} = \varnothing$ and $\mathcal{C}^+ = \binom{[n]}{2}$. However, strictly speaking, any problem with $\mathcal{E}_{\text{init}} = \varnothing$ is not a valid k-ESP$^+$ since in the definition of k-ESP$^+$ it is explicitly assumed that $\mathcal{G}_{\text{init}}$ has to be connected. In some applications, $\mathcal{G}_{\text{init}}$ is naturally connected and therefore we do not lose any practical generality by having this assumption; e.g., in pose-graph SLAM, the odometry subgraph (i.e., the path graph with n vertices and weighted edges) is a natural choice for $\mathcal{G}_{\text{init}}$. Nevertheless, for technical reasons, the algorithms designed in this chapter explicitly assume that $\mathcal{G}_{\text{init}}$ is connected. We will revisit this assumption in Section 5.6 and discuss how it can be relaxed in practice.

In general, any instance of k-ESP$^+$ can be posed as an instance of k'-ESP$^-$ (with different pa-

Algorithm 5 Optimal Edge (1-ESP$^+$)

```
1: function NextBestEdge(𝒞⁺,C)          ▷ 𝒞⁺: Candidate Set, C: Cholesky factor of L_init
2:     m ← 0                              ▷ Maximum value
3:     for all e ∈ 𝒞⁺ do                  ▷ Parallelizable loop
4:         w_e ← w(e)
5:         Δ_e ← Reff(e,C)
6:         if w_e Δ_e > m then
7:             e* ← e
8:             m ← w_e Δ_e
9:         end if
10:    end for
11:    return e*
12: end function
```

Algorithm 6 Effective Resistance

```
1: function Reff(e_uv,C)                 ▷ Effective Resistance
2:     // column of the reduced incidence matrix
3:     a_uv ← e_u − e_v
4:     // solve C x_uv = a_uv
5:     x_uv ← ForwardSolver(C,a_uv)      ▷ Lower triangular
6:     Δ_uv ← ‖x_uv‖²
7:     return Δ_uv
8: end function
```

rameters), and vice versa. For instance, selecting k edges among c new candidates can be done by (i) adding all c candidates to the graph, and (ii) pruning the $k' \triangleq c-k$ unwanted edges—i.e., an instance of k'-ESP$^-$. Hence, in the rest of this chapter we focus mainly on the k-ESP$^+$ problem.

5.2.1 1-ESP$^+$

In this section, we discuss 1-ESP$^+$ as the simplest instance of k-ESP$^+$. As it 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.

The optimal edge in 1-ESP$^+$ can be found by examining every available candidate edge in $\mathcal{C}^+$, and choosing the one that maximizes the weighted number of spanning trees of the resulting graph $([n],\mathcal{E}_{\text{init}} \cup \{e\})$. Hence, finding the 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, computing tree-connectivity can be done in $O(n^3)$ operations using the Cholesky decomposition of the reduced weighted Laplacian matrix (similar to Algorithm 4). Hence, the total time complexity in this case is $O(cn^3)$.

1-ESP$^+$ can be solved in $O(n^3+cn^2)$ using Theorem A.4.7. According to this theorem, the solution

of 1-ESP$^+$ is given by,

$$e^\star = \arg\max_{e \in \mathcal{C}^+} \; w(e)\Delta_e, \tag{5.11}$$

where $\Delta_e \triangleq \mathbf{a}_e^\top \mathbf{L}_{\text{init}}^{-1} \mathbf{a}_e$.

Remark 9 (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 gives a metric on graphs; i.e., among other properties, it satisfies the triangle inequality. Therefore, we can interpret Δ_e as the* distance *between the two endpoints of e in the base graph (*resistance distance*). The effective resistance between u and v is equal to the resistance measured between these two vertices when the edges of the graph (electrical circuit) represent resistors with conductances equal to their weights (hence the name). This metric arises also in a few other applications; see [58].*

From this perspective, (5.11) can be interpreted as follows.

Corollary 5.2.1. *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.*

Note that if the Cholesky decomposition of $\mathbf{L}_{\text{init}}$ is available, we can compute $w(e)\,\Delta_e$ for all candidate edges in $O(c\,n^2)$ time. Therefore, the Cholesky decomposition of $\mathbf{L}_{\text{init}}$ needs to be computed only once, which takes $O(n^3)$ operations. Hence, as mentioned earlier, the total time in this case will be $O(n^3 + c\,n^2)$. Solving the 1-ESP$^+$ using this procedure will therefore be somewhat faster than the brute force algorithm outlined above, especially when c is large. This algorithm is summarized in Algorithm 5.

5.2.2 Exhaustive Search

Solving the general case of k-ESP$^+$ by exhaustive search requires examining $\binom{c}{k}$ graphs which is usually impractical. For example, for $c = 30$ and $k = 10$, exhaustive search has to compute more than 3×10^7 Cholesky decompositions. To the best of our knowledge, there is no known efficient algorithm for finding t-optimal graphs or solving the k-ESP$^+$ problem in polynomial time. Hence, in the following section we design efficient near-optimal approximation algorithms for this problem.

5.3 Approximation Algorithms for k-ESP$^+$

5.3.1 Greedy Algorithm

Algorithm and Complexity Analysis

The greedy algorithm finds an approximate solution for k-ESP$^+$ by decomposing it into a sequence of k 1-ESP$^+$ subproblems, each of which can be solved using the procedure described above. After solving each subproblem, the optimal edge is moved from the candidate set $\mathcal{C}^+$ to the base graph. The next 1-ESP$^+$ subproblem then is defined using the updated candidate set and base graph. If $\mathcal{G}_{\text{init}}$ is densely connected, a naive implementation of the greedy algorithm requires less than $O(k\,c\,n^3)$ operations.

As mentioned in Section 5.2.1, we can do better than $O(k\,c\,n^3)$ by using the result of Theorem A.4.7. 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 the Cholesky factor of the base graph in the previous round. This operation can be done in $O(n^2)$ time. Therefore, in total we need $O(n^3 + k\,c\,n^2)$ time to run the greedy algorithm on an instance of k-ESP$^+$. This procedure is described in Algorithm 7.

Performance Analysis

In this section, we analyze the performance of the greedy algorithm described in Algorithm 7.

Definition 5.3.1 (Tree-Connectivity Gain)**.** Suppose an instance of k-ESP$^+$ is given with some $\mathcal{G}_{\text{init}}$, $\mathcal{C}^+$ and w. The tree-connectivity gain is defined as,

$$\Phi_w : \mathcal{E} \mapsto \tau_{n,w}(\mathcal{E}_{\text{init}} \cup \mathcal{E}) - \tau_{n,w}(\mathcal{E}_{\text{init}}). \tag{5.12}$$

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

Φ_w is a set function that takes as input a subset of the candidate edges $\mathcal{E} \subseteq \mathcal{C}^+$, and returns the marginal increase in weighted tree-connectivity after adding the edges in $\mathcal{E}$ to the base graph $\mathcal{G}_{\text{init}}$. Recall that in k-ESP$^+$, it is assumed that $\mathcal{G}_{\text{init}}$ is connected and $w(e) \geq 1$ for all $e \in \binom{[n]}{2}$. Strictly speaking, according to the above definition, Φ_w is parametrised by $\mathcal{G}_{\text{init}} = ([n],\mathcal{E}_{\text{init}})$ and w. Nonetheless, to simplify our notation, we may write $\Phi_w(\mathcal{E})$ instead of $\Phi_w(\mathcal{E}\,;\mathcal{G}_{\text{init}},w)$ whenever $\mathcal{G}_{\text{init}}$

Algorithm 7 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**
    $e^\star_{uv} \leftarrow$ NextBestEdge($\mathcal{C}^+ \setminus \mathcal{E}$,$\mathbf{C}$)
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{e^\star_{uv}\}$
    // column of the reduced incidence matrix
    $\mathbf{a}_{uv} \leftarrow \mathbf{e}_u - \mathbf{e}_v$
    $\mathbf{L} \leftarrow \mathbf{L} + w(e^\star_{uv})\mathbf{a}_{uv}\mathbf{a}_{uv}^\top$
    $\mathbf{C} \leftarrow$ CholeskyUpdate($\mathbf{C}$,$\sqrt{w(e^\star_{uv})}\mathbf{a}_{uv}$) ▷ Rank-one update
  end while
  return $\mathcal{E}$
end function

and w are clear from the context. Since $\tau_{n,w}(\mathcal{E}_{\text{init}})$ is constant, k-ESP$^+$ (5.9) can be expressed as

$$\begin{aligned} \underset{\mathcal{E}\subseteq\mathcal{C}^+}{\text{maximize}} \quad & \Phi_w(\mathcal{E}) \\ \text{subject to} \quad & |\mathcal{E}| = k. \end{aligned} \tag{5.13}$$

Theorem 5.3.1. *$t_{n,w}$ is normalized monotone supermodular for any $n \geq 2$ and positive weight function w.*

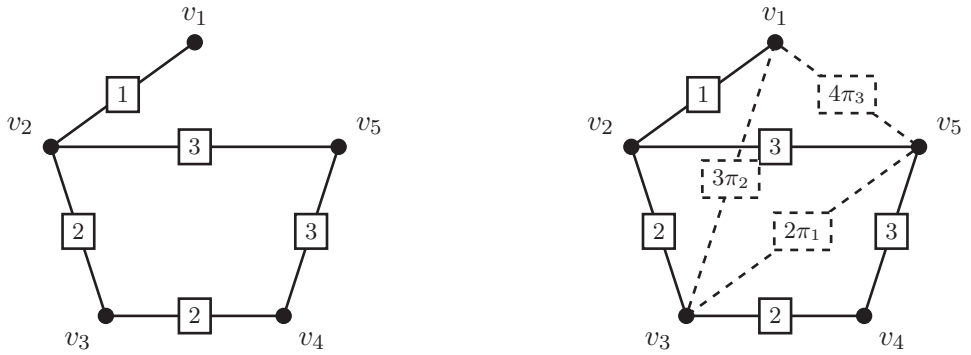
Theorem 5.3.2. *Φ_w is normalized monotone submodular for any $n \geq 2$, positive weight function w and connected base graph.*

Maximizing an arbitrary monotone submodular function subject to a cardinality constraint *can be* NP-hard in general (see e.g., the Maximum Coverage problem [72]). A classic result is due to Nemhauser et al. [118] who have shown that the greedy algorithm is a constant-factor approximation algorithm with a factor of $\eta \triangleq (1-1/e) \approx 0.63$ for maximizing any normalized monotone submodular function subject to a cardinality constraint (see Theorem A.3.1).

Let OPT be the optimum value of (5.9), $\mathcal{E}_{\text{greedy}}$ be the edges chosen by the greedy algorithm (Algorithm 7), $\tau_{\text{greedy}} \triangleq \tau_{n,w}(\mathcal{E}_{\text{greedy}} \cup \mathcal{E}_{\text{init}})$ and $\tau_{\text{init}} \triangleq \tau_{n,w}(\mathcal{E}_{\text{init}})$. Corollary 5.3.3 follows directly from Theorem 5.3.2 and Theorem A.3.1.

Corollary 5.3.3.

$$\tau_{\text{greedy}} \geq \eta \cdot \mathsf{OPT} + 1/e \cdot \tau_{\text{init}}. \tag{5.14}$$



(a) $\mathcal{G}_{\text{init}}$. **(b)** k-ESP$^+$.

Figure 5.2: Convex relaxation for k-ESP

5.3.2 Convex Relaxation

Relaxation

In this section, we design a second approximation algorithm for k-ESP$^+$ through convex relaxation. Let us begin by assigning an auxiliary variable $0 \leq \pi_i \leq 1$ to each candidate edge $e_i \in \mathcal{C}^+$. The idea is to reformulate the problem such that finding the optimal set of candidate edges is equivalent to finding the optimal value for π_i's. Let $\boldsymbol{\pi} \triangleq [\pi_1 \ \pi_2 \ \cdots \ \pi_c]^\top$ be the stacked vector of auxiliary variables. Let $\mathbf{L}_{\text{init}}$ denote the reduced weighted Laplacian matrix of $\mathcal{G}_{\text{init}}$. Define,

$$\mathbf{L}_w(\boldsymbol{\pi}) \triangleq \mathbf{L}_{\text{init}} + \sum_{e_i \in \mathcal{C}^+} \pi_i w(e_i) \mathbf{L}_{e_i} = \mathbf{A}\mathbf{W}^\pi \mathbf{A}^\top, \tag{5.15}$$

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

$$w^\pi(e_i) = \begin{cases} \pi_i w(e_i) & e_i \in \mathcal{C}^+, \\ w(e_i) & e_i \notin \mathcal{C}^+. \end{cases} \tag{5.16}$$

Lemma 5.3.1. *If $\mathcal{G}_{\text{init}}$ is connected, $\mathbf{L}_w(\boldsymbol{\pi})$ is positive definite for any $\boldsymbol{\pi} \in [0,1]^c$.*

Proof. The proof follows directly from Corollary A.4.3. □

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

problem over $\boldsymbol{\pi}$.

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

P_1 is equivalent to the definition of k-ESP$^+$ in (5.9). First note that from the generalized matrix-tree theorem (Theorem A.4.6) we know that the objective function is equal to the weighted tree-connectivity of $\mathcal{G}_\mathrm{T} = ([n], \mathcal{E}_\text{init} \cup \mathcal{C}^+)$ whose edges are weighted by w^π. The auxiliary variables act as selectors: the ith candidate edge is selected iff $\pi_i = 1$. The combinatorial difficulty of k-ESP$^+$ here is embodied in the non-convex ℓ_0-norm constraint. It is easy to see that in P_1, at the optimal solution, auxiliary variables take binary values. Hence, it is more natural to assume π_i's belong to $\{0,1\}$. This is why the following non-convex program is equivalent to P_1.

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

A natural choice for relaxing P'_1 is to replace $\pi_i \in \{0,1\}$ with $0 \leq p_i \leq 1$; i.e.,

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

where $\mathbf{p}$ is the stacked vector of p_i's. The feasible set of P_2 contains that of P'_1, and therefore the optimum value of P_2 is an upper bound for the optimum of P_1 (or, equivalently, P'_1). Note that here, the ℓ_1-norm constraint $\|\mathbf{p}\|_1 = k$ is identical to $\sum_{i=1}^c p_i = k$. P_2 is a convex optimization problem since the objective function ($\log\det$) is concave and the constraints are linear and affine in $\mathbf{p}$. In fact, P_2 is an instance of the MAXDET problem [151] subject to additional affine constraints on $\mathbf{p}$. It is worth noting that P_2 can be reached also by relaxing the non-convex ℓ_0-norm constraint in P_1 $\|\boldsymbol{\pi}\|_0 = k$ into the convex ℓ_1-norm constraint $\|\mathbf{p}\|_1 = k$. Furthermore, P_2 is also closely related to a ℓ_1-regularised variant of MAXDET,

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

This problem is a penalized form of P_2; these two problems are equivalent for some positive value of λ. Problem P_3 is also a convex optimization problem for non-negative λ. The ℓ_1-norm in P_3 penalizes the loss of sparsity, while the log-determinant rewards stronger tree-connectivity. The penalty coefficient λ is a parameter that specifies the desired degree of sparsity; i.e., a larger λ yields a sparser $\mathbf{p}$. P_3 is closely related to graphical lasso [57].

Rounding

P_2 (and P_3) can be solved globally in polynomial time using interior-point methods [15, 81]. After finding a globally optimal solution $\mathbf{p}^\star$ for the relaxed problem P_2, we ultimately need to map it into a feasible $\boldsymbol{\pi}$ for P'_1; i.e., choosing k edges from the candidate set $\mathcal{C}^+$.

Lemma 5.3.2. *$\mathbf{p}^\star$ is an optimal solution for k-ESP$^+$ iff $\mathbf{p}^\star \in \{0,1\}^c$.*

Proof. Note that $\mathbf{p}^\star \in \{0,1\}^c$ is a feasible solution for k-ESP$^+$ and it maximizes the objective function in k-ESP$^+$ over $\mathbf{p} \in [0,1]^c$. □

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 $p_i^\star$'s. We call this strategy the deterministic rounding heuristic. It will become clear shortly why this heuristic performs well in practice.

The idea behind the convex relaxation technique described so far can be seen as a graph-theoretic special case of the algorithm proposed in [81]. However, it is not clear yet how the solution of the relaxed convex problem P_2 is related to the original non-convex k-ESP$^+$ in the integer program P'_1. To answer this question, consider a randomized strategy in which one attempts to find a suboptimal solution for k-ESP$^+$ by randomly sampling candidates. In this case, for the ith candidate edge, we flip a coin whose probability of heads is p_i (independent of other candidates). The ith candidate is selected ($\pi_i = 1$) if the coin lands on head ($\pi_i = 0$ otherwise). To analyze this randomized strategy, we first need to compute the expected weighted number of spanning trees in anisotropic random graphs.

Definition 5.3.2 (Anisotropic Random Graphs)**.** Let $\mathcal{G}$ be an arbitrary anisotropic random simple undirected edge-weighted graph, in which the ith edge is *operational* with probability p_i (and *fails* with probability $1 - p_i$) independently. Let $\mathcal{G}_\bullet$ be the graph when every edge is operational, and $\mathbf{p}$ be the stacked vector of p_i's. Then we write, $\mathcal{G} \sim \mathbb{G}(\mathcal{G}_\bullet, \mathbf{p})$.

The naive procedure for computing the expected weighted number of spanning trees in such random graphs involves a summation over exponentially many terms. Theorem 5.3.4 offers an efficient and intuitive way of computing this expectation in terms of $\mathcal{G}_\bullet$ and $\mathbf{p}$.

Theorem 5.3.4 (Expected Weighted Number of Spanning Trees)**.** *For $\mathcal{G} \sim \mathbb{G}(\mathcal{G}_\bullet, \mathbf{p})$ we have, $\mathbb{E}[t_w(\mathcal{G})] = t_{\bar{w}}(\mathcal{G}_\bullet)$ where $\bar{w} : e_i \mapsto p_i w(e_i)$.*

Cohen [31] presents a different proof for this theorem for the case of unweighted graphs. Our result, however, concerns the case of weighted graphs. We also generalize this theorem to the case of random sum of arbitrary rank-one matrices in Theorem 5.3.10. The following theorem follows from Theorem 5.3.4.

Theorem 5.3.5. *Let the random variables $k^* \triangleq \sum_{i=1}^{c} \pi_i$ and $t_w^* \triangleq \det \mathbf{L}_w(\boldsymbol{\pi})$ denote, respectively, the number of chosen candidate edges and the corresponding weighted number of spanning trees achieved by the above randomized algorithm. Then,*

$$\mathbb{E}[k^*] = \sum_{i=1}^{c} p_i, \tag{5.17}$$

$$\mathbb{E}[t_w^*] = \det \mathbf{L}_w(\mathbf{p}). \tag{5.18}$$

According to Theorem 5.3.5, the randomized algorithm described above, on average, selects $\sum_{i=1}^{c} p_i$ candidate edges and achieves $\det \mathbf{L}_w(\mathbf{p})$ weighted number of spanning trees in expectation. Note that these two terms appear in the constraints and the objective of the relaxed problem P_2, respectively. Hence, the relaxed problem can be interpreted as the problem of finding the optimal sampling probabilities $\mathbf{p}$ for the randomized algorithm described above. This offers a new narrative:

Corollary 5.3.6. *The objective in P_2 is to find the optimal probabilities $\mathbf{p}^\star$ for sampling edges from $\mathcal{C}^+$ such that the weighted number of spanning trees is maximized in* expectation*, while the* expected *number of newly selected edges is equal to k.*

In other words, P_2 can be seen as a convex relaxation of P_1 at the expense of maximizing the

objective and satisfying the constraint, both *in expectation*. Similarly, e.g., (P_3) is equivalent to,

$$\begin{aligned} \underset{\mathbf{p}}{\text{maximize}} \quad & \log \mathbb{E}\left[\det \mathbf{L}_w(\boldsymbol{\pi})\right] - \lambda\, \mathbb{E}\left[\sum_{i=1}^{c} \pi_i\right] \\ \text{subject to} \quad & 0 \le p_i \le 1, \qquad\qquad \forall i \in [c]. \end{aligned} \tag{P_3'}$$

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 be used as a basis for designing randomized rounding procedures based on the randomized technique described above. If one uses $\mathbf{p}^\star$ (the fractional solution of the relaxed problem P_2) in the aforementioned randomized rounding scheme, Theorem 5.3.5 ensures that, on average, such a method attains $\det \mathbf{L}(\mathbf{p}^\star)$ by picking k new edges in expectation.

Note that merely randomly sampling candidate edges with the probabilities in $\mathbf{p}^\star$ is not sufficient to guarantee (e.g., with high probability) a feasible solution (i.e., we may select more/less than exactly k candidates). The following theorems provide a probabilistic bound on how much k^* can deviate from its expected value k.

Theorem 5.3.7 (Chernoff Bound)**.** *Let $k^* \triangleq \sum_{i=1}^{c} \pi_i$ be the number of candidates selected by randomly selecting candidates with probabilities in $\mathbf{p}^\star$. For any $0 < \delta < 1$,*

$$\mathbb{P}\left[|k^* - k| > \delta k\right] < 2 \cdot \exp\left(-\delta^2 k/3\right). \tag{5.19}$$

Theorem 5.3.8. *$|k^* - k| = O(\sqrt{k \log(k)}\,)$ with high probability.*[1]

Proof. This theorem directly follows from Theorem 5.3.7 for $\delta = O(\sqrt{\log(k)/k})$. □

Remark 10. *Note that in (P_1'), $t^* \triangleq \sum_{i=1}^{c} \pi_i$ has to be equal to k with probability one. Intuitively speaking, Theorem 5.3.8 suggests that in the relaxed convex program, almost the entire probability mass for t^* is centered around k with a width of $O(\sqrt{k \log(k)})$.*

We wrap up this section by revisiting our original deterministic rounding heuristic (i.e., selecting the k candidates with the largest $p_i^\star$'s). We call this strategy the sorting rounding. Let $\binom{\mathcal{C}^+}{k}$ denote the set of all k-subsets of $\mathcal{C}^+$. For any $\mathcal{S} \subseteq \mathcal{C}^+$, let $p_k(\mathcal{S})$ denote the conditional probability of the event in which the above randomized algorithm (i.e., selecting candidates via independent coin flips with

[1] Here, "with high probability" means with probability $p = 1 - 1/P(k)$ where $P(k)$ is a polynomial in k.

probabilities in $\mathbf{p}^\star$) chooses $\mathcal{S}$, given that $k^* = k$ candidates have been selected; i.e.,

$$p_k(\mathcal{S}) \triangleq \mathbb{P}\left[\mathcal{S} \text{ is selected} \,\middle|\, k^* = k\right]. \tag{5.20}$$

Now note that $p_k(\mathcal{S}) \propto \mathbb{P}\left[\mathcal{S} \text{ is selected} \wedge |\mathcal{S}| = k\right]$, where

$$\mathbb{P}\left[\mathcal{S} \text{ is selected} \wedge |\mathcal{S}| = k\right] = \begin{cases} \prod_{e_i \in \mathcal{S}} p_i^\star \prod_{e_j \in \mathcal{C}^+ \setminus \mathcal{S}} (1 - p_j^\star) & |\mathcal{S}| = k, \\ 0 & \text{otherwise.} \end{cases} \tag{5.21}$$

Theorem 5.3.9. *Let $\mathcal{S}_{\text{sort}}$ be the edges selected by the deterministic heuristic. We have,*

$$\mathcal{S}_{\text{sort}} = \arg\max_{\mathcal{S} \subseteq \mathcal{C}^+} p_k(\mathcal{S}). \tag{5.22}$$

Proof. Suppose $p_1^* \leq p_2^* \leq \cdots \leq p_c^*$ is the sorted version of $\{\mathbf{p}_i^\star\}_{i=1}^c$. Now for all $i \in [c]$, define $q_i^* \triangleq 1 - p_i^*$. Hence, we have $q_1^* \geq q_2^* \geq \cdots \geq q_c^*$. Now (5.22) can be expressed as,

$$\underset{\mathcal{A} \subseteq [c], |\mathcal{A}| = k}{\text{maximize}} \prod_{i \in \mathcal{A}} p_i^* \prod_{j \in [c] \setminus \mathcal{A}} q_j^*. \tag{5.23}$$

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. □

According to Theorem 5.3.9, our deterministic rounding heuristic described earlier is equivalent to selecting the most probable feasible set of k candidates, when candidates are selected with the probabilities in $\mathbf{p}^\star$.

Remark 11. *Theorem 5.3.5 (and the analysis we presented afterwards that ultimately led to Theorem 5.3.9) can be extended to the more general case of D-optimal sensor selection (see, e.g., [81]). The only nontrivial part of this extension is* (5.18) *in Theorem 5.3.5. This part was proved in Theorem 5.3.4. Therefore, in the following theorem we extend Theorem 5.3.4.*

Theorem 5.3.10 (Determinant of Random Sum of Rank-One Matrices)**.** *Suppose we are given a pair of m real n-vectors, $\{\mathbf{u}_i\}_{i=1}^m$ and $\{\mathbf{v}_i\}_{i=1}^m$. Let $\{\pi_i\}_{i=1}^m$ be m* independent *Bernoulli random variables distributed*

as,

$$\pi_i \sim \text{Bern}(p_i) \qquad i \in [m] \tag{5.24}$$

$$\pi_i \perp \pi_j \qquad i,j \in [m], i \neq j \tag{5.25}$$

where $\{p_i\}_{i=1}^m$ *are given. We have,*

$$\mathop{\mathbb{E}}_{\pi_1,\ldots,\pi_m}\Big[\det\Big(\sum_{i=1}^m \pi_i \mathbf{u}_i \mathbf{v}_i^\top\Big)\Big] = \det\Big(\sum_{i=1}^m p_i \mathbf{u}_i \mathbf{v}_i^\top\Big). \tag{5.26}$$

Note that Theorem 5.3.4 is a special case of Theorem 5.3.10 in which $\mathbf{u}_i$ and $\mathbf{v}_i$ are equal to the ith column of the reduced (weighted) incidence matrix of graph $\mathcal{G}_\bullet$.

5.3.3 Certifying Near-Optimality

In "large" instances of k-ESP$^+$, it is impractical to compute OPT through exhaustive search. Fortunately, the approximation algorithms developed in this chapter yield lower and upper bounds for OPT that can be quite tight in practice. Let $\tau^\star_{\text{cvx}}$ be the optimum value of P_2. Moreover, let 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 5.3.11 follows from the analysis presented for our greedy and convex approximation algorithms.

Corollary 5.3.11.

$$\max\Big\{\tau_{\text{greedy}},\tau_{\text{cvx}}\Big\} \leq \mathsf{OPT} \leq \min\Big\{\mathcal{U}_{\text{greedy}},\tau^\star_{\text{cvx}}\Big\} \tag{5.27}$$

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

The lower bounds in Corollary 5.3.11 correspond to the suboptimal solutions of the proposed approximation algorithms. $\mathcal{U}_{\text{greedy}}$ can be obtained by a straightforward manipulation of Theorem A.3.1 and Theorem 5.3.2. Furthermore, $\tau^\star_{\text{cvx}}$ is an upper bound for OPT because 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 asses the quality of any feasible design. 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) solution of the greedy algorithm, (ii) 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 $\mathcal{L}$ and $\mathcal{U}$ denote the lower and upper bounds in (5.27), respectively. From Corollary 5.3.11 we have,

$$\max\{0,\mathcal{L}-\tau'\} \leq \underbrace{\mathsf{OPT}-\tau'}_{\text{approximation gap}} \leq \mathcal{U}-\tau', \tag{5.28}$$

$$\max\{1,\mathcal{L}/\tau'\} \leq \underbrace{\mathsf{OPT}/\tau'}_{\text{approximation ratio}} \leq \mathcal{U}/\tau'. \tag{5.29}$$

Thus, $\mathcal{U}-\tau'$ and $\mathcal{U}/\tau'$ can be used as near-optimality certificates for any arbitrary design.

5.4 Dual Problem: δ-ESP*

The main purpose of this chapter is to design sparse graphs with strong tree-connectivity. We pursued this objective in k-ESP$^+$ by searching for the graph with the maximum weighted tree-connectivity among a class of graphs with a specified degree of sparsity. Alternatively, we may ask ourselves: "what is the sparsest graph that achieves a desired level of weighted tree-connectivity?". Let us first formalize the dual problem.

Problem 5.4.1 (δ-ESP*)**.** 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 $0\leq\delta\leq\Phi_w(\mathcal{C}^+)$; i.e.,

$$\begin{aligned}&\underset{\mathcal{E}\subseteq\mathcal{C}^+}{\text{minimize}} \quad |\mathcal{E}|\\ &\text{subject to} \quad \Phi_w(\mathcal{E})\geq\delta.\end{aligned} \tag{5.30}$$

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*.

5.4.1 Greedy Algorithm

Algorithm

The greedy algorithm designed for k-ESP$^+$ in Algorithm 7 solves k instances of 1-ESP$^+$. The very same algorithm can be applied to the dual problem, δ-ESP*, after a minor modification of the stopping criterion: instead of solving exactly k instances of 1-ESP$^+$, we keep solving 1-ESP$^+$ subproblems until the resulting tree-connectivity gain is at least δ (or, alternatively, when there are no more edges

left in $\mathcal{C}^+$, which indicates an empty feasible set). The greedy algorithm for approximating the solution of (5.30) is described by Algorithm 8.

Analysis

Our analysis of the greedy algorithm in k-ESP$^+$ was based on the seminal work of Nemhauser et al. [118] and our Theorem 5.3.2 which established the monotone log-submodularity of $\tau_{n,w}$. Motivated by the Set Cover problem, Wolsey [158, Problem (Q)] has analyzed the performance of the greedy algorithm for solving the following general class of problems,

$$\begin{aligned} &\underset{\mathcal{A}\subseteq\mathcal{W}}{\text{minimize}} \quad \sum_{a_i\in\mathcal{A}} f_i \\ &\text{subject to} \quad z(\mathcal{A}) = z(\mathcal{W}), \end{aligned} \tag{5.31}$$

in which $f_i > 0$ for all $a_i \in \mathcal{W}$ and $z : 2^{\mathcal{W}} \to \mathbb{R}$ is any monotone submodular function. Now note that the dual problem, (5.30), is a special case of (5.31), in which $\mathcal{W} = \mathcal{C}^+$, $f_i = 1$ for all $a_i \in \mathcal{A}$, and finally z is $\mathcal{A} \mapsto \min\{\delta, \Phi_w(\mathcal{A})\}$. Note that this function is monotone submodular since,

1. Φ_w is monotone submodular according to Theorem 5.3.2, and
2. constant truncation preserves both monotonicity and submodularity according to Theorem A.3.2 as noted by Wolsey [158].

Let f_{greedy} and OPT be the value of the greedy algorithm and the optimum value for (5.31). For the general case of (5.31)—where z does not need to be an integer-valued function—Wolsey [158, Theorem 1] has established three *a posteriori* bounds of the form,

$$f_{\text{greedy}} \leq (1 + \log\gamma)\mathsf{OPT}, \tag{5.32}$$

for 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 8) in δ-ESP*.

Corollary 5.4.1. *Let k_{opt} and k_{greedy} be the optimum value of* (5.30) *and the value achieved by Algorithm 8, respectively. Also, let $\widetilde{\Phi}_{\text{greedy}}$ be the tree-connectivity gain achieved by the greedy algorithm one step before termination. Define $s \triangleq \delta/(\delta - \widetilde{\Phi}_{\text{greedy}})$. Then we have*

$$k_{\text{greedy}} \leq (1 + \log s)k_{\mathsf{opt}}. \tag{5.33}$$

Algorithm 8 Greedy Dual Edge Selection

function GreedyDualESP($\mathbf{L}_{\text{init}}$,$\mathcal{C}^+$,δ)
  $\mathcal{E} \leftarrow \varnothing$
  $\mathbf{L} \leftarrow \mathbf{L}_{\text{init}}$
  $\mathbf{C} \leftarrow$Cholesky($\mathbf{L}$)
  while $(\log\det \mathbf{L} - \log\det \mathbf{L}_{\text{init}} < \delta) \wedge (\mathcal{E} \neq \mathcal{C}^+)$ **do**
    $e^\star_{uv} \leftarrow$ NextBestEdge($\mathcal{C}^+ \setminus \mathcal{E}$,$\mathbf{C}$)
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{e^\star_{uv}\}$
    $\mathbf{a}_{uv} \leftarrow \mathbf{e}_u - \mathbf{e}_v$
    $\mathbf{L} \leftarrow \mathbf{L} + w(e^\star_{uv})\mathbf{a}_{uv}\mathbf{a}_{uv}^\top$
    $\mathbf{C} \leftarrow$ CholeskyUpdate($\mathbf{C}$,$\sqrt{w(e^\star_{uv})}\mathbf{a}_{uv}$) ▷ Rank-one update
  end while
  return $\mathcal{E}$
end function

5.4.2 Convex Relaxation

Relaxation

In this section, we demonstrate how the dual problem, δ-ESP*, can be formulated as an integer program similar to what we did in P'_1. We then study a convex relaxation of δ-ESP* and discuss rounding strategies for mapping the fractional solution of the convex program back to a feasible suboptimal design for δ-ESP*.

As before, let $\tau_{\text{init}} \triangleq \log\det \mathbf{L}(\mathbf{0})$. The dual problem can be expressed as

$$\begin{aligned} \underset{\boldsymbol{\pi}}{\text{minimize}} \quad & \sum_{i=1}^{c} \pi_i \\ \text{subject to} \quad & \log\det \mathbf{L}(\boldsymbol{\pi}) \geq \delta + \tau_{\text{init}}, \\ & \pi_i \in \{0,1\}, \qquad \forall i \in [c]. \end{aligned} \tag{D$_1$}$$

The combinatorial difficulty of δ-ESP* is manifested in the integral constraints of D_1. Relaxing the integral constraints on $\boldsymbol{\pi}$ (and like before, replacing binary π_i's with probabilities p_i's) yields the following convex optimization problem,

$$\begin{aligned} \underset{\mathbf{p}}{\text{minimize}} \quad & \sum_{i=1}^{c} p_i \\ \text{subject to} \quad & \log\det \mathbf{L}(\mathbf{p}) \geq \delta + \tau_{\text{init}}, \\ & 0 \leq p_i \leq 1, \qquad \forall i \in [c]. \end{aligned} \tag{D$_2$}$$

Algorithm 9 Deterministic rounding for δ-ESP*

```
function RoundDualESP(p*)
    π ← 0
    i ← 0
    // returns the indices of the sorted p*
    s ←SortDescending(p*)                    ▷ Such that p*_{s_1} ≥ p*_{s_2} ≥ ··· ≥ p*_{s_c}
    while (log det L(π) − log det L_init < δ) ∧ (π ≠ 1) do
        π_{s_i} ← 1
        i ← i + 1
    end while
    return π_sort = π
end function
```

D_2 is a convex optimization problem since the objective is linear and log-determinant is concave for positive definite matrices—here $\mathbf{L}(\boldsymbol{\pi})$ is guaranteed to be positive definite, e.g., if the base graph $\mathcal{G}_{\text{init}}$ is connected, see Lemma 5.3.1. D_2 can be solved efficiently using interior-point methods. Let $\mathbf{p}^\star$ be the minimizer of D_2. Note that $\lceil \sum_{i=1}^{c} p_i^\star \rceil$ is a lower bound for k_{opt}, the optimum value of δ-ESP*, since the feasible set of D_1 is a subset of the feasible set of D_2.

Rounding

Lemma 5.4.1. *$\mathbf{p}^\star$ is an optimal solution for δ-ESP* iff $\mathbf{p}^\star \in \{0,1\}^c$.*

Proof. Similar to the proof of Lemma 5.3.2. □

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 9. As we saw before, the fractional values $\mathbf{p}$ can be interpreted as the probability of independently selecting candidate edges. Hence, similar to Corollary 5.3.6, the following corollary readily follows from Theorem 5.3.5.

Corollary 5.4.2. *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 weighted number of spanning trees is at least $e^{\tau_{\text{init}}+\delta}$; i.e.,*

$$\begin{aligned} \underset{\mathbf{p}}{\text{minimize}} \quad & \mathbb{E}\left[\sum_{i=1}^{c} \pi_i\right] \\ \text{subject to} \quad & \log \mathbb{E}\left[\det \mathbf{L}(\boldsymbol{\pi})\right] \geq \delta + \tau_{\mathit{init}}, \\ & 0 \leq p_i \leq 1, \qquad \forall i \in [c]. \end{aligned} \tag{D_2'}$$

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

This narrative suggests a randomized rounding scheme in which $e_i \in \mathcal{C}^+$ is selected with probability $p_i^\star$. The expected number of selected edges by this procedure is

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

while on average the δ-constraint is satisfied. Needless to say, this is insufficient for guaranteeing a feasible solution (e.g., with high probability). However, this new narrative explains why the deterministic rounding procedure in Algorithm 9 performs well in practice. Recall that in (5.20), $p_k(\mathcal{S})$ was defined to be the conditional probability of sampling exactly the set $\mathcal{S} \subseteq \mathcal{C}^+$, given that exactly k candidates have been selected. This probability was computed in (5.21) up to a constant factor. Theorem 5.4.3 readily follows from Theorem 5.3.9 and Algorithm 9.

Theorem 5.4.3. *For any $k \in [c]$ define $\mathcal{S}_k^\star \triangleq \arg\max_{\mathcal{S}\subseteq\mathcal{C}^+} p_k(\mathcal{S})$. $\mathcal{S}_k^\star$ is the most probable outcome when k edges have been selected after flipping coins with probabilities in $\mathbf{p}^\star$. The output of Algorithm 9 corresponds to $\mathcal{S}_{k_{\min}}^\star$ where $k_{\min}$ is the smallest $k \in [c]$ such that $\Phi_w(\mathcal{S}_k^\star) \geq \delta$.*

5.4.3 Certifying Near-Optimality

Computing the optimal solution of δ-ESP* by brute force is impractical in large instances of the dual problem, and thus k_{opt} is generally beyond our reach in such scenarios. That being said, as we saw in the case of k-ESP$^+$, our approximation algorithms can provide lower and upper bounds for k_{opt}.

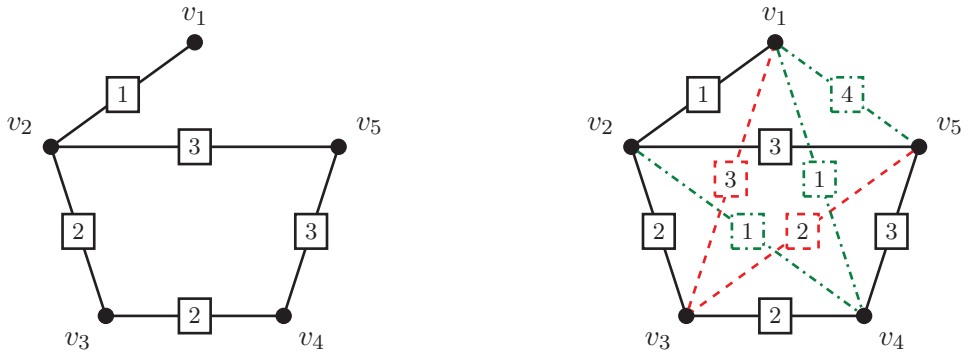
Corollary 5.4.4. *Define $\zeta^* \triangleq 1/(1+\log s)$ where s is the parameter defined in Theorem 5.4.1. Let k_{cvx} be the number of new edges selected by the deterministic rounding procedure in Algorithm 9. Then,*

$$\max\left\{\left\lceil \zeta^* k_{\text{greedy}} \right\rceil, \left\lceil \sum_{i=1}^{c} p_i^\star \right\rceil\right\} \leq k_{\mathsf{opt}} \leq \min\left\{k_{\text{greedy}}, k_{\text{cvx}}\right\}. \tag{5.34}$$

The bounds provided by Corollary 5.4.4 can be used to bound the gap between k_{opt} and any suboptimal design $\mathcal{E}'$ with a value of k'. Let $k_{\mathcal{L}}$ and $k_{\mathcal{U}}$ denote the lower and upper bounds in Corollary 5.4.4. These bounds can be computed by running the corresponding approximation algorithms. Now we have,

$$\max\{0, k' - k_{\mathcal{U}}\} \leq \underbrace{k' - k_{\mathsf{opt}}}_{\text{approximation gap}} \leq k' - k_{\mathcal{L}}, \tag{5.35}$$

$$\max\{1, k'/k_{\mathcal{U}}\} \leq \underbrace{k'/k_{\mathsf{opt}}}_{\text{approximation ratio}} \leq k'/k_{\mathcal{L}}. \tag{5.36}$$



(a) $\mathcal{G}_{\text{init}}$. **(b)** k-ESP$^+$.

Figure 5.3: Partition matroid constraints.

5.5 k-ESP$^+$ Over a Partition Matroid

In this section, we extend k-ESP$^+$ by imposing a partition matroid constraint.[2]

Lemma 5.5.1. *Replacing the cardinality constraint $|\mathcal{E}| = k$ with $|\mathcal{E}| \leq k$ in k-ESP$^+$ does not change the optimal solutions.*

Proof. Recall that Φ_w is monotone, and therefore picking additional candidates cannot reduce the tree-connectivity. Hence, the tree-connectivity of any design with $|\mathcal{E}| < k$ edges can be improved by picking $k - |\mathcal{E}|$ additional candidates. □

Therefore, according to Lemma 5.5.1, the cardinality constraint in k-ESP$^+$ is equivalent to imposing $\mathcal{E} \in \mathcal{I}_{\mathtt{u}}$, where $\mathcal{I}_{\mathtt{u}}$ represents the uniform matroid $(\mathcal{C}^+, \mathcal{I}_{\mathtt{u}})$ and is defined as

$$\mathcal{I}_{\mathtt{u}} \triangleq \Big\{\mathcal{A} \subseteq \mathcal{C}^+ : |\mathcal{A}| \leq k\Big\}. \tag{5.37}$$

Now let $\{\mathcal{C}_i^+\}_{i=1}^r$ be a partition for $\mathcal{C}^+$. We call each $\mathcal{C}_i^+$ a *block*. Furthermore, suppose for each $\mathcal{C}_i^+$, there is an integer k_i such that $k_i \leq c_i \triangleq |\mathcal{C}_i^+|$.[3] Define

$$\mathcal{I}_{\mathtt{p}} \triangleq \Big\{\mathcal{A} \subseteq \mathcal{C}^+ \ : \ |\mathcal{A} \cap \mathcal{C}_i^+| \leq k_i \ \ \forall i \in [r]\Big\}. \tag{5.38}$$

$(\mathcal{C}^+, \mathcal{I}_{\mathtt{p}})$ is a partition matroid that generalizes the uniform matroid $(\mathcal{C}^+, \mathcal{I}_{\mathtt{u}})$. Consequently, k-ESP$^+$ can be generalized as,

$$\begin{aligned} &\underset{\mathcal{E} \subseteq \mathcal{C}^+}{\text{maximize}} && \Phi_w(\mathcal{E}) \\ &\text{subject to} && \mathcal{E} \in \mathcal{I}_{\mathtt{p}}. \end{aligned} \tag{5.39}$$

[2] See Section A.3 for the definition of uniform and partition matroids.
[3] As it will become clear shortly, the case of $k_i = 0$ is equivalent to excluding $\mathcal{C}_i^+$ from the candidate set.

Intuitively speaking, (5.39) generalizes k-ESP$^+$ by enforcing local cardinality constraints (i.e., k_i's) on disjoint subsets of the candidate set $\mathcal{C}^+$. Note that k-ESP$^+$ is a special case of (5.39) with $r = 1$ and $k_1 = k$.

Example 5.5.1. A simple instance of (5.39) is depicted in Figure 5.3, where the candidate set is partitioned into two subsets (shown by the green and red edges) with $k_1 = 2$ and $k_2 = 1$.

By choosing different partitions for $\mathcal{C}^+$ and enforcing different local budgets, we can model a wide variety of graph synthesis problems as instances of (5.39). For example, one may design a partitioning-budgeting scheme in which partitions represent a hierarchy among the candidate edges (e.g., based on costs, significance, etc).

Example 5.5.2. An interesting scenario is when the partitions are chosen to enforce an upper bound on the degree of a subset of vertices $\{v_i\}_{i=1}^r \subseteq \mathcal{V}$ such that $\{v_i,v_j\} \notin \mathcal{C}^+$ for all $i,j \in [r]$:

$$\begin{aligned} \underset{\mathcal{E}\subseteq\mathcal{C}^+}{\text{maximize}} \quad & \Phi_w(\mathcal{E}) \\ \text{subject to} \quad & |\mathcal{E}| = k, \\ & \deg(v_i) = d_i, \qquad \forall i \in [r]. \end{aligned} \tag{5.40}$$

In the following, we explain why (5.40) is a valid instance of (5.39). We do so by constructing an instance of (5.39) that has the same solution as (5.40). First, note that in the equality constraints in (5.40), $=$ can be replaced by $\leq$ since Φ_w is monotone (see Lemma 5.5.1). Then,

$$k_i = d_i \qquad i \in [r] \tag{5.41}$$

$$\mathcal{C}_i^+ = \{e \in \mathcal{C}^+ : v_i \in e\} \qquad i \in [r] \tag{5.42}$$

$$\mathcal{C}_{r+1}^+ = \mathcal{C}^+ \setminus \bigcup_{i=1}^{r} \mathcal{C}_i^+ \tag{5.43}$$

$$k_{r+1} = k - \sum_{i=1}^{r} d_i. \tag{5.44}$$

It is easy to verify that, under the aforementioned assumptions, the above construction partitions $\mathcal{C}^+$ and hence is a valid instance of (5.39) with the same optimal solutions as (5.40).

5.5.1 Greedy Algorithm

Algorithm and Complexity Analysis

The greedy algorithm for solving the edge selection problem under a partition matroid constraint (5.39) is summarized in Algorithm 10. The only difference between Algorithm 7 and this algorithm is that Algorithm 10 abides by the partition matroid constraint and selects the next available candidate edge only if adding it to the graph does not violate the corresponding local cardinality constraint. This is done efficiently in Algorithm 10 by removing each block from the set of remaining candidates immediately after we reach the corresponding cardinality constraint. Define $k_{\text{total}} \triangleq \sum_{i=1}^{r} k_i$. The time complexity of Algorithm 10 is bounded by $O(n^3 + k_{\text{total}} c n^2)$.[4]

Performance Analysis

Fisher et al. [52] have proved that the greedy strategy yields a $1/2$-approximation for the general case of maximizing a monotone submodular function subject to a matroid constraint.[5] The following corollary is a direct result of [52] and Theorem 5.3.2.

[4] See the complexity analysis presented for Algorithm 7 in Section 5.3.1.

[5] As noted earlier, for the special case of uniform matroids, the greedy heuristic is guaranteed to provide a η-approximation, where $\eta \triangleq 1 - 1/e \approx 0.63$.

Algorithm 10 Greedy Algorithm for the Generalized k-ESP$^+$

function GreedyPartitionESP($\mathbf{L}_{\text{init}}$,$\{\mathcal{C}_i^+\}_{i=1}^r$,$\{k_i\}_{i=1}^r$)
  $\mathcal{E} \leftarrow \varnothing$
  $\mathbf{L} \leftarrow \mathbf{L}_{\text{init}}$
  $\mathbf{C} \leftarrow$Cholesky($\mathbf{L}$)
  $\mathcal{C}^+ \leftarrow \bigcup_{i=1}^r \mathcal{C}_i^+$
  while $\mathcal{C}^+ \neq \varnothing$ **do**
    $e_{uv}^\star \leftarrow$ NextBestEdge($\mathcal{C}^+$,$\mathbf{C}$)
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{e_{uv}^\star\}$
    $\mathbf{a}_{uv} \leftarrow \mathbf{e}_u - \mathbf{e}_v$
    $\mathbf{L} \leftarrow \mathbf{L} + w(e_{uv}^\star)\mathbf{a}_{uv}\mathbf{a}_{uv}^\top$
    $\mathbf{C} \leftarrow$ CholeskyUpdate($\mathbf{C}$,$\sqrt{w(e_{uv}^\star)}\mathbf{a}_{uv}$) ▷ Rank-one update
    // returns the index of the block that contains $e_{uv}^\star$
    $i_{uv} \leftarrow$ BlockIndex($e_{uv}^\star$)
    $k_{i_{uv}} \leftarrow k_{i_{uv}} - 1$
    $\mathcal{C}^+ \leftarrow \mathcal{C}^+ \setminus \{e_{uv}^\star\}$
    if $k_{i_{uv}} = 0$ **then**
      $\mathcal{C}^+ \leftarrow \mathcal{C}^+ \setminus \mathcal{C}_{i_{uv}}^+$
    end if
  end while
  return $\mathcal{E}$
end function

Corollary 5.5.1. *Let* τ_{greedy} *and* OPT *be the tree-connectivity achieved by Algorithm 10 and the optimum value of an instance of* (5.39)*, respectively. We have,*

$$\tau_{\text{greedy}} \geq 1/2 \left(\mathsf{OPT} + \tau_{\text{init}}\right). \tag{5.45}$$

It is worth mentioning that the $1/2$ approximation factor for maximizing any monotone submodular function over a matroid has been improved recently by a randomized algorithm. This result is due to Calinescu et al. [18], who have shown that there exists a randomized algorithm that yields a $(1 - 1/e)$-approximation.

5.5.2 Convex Relaxation

Relaxation

In this section, we present a convex relaxation approximation algorithm for the generalized k-ESP$^+$ over a partition matroid as defined in (5.39). First, note that using the indicator variables $\boldsymbol{\pi}$, we can express (5.39) as

$$\begin{aligned} \underset{\boldsymbol{\pi}}{\text{maximize}} \quad & \log\det \mathbf{L}(\boldsymbol{\pi}) \\ \text{subject to} \quad & \sum_{e_i \in \mathcal{C}_j^+} \pi_i \leq k_j, \qquad \forall j \in [r] \\ & \pi_i \in \{0,1\}, \qquad \forall i \in [c]. \end{aligned} \tag{P_4}$$

Since $\log\det \mathbf{L}(\boldsymbol{\pi})$ is non-decreasing in every π_i's, the optimal solution of P_4 is also the optimal solution of,

$$\begin{aligned} \underset{\boldsymbol{\pi}}{\text{maximize}} \quad & \log\det \mathbf{L}(\boldsymbol{\pi}) \\ \text{subject to} \quad & \sum_{e_i \in \mathcal{C}_j^+} \pi_i = k_j, \qquad \forall j \in [r] \\ & \pi_i \in \{0,1\}, \qquad \forall i \in [c]. \end{aligned} \tag{$\bar{\mathrm{P}}_4$}$$

Next, we relax the integral constraint on $\boldsymbol{\pi}$ (and, as we did before, replace it with $\mathbf{p}$),

$$\begin{aligned} \underset{\mathbf{p}}{\text{maximize}} \quad & \log\det \mathbf{L}(\mathbf{p}) \\ \text{subject to} \quad & \sum_{e_i \in \mathcal{C}_j^+} p_i = k_j, \qquad \forall j \in [r] \\ & 0 \leq p_i \leq 1, \qquad \forall i \in [c]. \end{aligned} \tag{P_5}$$

P_5 is a convex optimization problem and can be solved in polynomial time using the interior-point methods.

Rounding

Let $\mathbf{p}^\star$ be the optimal solution of P_5.

Lemma 5.5.2. *$\mathbf{p}^\star$ is an optimal solution for $\bar{\mathrm{P}}_4$ iff $\mathbf{p}^\star \in \{0,1\}^c$.*

Proof. Similar to the proof of Lemma 5.3.2. □

Once again, $\mathbf{p}^\star$ generally contains fractional values that need to be mapped into a feasible integral suboptimal solution for (5.39) by a rounding scheme. A simple deterministic rounding strategy for the solution of P_5 is to pick the k_i candidates in $\mathcal{C}_i^+$ that have the largest fractional $p_j^\star$'s for each $\mathcal{C}_i^+$ ($i \in [r]$).

Now let us revisit the idea of randomly sampling candidate edges with probabilities in $\mathbf{p}^\star$ (independently). Theorem 5.3.5 can be generalized as follows.

Theorem 5.5.2. *Let the random variables $k_1^*,k_2^*,\ldots,k_p^*$ and t_w^* denote, respectively, the number of chosen candidate edges from the ith block $\mathcal{C}_i^+$ ($i \in [r]$) and the corresponding weighted number of spanning trees achieved by the above randomized algorithm. Then,*

$$\mathbb{E}\left[k_i^*\right] = \sum_{e_j \in \mathcal{C}_i^+}^{c} p_j, \qquad \forall i \in [r] \tag{5.46}$$

$$\mathbb{E}\left[t_w^*\right] = \det \mathbf{L}_w(\mathbf{p}). \tag{5.47}$$

Proof. The proof is similar to the proof of Theorem 5.3.5. □

Corollary 5.5.3 follows from Theorem 5.5.2.

Corollary 5.5.3. *The objective in P_5 is to find the optimal probabilities $\mathbf{p}^\star$ for sampling edges from $\mathcal{C}^+$ such that the weighted number of spanning trees is maximized in* expectation, *while the* expected *number of newly selected edges in the ith block is equal to k_i for all $i \in [r]$. In other words, the following optimization problem is equivalent to* (P_5).

$$\begin{aligned} \underset{\mathbf{p}}{\text{maximize}} \quad & \log \mathbb{E}\left[\det \mathbf{L}(\boldsymbol{\pi})\right] \\ \text{subject to} \quad & \mathbb{E}\left[\sum_{e_i \in \mathcal{C}_j^+} \pi_i\right] = k_j, \qquad \forall j \in [r] \\ & 0 \le p_i \le 1, \qquad \forall i \in [c] \end{aligned} \tag{P_5'}$$

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

Finally, Theorem 5.3.9 can also be generalized to the case of partition matroids. For any $\mathcal{S} \subseteq \mathcal{C}^+$ define,

$$p_{k_{1:r}}(\mathcal{S}) \triangleq \mathbb{P}\left[\mathcal{S} \text{ is selected} \,\middle|\, \bigwedge_{i\in[r]} |\mathcal{S} \cap \mathcal{C}_i^+| = k_i\right]. \tag{5.48}$$

This conditional probability can be computed up to a constant factor:

$$p_{k_{1:r}}(\mathcal{S}) \propto \mathbb{P}\left[\mathcal{S} \text{ is selected} \bigwedge_{i\in[r]} |\mathcal{S} \cap \mathcal{C}_i^+| = k_i\right] \tag{5.49}$$

$$= \begin{cases} \prod_{e_i\in\mathcal{S}} p_i^\star \prod_{e_j\in\mathcal{C}^+\setminus\mathcal{S}}(1-p_j^\star) & \text{if } \forall i \in [r] : |\mathcal{S} \cap \mathcal{C}_i^+| = k_i, \\ 0 & \text{otherwise.} \end{cases} \tag{5.50}$$

The following theorem extends Theorem 5.3.9 and provides a justification for the deterministic rounding scheme described earlier.

Theorem 5.5.4. *Let $\mathcal{S}_{\text{sort}}$ be the edges selected by the deterministic heuristic described above applied to P_5. We have,*

$$\mathcal{S}_{\text{sort}} = \arg\max_{\mathcal{S}\subseteq\mathcal{C}^+} p_{k_{1:r}}(\mathcal{S}). \tag{5.51}$$

Proof. The proof follows closely that of Theorem 5.3.9. □

5.5.3 Certifying Near-Optimality

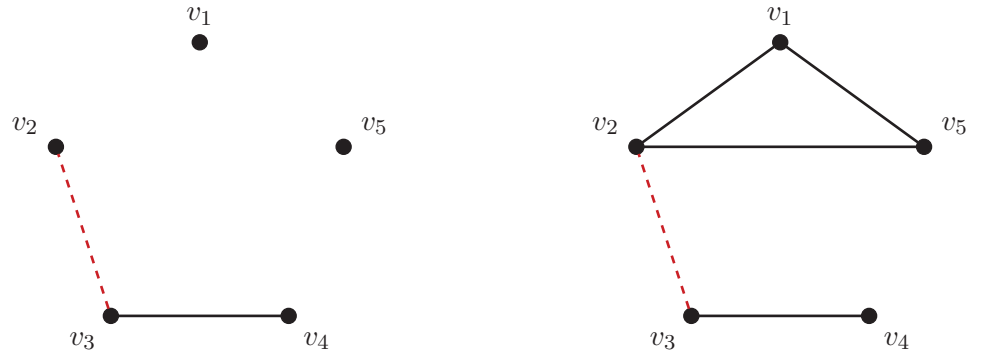
As we have seen multiple times, the approximation algorithms yield the following bounds on OPT (i.e., maximum tree-connectivity in P_4). Define $\tau^\star_{\text{cvx}} \triangleq \log\det \mathbf{L}(\mathbf{p}^\star)$. Moreover, let τ_{cvx} be the value of P_4 after rounding the fractional solution $\mathbf{p}^\star$.

Corollary 5.5.5.

$$\max\left\{\tau_{\text{greedy}}, \tau_{\text{cvx}}\right\} \leq \mathsf{OPT} \leq \min\left\{\mathcal{U}_{\text{greedy}}, \tau^\star_{\text{cvx}}\right\} \tag{5.52}$$

where $\mathcal{U}_{\text{greedy}} \triangleq 2\,\tau_{\text{greedy}} - \tau_{\text{init}}$.

As we saw in k-ESP$^+$, the bounds provided by the approximation algorithms can be used to assess the quality of any suboptimal design whenever it is impractical to compute OPT. Let $\mathcal{L}$ and



(a) $\mathcal{G}_{\mathcal{A}} = (\mathcal{V},\mathcal{A})$. **(b)** $\mathcal{G}_{\mathcal{B}} = (\mathcal{V},\mathcal{B})$.

Figure 5.4: A counterexample: tree-connectivity $\tau_{n,w}$ is *not* submodular if the underlying graph is disconnected.

$\mathcal{U}$ denote the lower and upper bounds in (5.52), respectively. Then, according to Corollary 5.5.5, for any arbitrary (feasible) design $\mathcal{E}'$ we have,

$$\max\{0,\mathcal{L}-\tau'\} \leq \underbrace{\mathsf{OPT}-\tau'}_{\text{approximation gap}} \leq \mathcal{U}-\tau', \tag{5.53}$$

$$\max\{1,\mathcal{L}/\tau'\} \leq \underbrace{\mathsf{OPT}/\tau'}_{\text{approximation ratio}} \leq \mathcal{U}/\tau'. \tag{5.54}$$

The above bounds on the approximation ratio/gap can be computed easily by running the proposed approximation algorithms.

5.6 Improper k-ESP$^+$

An assumption that was made in the definition of k-ESP$^+$ and its variants is that the base graph $\mathcal{G}_{\text{init}}$ has to be connected. By an *improper instance of* k-ESP$^+$, here we refer to a scenario in which this assumption does not hold. Note that our convex relaxation algorithm works fine for improper k-ESP$^+$ as long as $([n],\mathcal{E}_{\text{init}} \cup \mathcal{C}^+)$ is connected.[6] At first glance, the importance of this assumption may not be clear. However, as the following counterexample demonstrates, this assumption turns out to be essential for guaranteeing the performance of the greedy algorithm (Algorithm 7).

[6] If this condition is not met, any feasible design is an optimal solution and $\mathsf{OPT} = 0$.

Example 5.6.1 (Counterexample). Figure 5.4 illustrates two graphs, $\mathcal{G}_\mathcal{A}$ and $\mathcal{G}_\mathcal{B}$, both over the same set of vertices $\mathcal{V} = \{v_1, v_2, \dots, v_5\}$. The edges in both graphs are shown with black solid lines. The edge set of graph $\mathcal{G}_\mathcal{A}$ is a subset of the edges in $\mathcal{G}_\mathcal{B}$, i.e., $\mathcal{E}_\mathcal{A} \subset \mathcal{E}_\mathcal{B}$. For simplicity, we assume unit weights for all edges. Now consider the case in which a new edge, $e = \{v_2, v_3\} \notin \mathcal{E}_\mathcal{B}$ (drawn with the red dashed line) is added to both graphs. From the definition of sumodularity (Definition A.3.2) we know that for $\tau_{n,w}$ to be submodular, we must have

$$\tau_{n,w}(\mathcal{E}_\mathcal{A} \cup \{e\}) - \tau_{n,w}(\mathcal{E}_\mathcal{A}) \overset{?}{\geq} \tau_{n,w}(\mathcal{E}_\mathcal{B} \cup \{e\}) - \tau_{n,w}(\mathcal{E}_\mathcal{B}). \tag{5.55}$$

Note that,

$$\tau_{n,w}(\mathcal{E}_\mathcal{A}) = \tau_{n,w}(\mathcal{E}_\mathcal{B}) = 0, \tag{5.56}$$

since both graphs are initially disconnected. Similarly, note that $\mathcal{G}_\mathcal{A}$ remains disconnected even after adding e; thus $\tau_{n,w}(\mathcal{E}_\mathcal{A} \cup \{e\}) = 0$. Therefore, the LHS of (5.55) is equal to zero, while the RHS in this example is equal to $\log(3) \not\leq 0$.

As illustrated by Example 5.6.1, tree-connectivity $\tau_{n,w}$ can be non-submodular if the graph is not connected. This is why we defined the tree-connectivity gain function Φ_w with respect to a base graph $\mathcal{G}_{\text{init}} = ([n], \mathcal{E}_{\text{init}})$ (see Definition 5.3.1).

It is important to note that the analysis and performance guarantees provided earlier for the greedy algorithm in k-ESP$^+$ hold only in the proper instances of the problem where $\mathcal{G}_{\text{init}}$ is connected. As we see shortly in the following sections, this assumption naturally holds in the standard SLAM graphs, since the odometry spanning tree always guarantees connectivity. That being said, there are also important scenarios in which $\mathcal{G}_{\text{init}}$ is not necessarily connected. Take for example the classic problem of finding t-optimal graphs in $\mathbb{G}_{n,m}$ in which $\mathcal{E}_{\text{init}} = \varnothing$. In the rest of this section, we show how the greedy algorithm can be adapted to such improper instances of k-ESP$^+$.

5.6.1 Most Valuable Spanning Tree

In this section, we study an improper instance of improper k-ESP$^+$ with $\mathcal{G}_{\text{init}} = ([n], \varnothing)$ and $k = n-1$. As it will become clear shortly, the optimal design in this case is the most valuable spanning tree (MVST). This special case is significant due to multiple reasons:

1. Any improper k-ESP$^+$ with $\mathcal{G}_{\text{init}} = ([n], \varnothing)$ and $k < n-1$ is a degenerate problem with $\mathsf{OPT} = 0$.

2. We use this special case as a tool for tackling the more general improper instances of k-ESP$^+$.

3. This problem emerges from SLAM and other applications (see Section 5.7).

4. A greedy algorithm can find the MVST in polynomial time.

Unweighted Graphs: Let us begin by considering unweighted graphs (or, equivalently, when all edges have the same weight). Let $\mathcal{E} \subseteq \mathcal{C}^+$ be a feasible set of edges; i.e., $|\mathcal{E}| = n - 1$. Then,

$$t_{n,w}(\mathcal{E}) = \begin{cases} 1 & \text{if } ([n],\mathcal{E}) \text{ is a tree,} \\ 0 & \text{otherwise.} \end{cases} \tag{5.57}$$

Therefore, any $\mathcal{E}$ for which $([n],\mathcal{E})$ is a spanning tree of $([n],\mathcal{C}^+)$ will be an optimal design.

Weighted Graphs: If the edges are weighted by some $w : \binom{[n]}{2} \to \mathbb{R}_{\geq 1}$, we have

$$t_{n,w}(\mathcal{E}) = \begin{cases} \mathbb{V}_w([n],\mathcal{E}) & \text{if } ([n],\mathcal{E}) \text{ is a tree,} \\ 0 & \text{otherwise.} \end{cases} \tag{5.58}$$

Here $\mathbb{V}_w$ is the value function defined in Definition A.4.1 as,

$$\mathbb{V}_w([n],\mathcal{E}) = \prod_{e \in \mathcal{E}} w(e). \tag{5.59}$$

Hence, in this case the optimal choice for $\mathcal{E}$ is the spanning tree of $([n],\mathcal{C}^+)$ which has the maximum "value", i.e., the MVST. The following theorem transforms the problem of finding the MVST to the classic problem of finding the minimum spanning tree (MST).

Theorem 5.6.1. *Define $\mathcal{G} \triangleq ([n],\mathcal{C}^+)$. The following sets are identical:*

1. *The set of t-optimal subgraphs of $\mathcal{G}$ with $n-1$ edges weighted by w.*

2. *The set of MVSTs of $\mathcal{G}$ when edges are weighted by w.*

3. *The set of MSTs of $\mathcal{G}$ when edges are weighted by $w^* : e \mapsto -\log w(e)$.*

Proof. The equivalence between 1 and 2 was shown above. $\mathcal{T}$ is an MST of $\mathcal{G}$ iff the sum of its edge

weights $\sum_{e\in\mathcal{E}(\mathcal{T})} w^*(e)$ is minimum among all spanning trees of $\mathcal{G}$. This can be written as,

$$\sum_{e\in\mathcal{E}(\mathcal{T})} w^*(e) = -\sum_{e\in\mathcal{E}(\mathcal{T})} \log w(e) \tag{5.60}$$

$$= -\log \prod_{e\in\mathcal{E}(\mathcal{T})} w(e) \tag{5.61}$$

$$= -\log \mathbb{V}_w(\mathcal{T}). \tag{5.62}$$

Therefore, every MST must have maximum value among all spanning trees of $\mathcal{G}$ (with respect to w). □

According to Theorem 5.6.1, $(n-1)$-ESP$^+$ with $\mathcal{G}_{\text{init}} = ([n],\varnothing)$ can be solved (exactly) by finding an MST of $([n],\mathcal{C}^+)$ under a different weight function w^*. The classic problem of finding an MST is a well-studied problem in combinatorial optimization [32]. Two standard greedy algorithms for finding MST are due to Kruskal [101] and Prim [128]. Both algorithms find an exact solution in polynomial time. When $\mathcal{G}$ is sparse, Kruskal's algorithm is the preferred algorithm with a time complexity of $O(c\log n)$ [32].

5.6.2 Greedy Heuristics for Improper k-ESP$^+$

As demonstrated by Example 5.6.1, the performance guarantee provided for the greedy algorithm in Corollary 5.3.3 does not necessarily hold in improper instances of k-ESP$^+$. In what follows, we will describe two heuristic adaptations of the greedy algorithm (Algorithm 7) for dealing with improper instances of k-ESP$^+$.

Greedy-Greedy

The main idea behind this approach is to first, transform the given improper instance of k-ESP$^+$ into the "closest" proper instance of ESP$^+$, and then use the greedy algorithm in Algorithm 7 on the resulting proper ESP$^+$ for selecting the remaining number of candidates. Therefore, in the first stage of this method, our goal is to select the minimal set of edges for converting the disconnected base graph into a connected base graph. Hence, if the original base graph has n_c connected components, we need to select $n_c - 1$ edges from the candidates in the first stage. We select these edges such that the resulting graph has the maximum weighted number of spanning trees. This procedure is illustrated in Figure 5.5.

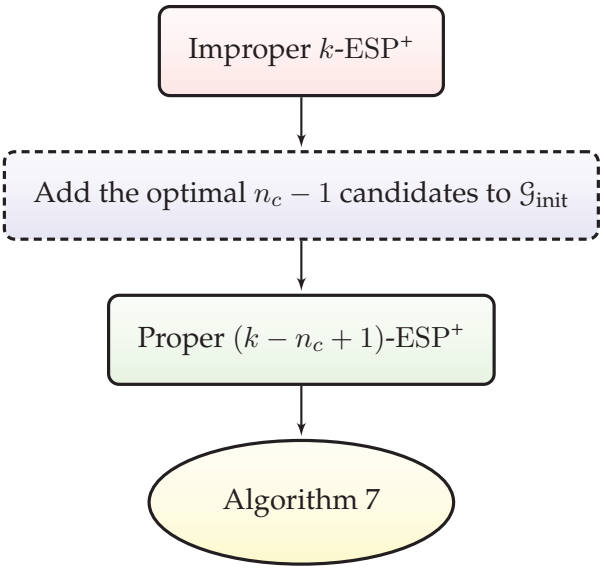


Figure 5.5: Illustration of the Greedy-Greedy algorithm.

Algorithm: Suppose an improper instance of k-ESP$^+$ with $\mathcal{G}_{\text{init}} = ([n],\mathcal{E}_{\text{init}})$ is given. Let n_c be the number of connected components if $\mathcal{G}_{\text{init}}$. Any instance falls into one of the following categories:

1. $k < n_c - 1$: Since at least $n_c - 1$ edges is needed to have a connected graph, this case is a degenerate instance of ESP$^+$ in which every design is optimal and $\mathsf{OPT} = 0$.

2. $k = n_c - 1$: If $n_c = n$, this case is an instance of MVST, for which exact solutions can be computed in polynomial time using the greedy MST algorithms (see Section 5.6.1). This technique can be extended to find an exact solution for arbitrary n_c. Let t_i be the weighted number of spanning trees in the ith connected component of $\mathcal{G}_{\text{init}}$ ($i \in [n_c]$). Consider the scenario in which we connect the components of the graph to each other via exactly $n_c - 1$ bridges. Note that choosing any other design would result in a disconnected graph. Hence, we just need to find the optimal choice for the $n_c - 1$ bridges (each bridge connects exactly one pair of connected components). Let w_j be the weight of the jth bridge ($j \in [n_c - 1]$). Then, the weighted number of spanning trees of the resulting graph can be computed as $\prod_{i\in[n_c]} t_i \prod_{j\in[n_c-1]} w_j$. Since t_i's are not affected by the choice of bridges ($i \in [n_c]$), this term can be maximized by maximizing $\prod_{j\in[n_c-1]} w_j$. In what follows, we show that this problem can be solved by finding a MVST. Let $\mathcal{G}$ be the graph whose vertices correspond to the connected components of $\mathcal{G}_{\text{init}}$. More precisely, $v_i \in \mathcal{V}(\mathcal{G})$ represents $\{j \in [n] : j \in \mathcal{V}_i\}$ in which $\mathcal{V}_i \subseteq [n]$ is the vertices in the ith connected component of $\mathcal{G}_{\text{init}}$. For each $\{r,s\} \in \mathcal{C}^+$ where $r \in \mathcal{V}_i$ and $s \in \mathcal{V}_j$ ($i \neq j$), there is an edge between v_i and v_j in $\mathcal{G}$ (we may have parallel edges). First, we can eliminate parallel edges by keeping only the one with the largest weight and discarding the rest (for the parallel edges be-

tween v_i and v_j). Then, the tree-connectivity can be maximized by finding the MSVT in $\mathcal{G}$ (see Section 5.6.1). Adding the edges in MSVT to the base graph $\mathcal{G}_{\text{init}}$ results in a connected graph with the maximum tree-connectivity achievable.

3. $k > n_c - 1$: A greedy approach would be:

   (a) Solve an $(n_c - 1)$-ESP$^+$ problem using the procedure described above for the case of $k = n_c - 1$.

   (b) Add the edges selected by following the procedure above to the base graph $\mathcal{G}_{\text{init}}$, and solve a *proper* $(k - n_c + 1)$-ESP$^+$ with the remaining set of candidate edges using the greedy algorithm (Algorithm 7).

Example 5.6.2. An example is shown in Figure 5.6. Figure 5.6a illustrates an improper instance of 6-ESP$^+$ with $\mathcal{G}_{\text{init}} = (\mathcal{V},\varnothing)$. The dashed edges here correspond to the candidate edges. The greedy-greedy method described above tackles this problem by first constructing the MVST as illustrated in Figure 5.6b. After finding the MVST, we use it as the base graph for selecting the remaining number of candidates using the greedy algorithm (Algorithm 7).

Corollary 5.6.2 follows from Corollary 5.3.3 and the fact that the Greedy-Greedy approach initiates the greedy algorithm (Algorithm 7) from the MVST.

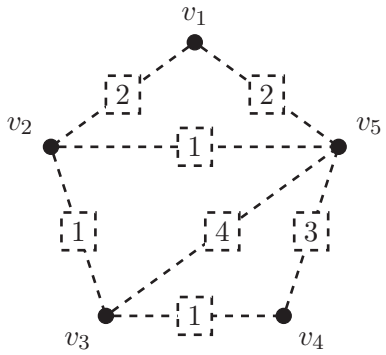
Corollary 5.6.2. *Let* $\mathsf{OPT}_{\mathsf{mvst}}$ *be the maximum tree-connectivity achievable by starting from the MVST (as described above) and selecting* $k - n_c + 1$ *edges using the greedy algorithm (Algorithm 7). Let* $\tau^\star_{\mathsf{mvst}}$ *be the tree-connectivity of the MVST. And finally, let* $\tau_{g\text{-}g}$ *be the tree-connectivity achieved by the above Greedy-Greedy algorithm. Then,*

$$\eta \cdot \mathsf{OPT}_{\mathsf{mvst}} + 1/e \cdot \tau^\star_{\mathsf{mvst}} \leq \tau_{\text{g-g}} \leq \mathsf{OPT}_{\mathsf{mvst}} \leq \mathsf{OPT}, \tag{5.63}$$

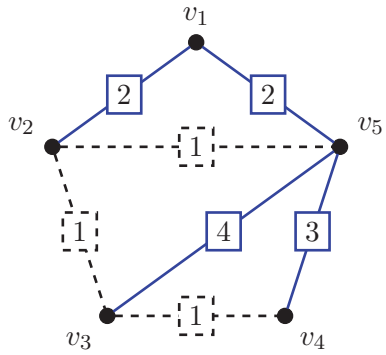
where $\eta \triangleq 1 - 1/e \approx 0.63$.

The ϵ-trick

Another simple heuristic technique for dealing with an improper instance of k-ESP$^+$ is to convert it into a proper instance of ESP$^+$ by borrowing extra "ϵ-edges". This approach is inspired by a similar idea proposed by Shamaiah et al. [136] in a non-graphical context. These edges are weighted by ϵ for some small value of ϵ. The ϵ-edges are chosen such that adding them to the base graph results



(a) An improper 6-ESP$^+$ with $\mathcal{E}_{\text{init}} = \varnothing$



(b) A proper 2-ESP$^+$ using the MVST as $\mathcal{G}_{\text{init}}$

Figure 5.6: Greedy-Greedy.

in a connected base graph. Therefore, upon adding the ϵ-edges to $\mathcal{G}_{\text{init}}$, we can apply the greedy algorithm (Algorithm 7) to chose k edges. The intuitive principle behind this technique is that when ϵ is sufficiently small, the impact of ϵ-edges on the decisions made by the greedy algorithm will be negligible. However, we also note that Algorithm 7 may become numerically unstable when ϵ is too small. After finding a suboptimal solution, we then remove the ϵ-edges from the graph. This process is illustrated in Figure 5.7.

The only remaining question is how the ϵ edges are selected? One option is to choose the minimal number of ϵ-edges (i.e., $n_c - 1$ if $\mathcal{G}_{\text{init}}$ has n_c connected components). Hence, in this case the ϵ-edges depend on $\mathcal{G}_{\text{init}}$. Alternatively, we can always add an ϵ-spanning tree to $\mathcal{G}_{\text{init}}$ to guarantee the connectivity of the resulting base graph. Several examples are shown in Figure 5.8.

5.7 Applications

The results presented in Chapter 4 demonstrate that the problem of designing D-optimal SLAM problems boils down to designing graphs with the maximum weighted number of spanning trees. This led to the study of the t-optimal graph synthesis problem in this chapter. Finally in this section, we discuss how our near-optimal graph synthesis frameworks can be used in several robotic applications, including SLAM.

5.7.1 Near-D-Optimal Measurement Selection

We begin by investigating the measurement selection problem in SLAM. As discussed in Chapter 4, maintaining sparsity is critical in SLAM as it determines the computational efforts required for solving the inference problem. If the underlying graph is dense, finding the maximum likelihood esti-

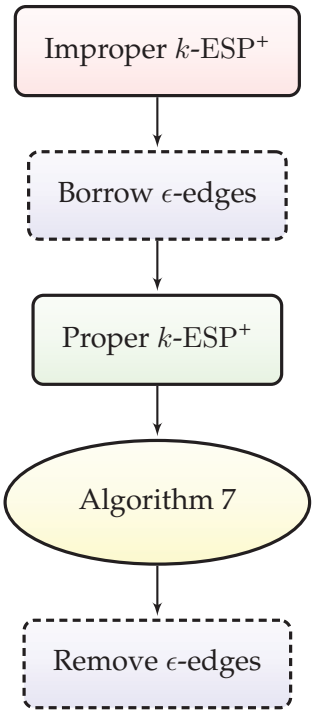


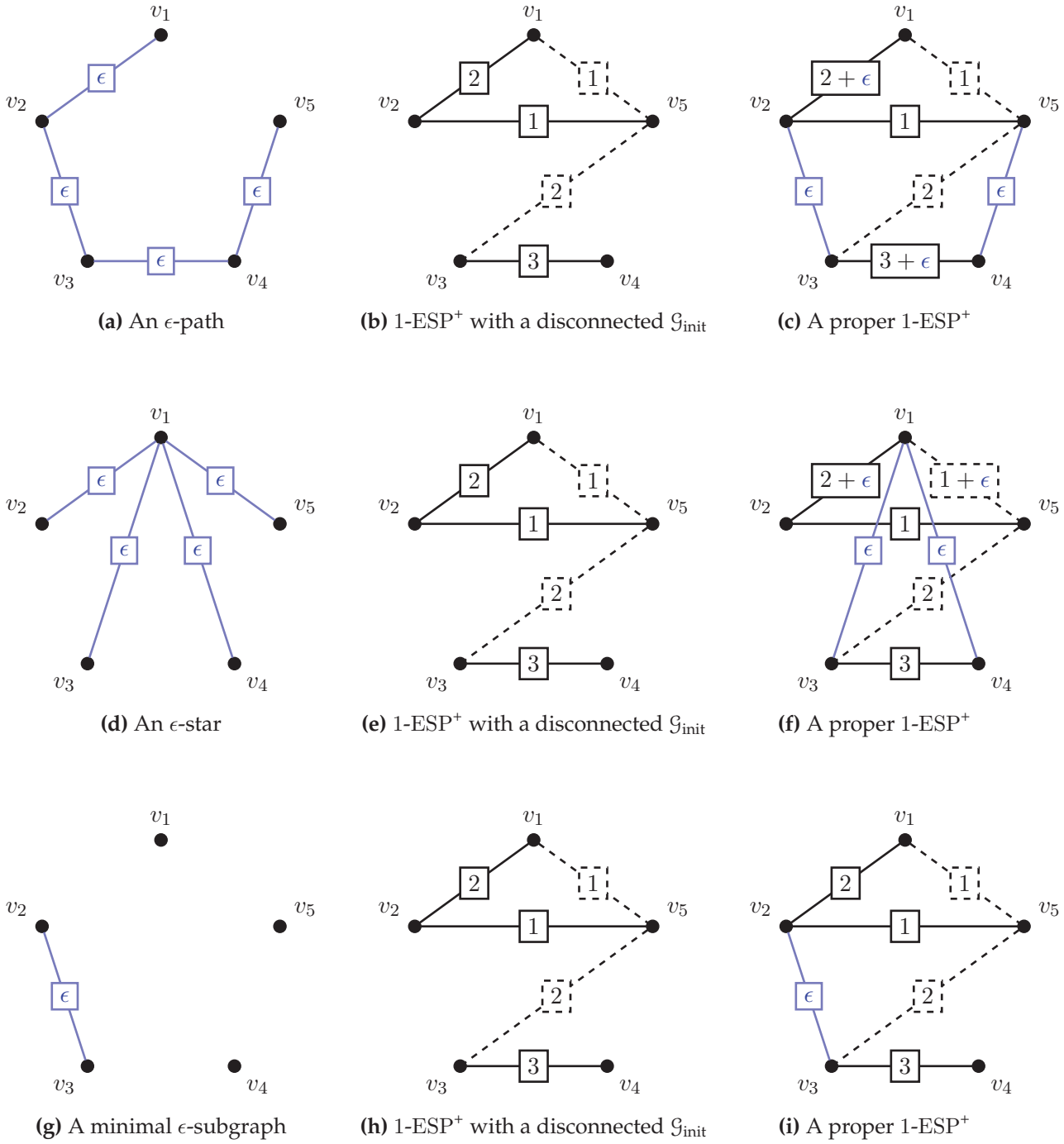
Figure 5.7: Illustration of the ϵ-trick.

mate in SLAM requires $O(n^3)$ time and space. This cost can be significantly reduced if the underlying graph is sparse. Maintaining sparsity is especially crucial in "long-term" missions, where n (i.e., number of robot poses) can easily be as large as $10^4 - 10^5$. Sparsity can be preserved by designing a measurement selection process to asses the significance of new or existing measurements. Such a vetting process can be implemented by:

(i) Assessing the quality any new measurement before adding it to the graph, and/or

(ii) Discarding a subset of the existing measurements if their contribution is deemed to be insufficient.

Many measurement selection and sparsification schemes have been proposed in the literature. For example, Joshi and Boyd [81] through convex relaxation, and Shamaiah et al. [136] through the greedy algorithm, have designed approximation algorithms for the problem of selecting sensors in linear-Gaussian systems. In the context of SLAM, Ila et al. [80] designed a vetting scheme based on the mutual information between a loop-closure measurement and the state. They demonstrated that this term can be computed in compact form, and noted that their proposed method can be used to avoid unnecessary costly scan-matching operations. Kretzschmar et al. [99, 100] designed a process for discarding laser scans based on approximating the mutual information between the scans and a grid map. Marginalizing out poses creates fill-in and therefore reduces the sparsity of the information



(a) An ϵ-path **(b)** 1-ESP$^+$ with a disconnected $\mathcal{G}_{\text{init}}$ **(c)** A proper 1-ESP$^+$

(d) An ϵ-star **(e)** 1-ESP$^+$ with a disconnected $\mathcal{G}_{\text{init}}$ **(f)** A proper 1-ESP$^+$

(g) A minimal ϵ-subgraph **(h)** 1-ESP$^+$ with a disconnected $\mathcal{G}_{\text{init}}$ **(i)** A proper 1-ESP$^+$

Figure 5.8: The ϵ-trick: augmenting a spanning tree.

matrix. Kretzschmar and Stachniss [99] maintain sparsity by marginalizing out the poses corresponding to the discarded laser scans using the Chow-Liu tree-based approximation [30]. This approach was later extended by Carlevaris-Bianco et al. [19]. Vial et al. [152] proposed a conservative sparsification method based on minimizing the Kullback-Leibler divergence between the original and the sparsified marginal distribution. Their method is designed to eliminate the fill-in created by pose marginalization and to achieve the desired sparsity pattern in a "consistent" manner. In a closely-

related work, Huang et al. [74] used lasso-like ℓ_1-regularized minimization to promote sparsity while minimizing the Kullback-Leibler divergence by a conservative approximation. More recently, Mazuran et al. [110] proposed a method in which a convex optimization problem is solved for approximating the dense marginal distribution by recovering nonlinear factors that are optimal with respect to the Kullback-Leibler divergence. Pose-graph sparsification based on our graph synthesis framework would be orthogonal to [19, 74, 110, 152], since we eliminate the edges (measurements) of the actual pose-graph, while such methods operate on the underlying Markov random field and prune, e.g., the fill-in created after marginalizing out a subset of nodes. Moreover, unlike [80, 81, 136], our approach does not rely on the linearization of the nonlinear measurement models. Finally, similar to [99, 100], we can discard loop-closure measurements by assessing their significance (ESP). However, note that we do so by examining the graph topology, while Kretzschmar et al. [99, 100] approximate the mutual information and conditional entropy for a specific setup (i.e., occupancy grid maps and laser scans).

The measurement selection problem is captured by our k-ESP$^+$ and k-ESP$^-$ t-optimal graph synthesis problems—as a result of Theorem 4.2.6. There is only one subtle difference between the two problems. In the synthesis problems studied so far, it was implicitly assumed that each edge is weighted by a single weight function. However, this is not necessarily the case in SLAM, where each relative pose measurement is comprised of two components (i.e., translational and rotational), each of which has its own precision, i.e., w_p and w_θ in Theorem 4.2.6. Thus we need to revisit the synthesis problem in a more general setting, where multiple weight functions assign weights, simultaneously, to a single edge.[7] Fortunately, our near-t-optimal graph synthesis framework and its analysis can be easily generalized to handle the expression that appears in Theorem 4.2.6 (i.e., log-determinant of the Fisher information matrix) with multiple weight functions.

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

$$\Psi \triangleq 2 \cdot \Phi_{w_p} + \Phi_{w_\theta} \tag{5.64}$$

which appears in Theorem 4.2.6. Note that Ψ is a linear combination of normalized monotone submodular functions with positive weights, and therefore is also normalized, monotone and submodular.

[7] It is important to note that this case (the expression in Theorem 4.2.6) is different from the case in which there are multiple (parallel) edges (with different weights) between two vertices.

2. **Convex Relaxation** : The convex relaxation algorithm can also be generalized by replacing the concave objective function $\log\det \mathbf{L}_w(\boldsymbol{\pi})$ with the following concave function,

$$2 \cdot \log\det \mathbf{L}_{w_p}(\boldsymbol{\pi}) + \log\det \mathbf{L}_{w_\theta}(\boldsymbol{\pi}) \tag{5.65}$$

which appears in Theorem 4.2.6.

5.7.2 D-Optimal Bootstrapping for SLAM

Under the additive Gaussian noise assumption, finding the ML (and MAP, with Gaussian priors) estimate requires solving a nonlinear least squares problem using an iterative solver. Iterative solvers need a reliable initial guess. Ideally, the initial guess should be:

(i) sufficiently close to the global minimum (ML/MAP estimate), and

(ii) easy to compute.

The process of finding a reliable initial guess for SLAM is called *bootstrapping*; see [73] for a survey of bootstrapping algorithms in SLAM. One of the most popular and effective bootstrapping techniques is due to Konolige et al. [98]. In [98], an initial guess is constructed by concatenating measurements along a spanning tree of the pose-graph. Fusing measurements along a spanning tree can be done efficiently, and therefore this method satisfies (ii). In order to satisfy (i) (i.e., reliability), Konolige et al. insightfully noted that the spanning tree should be chosen such that the accumulated uncertainty (i.e., error) is minimum. They proposed to use the shortest-path spanning tree rooted at the first pose (i.e., origin of the global frame). In [98], the graph is assumed to be unweighted; hence, the shortest-path spanning tree can be constructed by running a Breadth-first search (BFS) [32].

However, this is clearly not the best choice for the bootstrapping spanning tree, especially when measurements (edges) have different precisions (inverse of covariance). As a simple example, note that propagating measurements along a longer chain of precise measurements may result in a more reliable initial guess in comparison with the shortest chain whose components are relatively imprecise. To formalize this intuition, we can use Theorem 4.2.6 and find the D-optimal bootstrapping tree by finding the spanning tree with maximum weighted number of spanning trees—which in this case is equivalent to the most valuable spanning tree (MVST). In Section 5.6.1 we demonstrated how the exact MVST can be computed efficiently using the MST algorithms (e.g., Kruskal's algorithm needs $O(m\log(n))$ time). This procedure is summarized in Algorithm 11.

Algorithm 11 D-Optimal Spanning Tree Bootstrapping

1: Use the Kruskal's algorithm (see Section 5.6.1) to find $\mathcal{T}_\theta^\star$, i.e., the MVST when edge weights are assigned by w_θ.

2: Estimate $\hat{\boldsymbol{\theta}}_{\text{init}}$ by propagating rotational measurements along $\mathcal{T}_\theta^\star$.

3: Use the Kruskal's algorithm (see Section 5.6.1) to find $\mathcal{T}_p^\star$, i.e., the MVST when edge weights are assigned by w_p.

4: Estimate $\hat{\mathbf{p}}_{\text{init}}$ by propagating translational measurements along $\mathcal{T}_p^\star$ and by using $\hat{\boldsymbol{\theta}}_{\text{init}}$.

5: Return $\hat{\mathbf{x}}_{\text{init}} = (\hat{\mathbf{p}}_{\text{init}}, \hat{\boldsymbol{\theta}}_{\text{init}})$.

5.7.3 Network Reliability

We often need to design reliable networks that are made out of unreliable links. Designing reliable network topologies that remain connected under random link failure is one of the oldest applications of t-optimal graph synthesis. Network reliability theory has a rich history; see [11, 13, 116] for comprehensive surveys of some of the key results discovered in the last three decades. Some of the key results in characterizing t-optimal graphs among $\mathbb{G}_{n,m}$ are due to the network reliability theory community (see Section 5.1.1). Intuitively speaking, this should not be surprising, since by definition a network is connected if and only if it has at least one operational spanning tree; hence, it is reasonable to expect that maximizing the number of spanning trees, in some instances, could somehow increase the probability of the network remaining connected.

In this section, we only consider a special model of network reliability. *All-terminal network reliability* is formally defined as the probability of a network being connected when edges fail independently with probability p. This widely-used model of reliability has obvious applications across different domains, including communication, sensor, power and robotic networks. For any undirected graph $\mathcal{G} \in \mathbb{G}_{n,m}$ and $0 \leq p \leq 1$, we use $\mathrm{Rel}(\mathcal{G},p)$ to denote this value. Let c_i denote the number of connected subgraphs of $\mathcal{G}$ with i edges. Then, all-terminal reliability can be written as,

$$\mathrm{Rel}(\mathcal{G},p) \triangleq \mathbb{P}\left[\text{ random graph remains connected }\right] \tag{5.66}$$

$$= \sum_{i=1}^{m} c_i \cdot (1-p)^i p^{m-i}. \tag{5.67}$$

Since the minimal number of edges needed for having a connected subgraph is $n-1$, $c_i = 0$ for

$i \in [n-2]$. Hence, (5.67) is equal to,

$$\mathrm{Rel}(\mathcal{G},p) = \sum_{i=n-1}^{m} c_i \cdot (1-p)^i p^{m-i}. \tag{5.68}$$

By definition $c_{n-1} = t(\mathcal{G})$. An important negative result is due to Provan and Ball [129] who showed that computing $\mathrm{Rel}(\mathcal{G},p)$ is NP-hard (more accurately, #P-complete). Consequently, it is sensible to resort to approximation (e.g., through Monte Carlo sampling) and/or search for reasonable lower and upper bounds. An easy observation is that when $p \approx 1$ (i.e., when links are sufficiently unreliable) we have,

$$\mathrm{Rel}(\mathcal{G},p) \approx c_{n-1} \cdot (1-p)^{n-1} p^{m-n+1} \qquad (p \to 1) \tag{5.69}$$

$$= t(\mathcal{G}) \cdot (1-p)^{n-1} p^{m-n+1}. \tag{5.70}$$

When p, m and n are fixed, (5.70) is proportional to the number of spanning trees in $\mathcal{G}$. Hence, in case of (a fixed) $p \approx 1$, t-optimal graphs with respect to $\mathbb{G}_{n,m}$ are also the most reliable graphs. A similar analysis can be done for the case of $p \to 0$ (i.e., very reliable links); however, we do not discuss it here as it is not directly related to the number of spanning trees (see [9, 13, 116]).

For any fixed p, since $\mathbb{G}_{n,m}$ is finite, there exists a graph $\mathcal{G}$ whose reliability is greater than (or equal to) any other graph. A rather unintuitive discovery due to Kelmans [87] is that there exists $\mathcal{G}_1,\mathcal{G}_2 \in \mathbb{G}_{n,m}$ and $p,q \in [0,1]$ such that $\mathrm{Re}(\mathcal{G}_1,p) > \mathrm{Re}(\mathcal{G}_2,p)$ while $\mathrm{Re}(\mathcal{G}_1,q) < \mathrm{Re}(\mathcal{G}_2,q)$. A simple example of this phenomenon in $\mathbb{G}_{6,8}$ is discovered by Boesch [12, Figure 2]. This naturally leads to the following question: is there always a topology $\mathcal{G} \in \mathbb{G}_{n,m}$ whose reliability is greater than or equal to any other graph in $\mathbb{G}_{n,m}$ *for any* p? Such graphs are called *uniformly-most reliable* (or *uniformly optimally reliable*) with respect to $\mathbb{G}_{n,m}$. As we noted earlier, when $p \to 1$, t-optimal graphs give the most-reliable topologies. Hence, it immediately follows that a necessary condition for a graph to be uniformly-most reliable is to be t-optimal. Uniformly-most reliable graphs have been identified for several special cases; see [13] for a recent survey. Interestingly, it has been shown by Kelmans [88]—and later independently by Myrvold et al. [117]—that uniformly-most reliable graphs do *not* always exist. This phenomenon was demonstrated using an infinite family of counterexamples in which there exists a graph that is most reliable for $p \approx 0$ without being t-optimal.

So far we showed that the number of spanning trees plays a key role in maximizing reliability according to the $(\mathcal{G},p)$-model (for $p \to 1$ and in uniformly-most reliable networks). Hence, our near-t-optimal graph synthesis framework can be readily used for designing reliable network topologies.

Unfortunately, it is less clear how the $(\mathcal{G},p)$-model can be extended to multigraphs or the case in which edges may fail with different probabilities. Now suppose each edge e fails independently with probability p_e (and survives with probability $q_e \triangleq 1 - p_e$). The following scenarios show how our framework could be useful in this case.

- Consider the set of all spanning trees of $\mathcal{G}$, $\mathbb{T}(\mathcal{G})$. The probability of spanning tree $\mathcal{T} \in \mathbb{T}(\mathcal{G})$ being operational can be easily computed,

$$\mathbb{P}\left[\ \mathcal{T} \text{ is connected }\right] = \prod_{e \in \mathcal{E}(\mathcal{T})} q_e. \tag{5.71}$$

  Therefore, the most reliable spanning tree $\mathcal{T}^\star = \arg\max_{T \in \mathbb{T}(\mathcal{G})} \mathbb{P}\left[\ \mathcal{T} \text{ is connected }\right]$ can be obtained by finding the MVST of $\mathcal{G}$ when edge e is weighted by q_e. In Section 5.6.1, we showed that an exact solution to this problem can be obtained using the Kruskal's algorithm in $O(m\log(n))$ time.

- As we proved in Theorem 5.3.4, the expected number of spanning trees of $\mathcal{G}$ is equal to the weighted number of spanning trees when edges are weighted by the survival probabilities (i.e., q_e's). Hence, our synthesis framework can be used to design networks whose expected number of spanning trees is near-optimal.

5.7.4 Applications in Other Domains

The number of spanning trees also arises in several other domains. For instance, in Statistics, Cheng [29] noted that D-optimal incomplete block designs are associated to t-optimal concurrence graphs—see [3] for a survey. See also [16, 67] and [97] (and the references therein) for applications in, respectively, Chemistry and RNA modelling.

5.8 Experimental Results

In this section, we present numerical results on random graphs and a real pose-graph SLAM dataset. These experiments are designed to evaluate the proposed near-t-optimal graph synthesis framework developed for k-ESP$^+$ and the measurement selection problem in SLAM. We implemented our algorithms in MATLAB. Problem P_2 is modelled using YALMIP [107] and solved using SDPT3 [150]. We have used the deterministic rounding algorithm (i.e., via sorting) for rounding the fractional solution of the convex program.

5.8.1 Random Graphs

Figure 5.9 illustrates the performance of our approximate solutions to k-ESP$^+$ in randomly generated graphs. The set of candidates in these experiments is $\mathcal{C}^+ = \binom{[n]}{2} \setminus \mathcal{E}_{\text{init}}$. Figures 5.9a and 5.9b depict the resulting tree-connectivity as a function of the number of randomly generated edges for a fixed $k = 5$ and, respectively, $n = 20$ and $n = 50$. The results indicate that our approximation algorithms exhibit comparable near-optimal performances for $k = 5$. Note that computing OPT by exhaustive search is only practical in small instances such as Figure 5.9a. Nevertheless, computing the exact OPT is not crucial for evaluating our approximate algorithms, as Corollary 5.3.11 guarantees that OPT $\in [\tau^\star_{\text{greedy}}, \tau^\star_{\text{cvx}}]$; i.e., the space between each black $\cdot$ and the corresponding green $\times$. Figure 5.9c shows the results obtained for a varying k. The optimality gap for τ_{cvx} gradually grows as the planning horizon k increases. It is evident from Figure 5.9c that the greedy algorithm outperforms the convex relaxation in longer horizons (say, $k \geq 20$) and attains near-t-optimal designs.

5.8.2 Real Pose-Graph SLAM Dataset

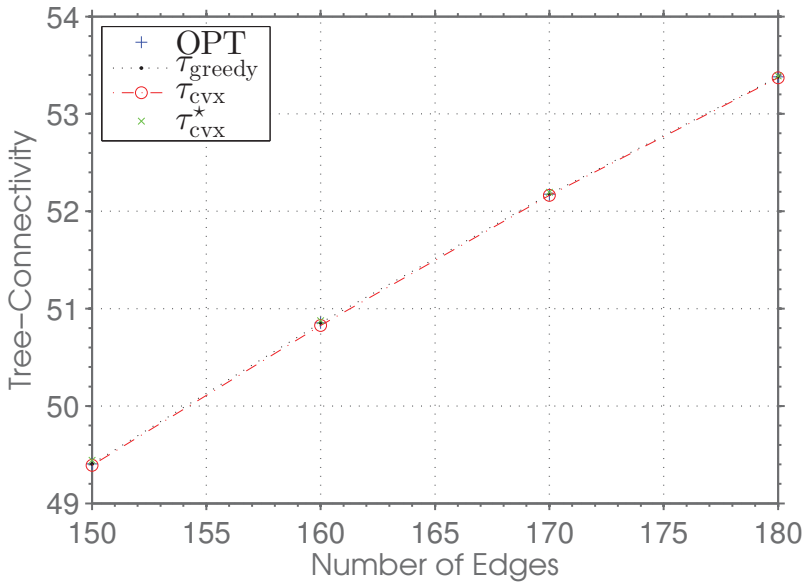
We evaluated the proposed algorithms on the Intel Research Lab dataset as a popular pose-graph SLAM benchmark.[8] In this scenario, $\mathcal{E}_{\text{init}}$ and $\mathcal{C}^+$ are the sets of odometry and loop-closure edges, respectively. The parameters in this graph are $n = 943$, $|\mathcal{E}_{\text{init}}| = 942$ and $|\mathcal{C}^+| = 895$. Obviously computing the true OPT via exhaustive search is intractable; e.g., for $k = 100$, there are more than 10^{134} possible graphs. For the edge weights, we are using the original information (precision) matrices provided in the dataset. Since the translational and rotational measurements have different precisions, two weight functions—w_p and w_θ—assign weights to each edge of the graph, and the objective is to maximize $\lim_{\gamma \to 0^+} -\log\det \mathbb{C}\text{ov}[\mathbf{x}^\star] = 2 \cdot \tau_{w_p}(\mathcal{G}) + \tau_{w_\theta}(\mathcal{G})$. Figure 5.10 depicts the resulting objective values for the greedy and convex relaxation approximation algorithms, as well as the upper bounds ($\mathcal{U}$) in Corollary 5.3.11. According to Figure 5.10, both algorithms have attained near-t-optimal (near-D-optimal) designs. Once again, the greedy algorithm outperforms the convex relaxation (with deterministic rounding). For small values of k, the upper bound $\mathcal{U}$ on OPT is given by $\mathcal{U}_{\text{greedy}}$ (blue curve). However, for $k \geq 60$, the convex relaxation provides a significantly tighter upper bound on OPT (green curve). Figure 5.11 illustrates the evolution of the greedy design for six different values of k. Figure 5.11f shows the original dataset with 895 loop closures. A video is

[8] We would like to thank the community for collecting, preprocessing and publishing this dataset. The original dataset is collected by Dirk Hähnel. We extracted the graph from the preprocessed version provided by the g2o team [102] at the University of Freiburg.
`https://svn.openslam.org/data/svn/g2o/trunk/data/2d/intel/intel.g2o`

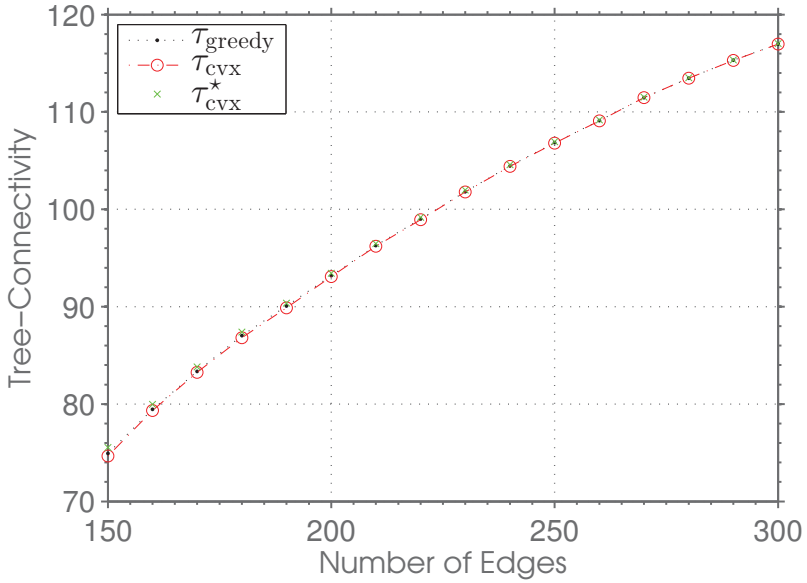
available at `https://youtu.be/5JZF2QiRbDE`.

5.9 Summary

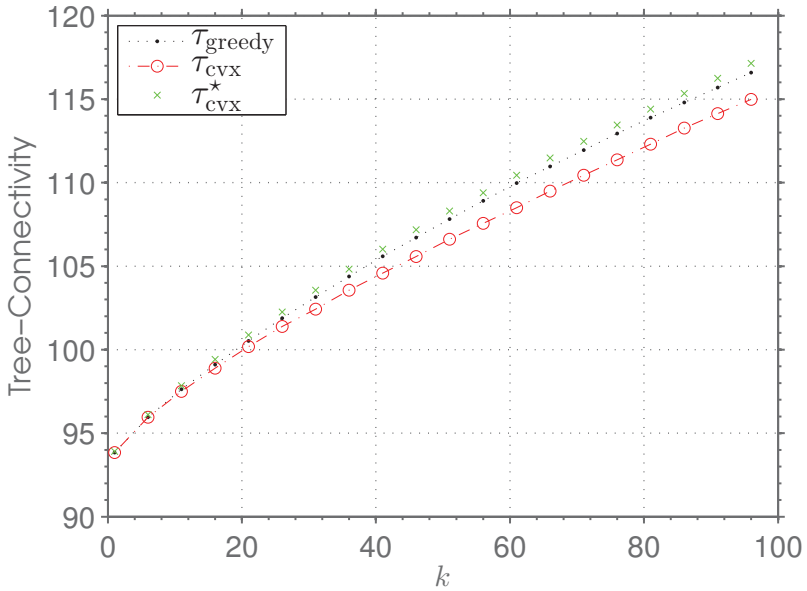
We studied the problem of designing near-t-optimal graphs under several types of constraints and formulations. Several new structures were revealed and exploited to design efficient approximation algorithms. In particular, we proved that the weighted number of spanning trees in connected graphs can be posed as a monotone log-submodular function of the edge set. Our approximation algorithms can find near-optimal solutions with performance guarantees. They also provide *a posteriori* near-optimality certificates for arbitrary designs.



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



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



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

Figure 5.9: k-ESP^{+} on randomly generated graphs with $\mathcal{C}^{+} = \binom{[n]}{2} \setminus \mathcal{E}_{\text{init}}$.

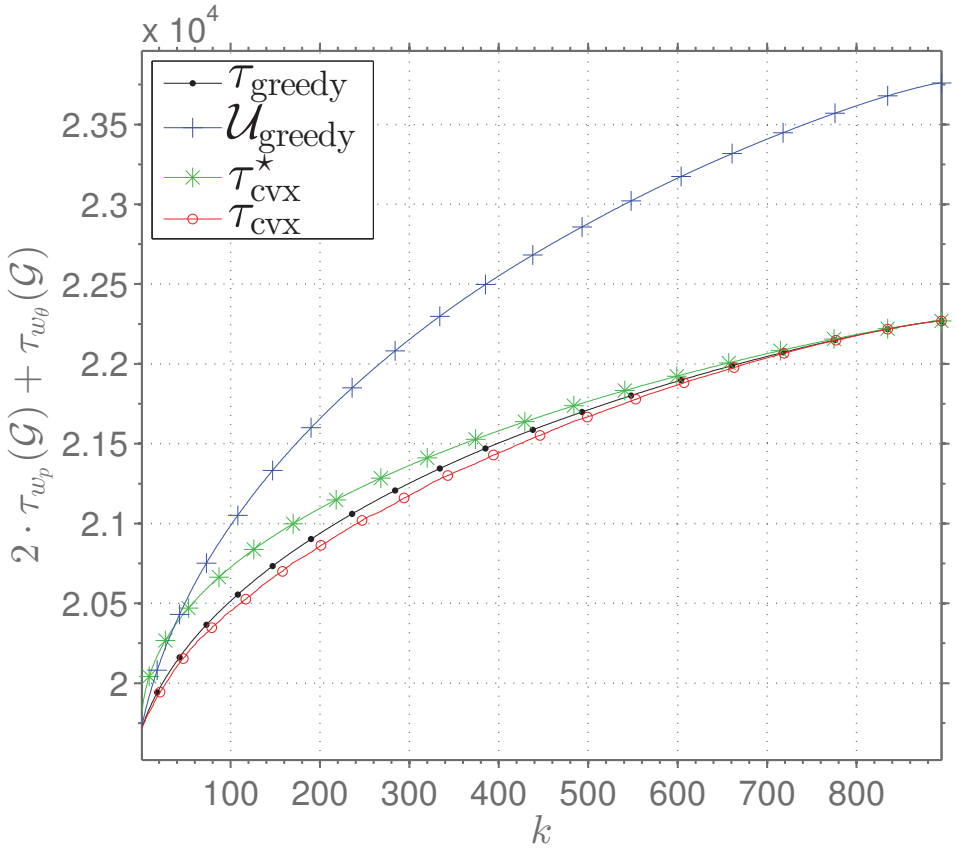
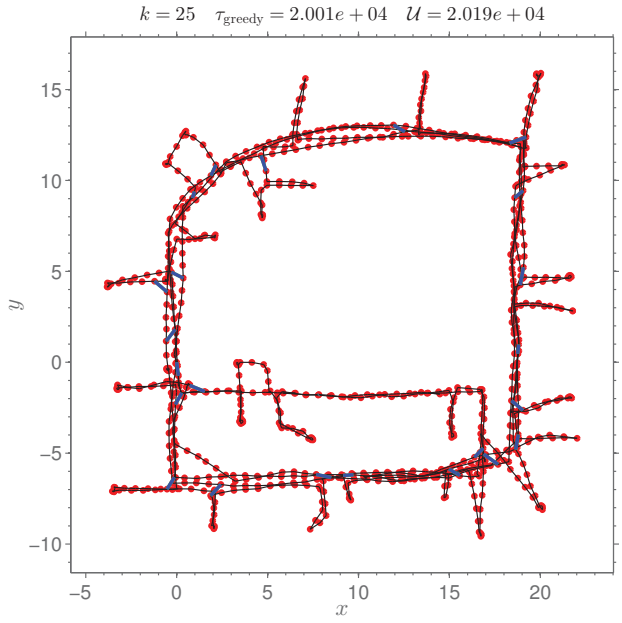
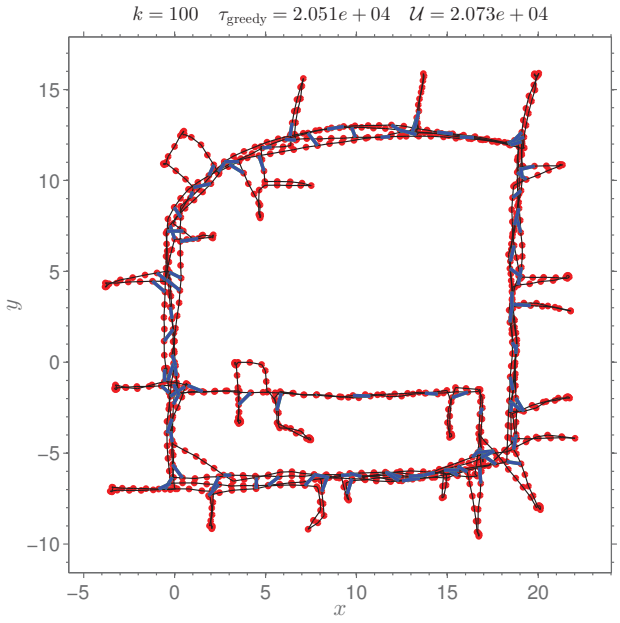


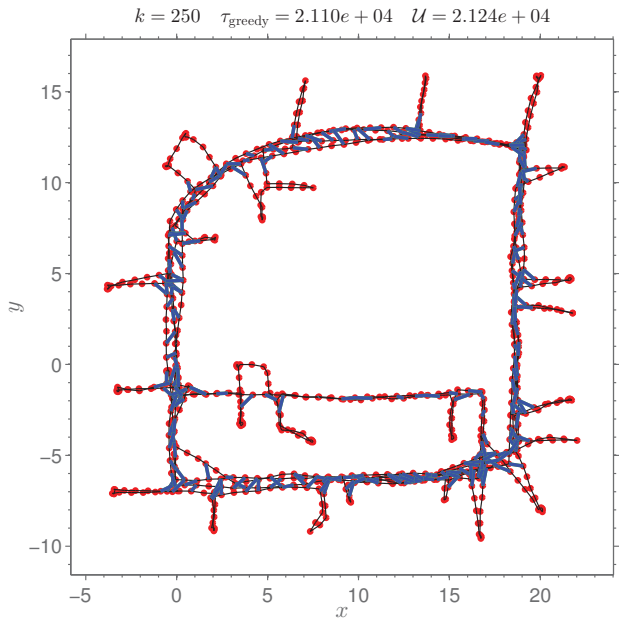
Figure 5.10: k-ESP$^+$ for pose-graph SLAM in the Intel Research Lab dataset.



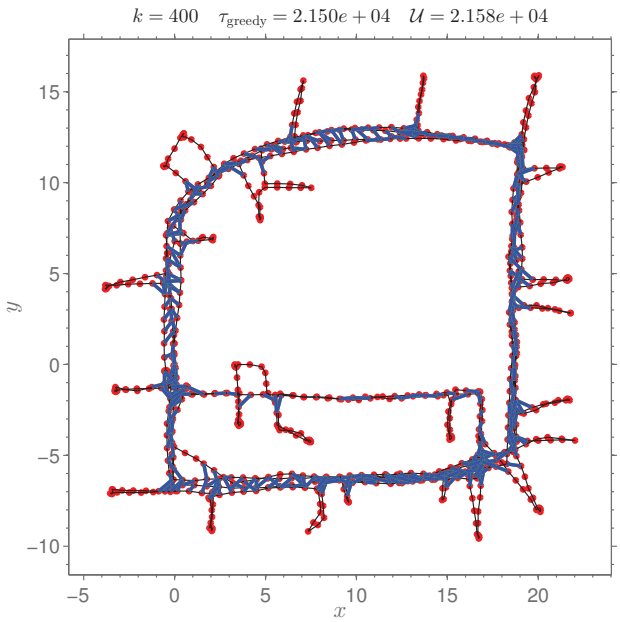
(a) $k = 25$



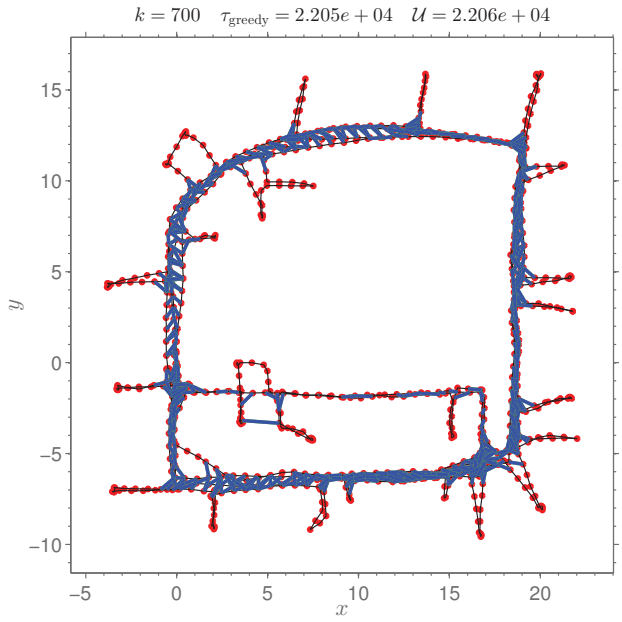
(b) $k = 100$



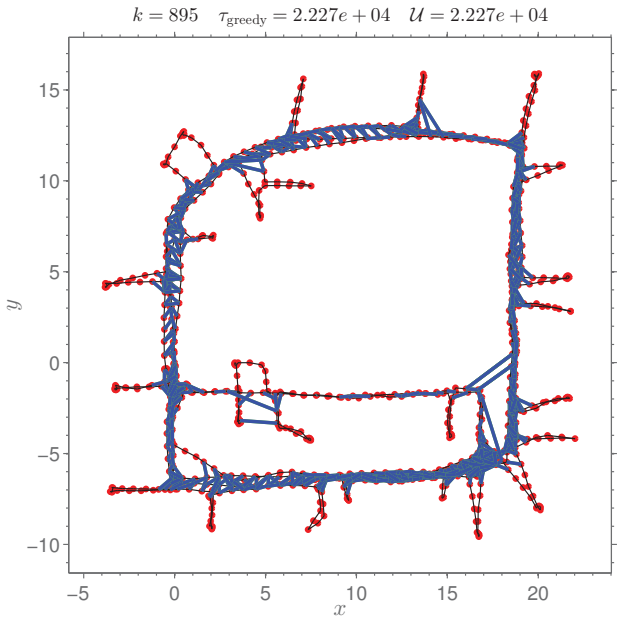
(c) $k = 250$



(d) $k = 400$



(e) $k = 700$



(f) $k = 895$

CHAPTER 6

Conclusion

In this thesis, we investigated two overlooked aspects SLAM, namely the separable structure of the negative log-likelihood cost function, and the impact of graph topology on the estimation error covariance of the maximum likelihood estimator. In this chapter, we briefly review our achievements and discuss future work.

6.1 SLAM: Sparse Separable Nonlinear Least Squares

In Chapter 3, we demonstrated that exploiting the separable structure of SLAM can improve the convergence speed of the state-of-the-art fast Newton-based solvers. Sparsity and separability are two structures that distinguish SLAM from other generic nonlinear least squares problems. Our algorithm exploits both structures simultaneously by computing the conditionally-optimal estimate for the "linear variables", while preserving the sparse structure of the normal equations. Intuitively, by taking advantage of the separable structure of SLAM, we gain a *global perspective* that can be used to guide the conventional local search techniques towards a local minimum of the problem.

Concluding Remarks

- Establishing a link to the rich literature on separable nonlinear least squares [61] is one of our key contributions. In particular, recognizing the equivalence between Algorithm 1 and Algorithm 2 was the missing link that enabled us to retain sparsity while exploiting the separable structure via Algorithm 3. This link also provides a solid theoretical justification for the proposed algorithm.

- The proposed algorithm can be applied to some of the most common forms of SLAM (i.e.,

2D/3D feature-based and pose-graphs) without any restrictive assumption on the structure of the noise covariance matrix. Furthermore, our approach is compatible with any nonlinear least squares solver and the benefits it brings along are orthogonal to those of other potential improvements such as using a more efficient implementation, alternative Newton-based solvers, trust-region methods or line search techniques.

- Exploiting separability is especially beneficial when Gauss-Newton iterations are relatively costly, or when it takes more than a few iterations to solve the full nonlinear least squares problem. High noise regimes and poorly initialized problems are likely to be among such cases.

- Through the projection gain (Section 3.5), we quantify the impact of exploiting the separable structure of the problem. As we demonstrated in Chapter 3, the relative gain is typically maximum at the beginning, and gradually decays as we approach a local minimum. Incorporating this gain into our sparse variable algorithm makes it computationally "fail-safe" by preventing further unnecessary projection steps.

- Chapter 3 mainly focused on the batch solvers. Incrementally solving SLAM [84, 85] is more suitable for online applications. The separable structure of SLAM is preserved in incremental formulation of SLAM and can be exploited by the same principles and techniques introduced in Chapter 3. As we mentioned earlier, our results indicate that exploiting separability is mostly beneficial if the initial guess is not already "too close" to the solution. An important advantage of incrementally solving SLAM is that we can initialize the solver at time t using the ML estimate at time $t-1$. This initial guess is usually quite good, and hence, exploiting separability may not be useful in such scenarios. Nevertheless, this is not the case if the information acquired at time t leads to a substantial "correction" of the whole trajectory. This is the case, for example, when the robot closes a large loop after a fairly long period of exploration (i.e., mostly dead reckoning). Incremental solvers that are capable of exploiting separability can therefore benefit from its fast convergence in such scenarios.

  Recall that the projection gain γ will be "small" whenever we are "close" to the solution. Therefore, by incorporating the projection gain into the proposed algorithm, we can naturally identify such scenarios and benefit from exploiting separability, while avoiding pointless (costly) projections in other cases.

Future Work

- Our current implementation (integrated with g2o [102]) is still under development. As part of our future work, we plan to reach full compatibility and integration with g2o for every (separable) measurement model, linear solver and iterative optimization algorithms currently supported by g2o.

6.2 SLAM: Estimation over Graphs

Any instance of SLAM is uniquely characterized by the true "geometry of the scene" (e.g., actual trajectory in pose-graph SLAM), one or more probabilistic observation model(s) (e.g., a noise-free measurement function corrupted by an additive Gaussian noise), realized measurements, and a graph (i.e., an adjacency list) that encodes "who observed whom". Our goal in Chapter 4 was to understand the exact mechanisms through which graph connectivity influences the estimation quality in maximum likelihood estimation. As our first contribution in this part, we provided a theoretical explanation for Olson and Kaess [123]'s empirical observation regarding the impact of average degree.

We then noted that the Fisher information matrix in SLAM, Compass-SLAM and Diff-Nets can be expressed in terms of the weighted Laplacian matrix of the graph.[1] These two matrices represent the two facets of SLAM; i.e., estimation-theoretic and graph-theoretic. We then established two links between these two matrices. First, we showed that in Compass-SLAM and Diff-Nets, the inverse of algebraic connectivity is a lower bound for the worst-case estimation error variance (i.e., "hyper-diameter" of the confidence hyper-ellipsoid). Subsequently, we proved that the determinant of Fisher information matrix (and consequently, that of the estimation error covariance of the maximum likelihood estimator and Cramér-Rao lower bound) is closely related to the weighted number of spanning trees in the graph. This quantity is known as the D-criterion (determinant-criterion) [130]. This relation was exact in Compass-SLAM and Diff-Nets. For SLAM, this relation was characterized in two ways:

1. a pair of lower/upper bounds (that depend on the weighted number of spanning trees) in Proposition 4.2.5, and

2. an asymptotic result in Theorem 4.2.6 that followed from Proposition 4.2.5.

Through extensive numerical evaluation on various real and synthetic datasets we showed that The-

[1] This result is due to Barooah and Hespanha [8] and Carlone [20].

orem 4.2.6 approximately holds even in the non-asymptotic regime. It is important to note that, under mild assumptions, this result enables us to accurately predict the D-criterion *only by assessing the structure of the weighted graph* (i.e., without using any information about the geometry of the scene or the realized measurements).[2]

Our analysis in Chapter 4 allows us to design near-D-optimal pose-graph SLAM problems through designing weighted graphs with the maximum weighted number of spanning trees (also known as t-optimal graphs). Consequently, in Chapter 5, we tackled the combinatorial optimization problem of designing sparse t-optimal graphs. To the best of our knowledge, there is no known efficient algorithm for solving this problem in the general case. So we did the next best thing by designing a complementary pair of efficient approximation algorithms with provable guarantees and near-optimality certificates. In order to analyze the proposed algorithms, we established several new theoretical results. In particular, we proved that the weighted number of spanning trees, under mild conditions, is monotone log-submodular. To the best of our knowledge, this is a new result in graph theory. Our analysis exploited this structure to prove that the greedy algorithm is near-optimal based on the seminal work of Nemhauser et al. [118]. The second approximation algorithm proposed in Chapter 5 is inspired by the convex relaxation approach due to Joshi and Boyd [81]. We expressed the original combinatorial optimization problem as an integer program, and then relaxed the integrality constraints to obtain a convex relaxation. An approximate solution is then obtained by mapping the globally-optimal solution of the convex program into a feasible (integral) solution using a *rounding* procedure. Our analysis extends that of [81] by shedding light on the connection between the original and the relaxed problems. This was made possible by our newly established result on the expected weighted number of spanning trees in random graphs (and its extension to the generalized linear model used in [81], Theorem 5.3.10). We also briefly discussed randomized rounding procedures and provided a justification for the deterministic rounding procedure proposed by Joshi and Boyd [81]. Our approximation algorithms and their analyses were extended to two additional problem formulations. Finally, we evaluated the performance of the proposed approximation algorithms using random graphs and a real pose-graph SLAM dataset. Our results suggest that the approximate design achieved by the greedy algorithm often dominates that of our convex relaxation, while the near-optimality certificate provided by our convex relaxation is normally tighter.

[2]The weight assigned to each edge is the precision (inverse of variance) of the corresponding pairwise measurement.

Concluding Remarks

- By (4.46) and (4.47), we showed that a simple approximation can provide an explanation for the empirical observations made by Olson and Kaess [123]. As we noted earlier, the average degree only reflects minimal information about the connectivity of the graph (i.e., ratio between the number of edges and vertices), and thus is not useful for decision making or planning. That being said, (4.46) has a nice interpretation. Consider the negative log-likelihood objective function in pose-graph SLAM. According to (4.46), the relative gap between the objective value of the ground truth and the value of the maximum likelihood estimate (i.e., $(f^\circ - f^\star)/f^\star$), in expectation, evolves approximately according to n/m where m is the number of measurements and n is the number of robot poses. This result suggests that in order to tighten this relative gap by some constant factor, one has to increase the number of measurements roughly by the same factor.

- Theorem 4.2.6 has a nice connection to one of the classic results in SLAM. Dissanayake et al. [38] in their seminal work proved that the determinant of the covariance matrix of Kalman filter reduces as the robot makes new observations. This result can be extended to cover the specific formulation studied in this thesis. From this perspective, Theorem 4.2.6 and Proposition 4.2.5 explain how fast this phenomenon occurs in terms of the weighted number of spanning trees in the graph.[3]

- It is important to note that there is a one-to-many relationship between the set of all weighted graphs and the set of all possible pose-graphs. Theorem 4.2.6 and Proposition 4.2.5 show that the D-criterion is almost entirely characterized by the weighted number of spanning trees in the underlying graph. Therefore, instead of searching for the D-optimal pose-graph among the infinite set of all feasible pose-graphs, we can efficiently search for near-t-optimal topologies in the finite set of all feasible topologies (assuming the edge weights are fixed).

- In Chapter 5, we implicitly assumed that the graphical representation of SLAM is perfectly known. Clearly, this is not always the case in practice. For example, in pose-graph SLAM, front-end may express its confidence by returning a probability with each new potential loop-closure edge. Another type of structural uncertainty arises in planning, where we may expect a loop-closure edge between two future poses with some probability (e.g., based on the topology

[3] It is easy to see that parallel edges (i.e., multiple measurements between two landmarks) can be replaced by a single edge whose weight is equal to the sum of the weights of the parallel edges.

of the environment). These uncertainties can be naturally incorporated in our framework by simply multiplying the weight of each edge by the corresponding probability. We already know through Theorem 5.3.4 that this results in the *expected weighted number of spanning trees*.

- As we saw in Table 4.1, computing the D-criterion through computing the weighted number of spanning trees in the graph is robust to convergence failures (e.g., local minima) and linearization errors.

Future Work

- There is an intriguing overlap between the parameters that emerge from our analysis of the impact of graph topology on the estimation error covariance, and those of [20]. In particular, $\lambda_1(\mathbf{L}_w)$ and δ (defined in Theorem 4.2.6) have also appeared in the convergence analysis due to Carlone [20]. We plan to investigate this connection in our future work.

- Theorem 4.2.6 can be straightforwardly generalized to 2D feature-based SLAM problems. Extensive empirical observations suggest that our analysis can also be extended to 3D SLAM with $\mathrm{SE}(3)$ relative-pose measurements. We will consider this extension in our future work.

- Searching for new loop closures via scan matching is a costly operation. The front-end can benefit from our work by prioritizing scan matching with potential candidates based on the impact of the resulting edge on the weighted number of spanning trees. We plan to investigate this idea in our future work.

- Our near-t-optimal graph synthesis framework can be readily used in any context where designing sparse networks with strong tree-connectivity is desired. Several applications were mentioned in Section 5.7. In particular, we plan to investigate the potential applications of our results in network reliability theory [13].

- We plan to extend our analysis of the convex relaxation algorithm proposed in Chapter 5. In particular, it is not clear whether there are other deterministic/randomized rounding procedures that can outperform the deterministic rounding we used in this thesis or lead to a stronger analysis (e.g., a constant approximation factor).

- The results presented in Chapters 4 and 5 are only the first steps towards the use of graph topology in areas such as active SLAM and decision making. We believe that designing reliable

graph topologies has the potential to be incorporated into the planning process. We aim to thoroughly investigate this idea in our future work.

Appendices

APPENDIX A

Preliminaries

A.1 Linear Algebra

Lemma A.1.1 (Schur's Determinant Formula)**.** *If* $\mathbf{A}^{-1}$ *exists,*

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

Proof. See, e.g., [111]. □

Lemma A.1.2. *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. Due 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 $\mathbf{M}^{\frac{1}{2}}\mathbf{N}^{-1}\mathbf{M}^{\frac{1}{2}}$,[1] 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. □

Lemma A.1.3 (Matrix Determinant Lemma)**.** *For any non-singular* $\mathbf{M} \in \mathbb{R}^{n\times n}$ *and* $\mathbf{c}$,$\mathbf{d} \in \mathbb{R}^n$,

$$\det(\mathbf{M} + \mathbf{c}\mathbf{d}^\top) = (1 + \mathbf{d}^\top\mathbf{M}^{-1}\mathbf{c}) \det \mathbf{M}. \tag{A.2}$$

Proof. See e.g., [111]. □

[1] Recall that $\mathbf{MN}$ and $\mathbf{NM}$ have the same spectrum.

Lemma A.1.4. *For any two* $\mathbf{N}$,$\mathbf{M} \in \mathbb{S}^n_{\geq 0}$ *we have*

$$\det(\mathbf{M}+\mathbf{N}) \geq \det(\mathbf{M}). \tag{A.3}$$

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

$$\det(\mathbf{M}+\mathbf{N}) = \det(\mathbf{M})\det(\mathbf{I}+\mathbf{M}^{-\frac{1}{2}}\mathbf{N}\mathbf{M}^{-\frac{1}{2}})) \tag{A.4}$$

$$= \det(\mathbf{M})\prod_{i=1}^{n}(1+\underbrace{\lambda_i(\mathbf{M}^{-\frac{1}{2}}\mathbf{N}\mathbf{M}^{-\frac{1}{2}})}_{\geq 0}) \tag{A.5}$$

$$\geq \det(\mathbf{M}). \tag{A.6}$$

□

A.2 Estimation Theory

Definition A.2.1 (Unbiased Estimator)**.** An estimator $\hat{\mathbf{x}}$ of parameter $\mathbf{x}$ is called an *unbiased* estimator iff $\mathbb{E}[\hat{\mathbf{x}}] = \mathbf{x}$.

Theorem A.2.1 (Cramér-Rao Lower Bound (CRLB))**.** *Under some regularity conditions [140], 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 Fisher information matrix (FIM),*

$$\mathbb{I}(\mathbf{x}) \triangleq \mathbb{E}_{\mathbf{z}}\,[\frac{\partial}{\partial \mathbf{x}}\log p(\mathbf{z};\mathbf{x})\ \frac{\partial^{\top}}{\partial \mathbf{x}}\log p(\mathbf{z};\mathbf{x})]. \tag{A.7}$$

Here the expectation is over $\mathbf{z}$ *and with respect to* $p(\mathbf{z};\mathbf{x})$*. Note that 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* $\mathbf{z}$*.*

Definition A.2.2 (Efficient Estimator)**.** An unbiased estimator that achieves CRLB is called an *efficient estimator*.

Corollary A.2.2. *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 Discrete Optimization

Suppose $\mathcal{W}$ is a finite *ground set*. Let $\xi : 2^{\mathcal{W}} \to \mathbb{R}$ be a real set function defined over the power set of $\mathcal{W}$. Finally, let $\mathcal{F} \subseteq 2^{\mathcal{W}}$ be the set of feasible subsets of $\mathcal{W}$. Then our goal is to solve the following optimization problem,

$$\begin{aligned} &\underset{\mathcal{A}\subseteq\mathcal{W}}{\text{maximize}} \quad \xi(\mathcal{A}) \\ &\text{subject to} \quad \mathcal{A} \in \mathcal{F}. \end{aligned} \tag{A.8}$$

- **Cardinality constraint**: The feasible set in this case is defined as

$$\mathcal{F}_{\text{card}} \triangleq \Big\{\mathcal{M} \subseteq \mathcal{W} : |\mathcal{M}| \leq k\Big\} \tag{A.9}$$

  for some $k \in \mathbb{N}$.

- **Matroid constraint**: The cardinality constraint is a special case of a more general class of constraints known as *matroid* constraints. More specifically, $(\mathcal{W},\mathcal{F}_{\text{card}})$ is a *uniform matroid* of rank k. Uniform matroids are the simplest kind of matroids. A more interesting example is the *partition matroid*.

Definition A.3.1 (Partition Matroid)**.** Let $\mathcal{W}_1, \ldots, \mathcal{W}_\ell$ be a *partition* for $\mathcal{W}$. Assign an integer (budget) $0 \leq k_i \leq |\mathcal{W}_i|$ to each $\mathcal{W}_i$. Define

$$\mathcal{I}_{\text{p}} \triangleq \Big\{\mathcal{M} \subseteq \mathcal{W} \,:\, |\mathcal{M} \cap \mathcal{W}_i| \leq k_i \;\text{ for } i \in [\ell]\Big\}.$$

The pair $(\mathcal{W},\mathcal{I}_{\text{p}})$ is called a *partition matroid*.

Definition A.3.2 (Set Function Properties)**.** Suppose $\mathcal{W}$ is a finite ground set. For any set function $\xi : 2^{\mathcal{W}} \to \mathbb{R}$,

1. ξ is called *normalized* if and only if $\xi(\varnothing) = 0$.

2. ξ is called *monotone* if $\xi(\mathcal{B}) \geq \xi(\mathcal{A})$ for every $\mathcal{A}$ and $\mathcal{B}$ s.t. $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{W}$.

3. ξ is called *submodular* if and only if for every $\mathcal{A}$ and $\mathcal{B}$ s.t. $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{W}$ and $\forall s \in \mathcal{W} \setminus \mathcal{B}$ we have,

$$\xi(\mathcal{A} \cup \{s\}) - \xi(\mathcal{A}) \geq \xi(\mathcal{B} \cup \{s\}) - \xi(\mathcal{B}). \tag{A.10}$$

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

5. ξ is called *modular* if and only if it is both submodular and supermodular

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

An equivalent condition for submodularity is as follows. ξ is submodular if and only if for every $\mathcal{A}$,$\mathcal{B} \subseteq \mathcal{W}$,

$$\xi(\mathcal{A} \cup \mathcal{B}) + \xi(\mathcal{A} \cap \mathcal{B}) \leq \xi(\mathcal{A}) + \xi(\mathcal{B}). \tag{A.11}$$

Furthermore, it is easy to show that if ξ is a normalized modular function, then we must have $\xi(\mathcal{A}) = \sum_{e \in \mathcal{A}} w(e)$ for some (weight) function $w : \mathcal{W} \to \mathbb{R}$.

Theorem A.3.1 (Nemhauser et al. [118])**.** *The greedy algorithm is a $(1 - 1/e)$-approximation algorithm for maximizing any normalized monotone submodular function subject to a cardinality constraint.*

Theorem A.3.2 introduces several operations under which both monotonicity and submodularity are preserved. Hence given any monotone submodular function, one can construct new monotone submodular functions by performing such operations.

Theorem A.3.2. *Both monotonicity and submodularity are preserved under the following operations.*

- ***Constant Addition**: For any constant $c \in \mathbb{R}$, $g : 2^{\mathcal{W}} \to \mathbb{R} : \mathcal{A} \mapsto f(\mathcal{A}) + c$ is monotone submodular if and only if $f : 2^{\mathcal{W}} \to \mathbb{R}$ is monotone submodular.*

- ***Constant Truncation**: For any constant $c \in \mathbb{R}$, $g : 2^{\mathcal{W}} \to \mathbb{R} : \mathcal{A} \mapsto \min\{f(\mathcal{A}),c\}$ is monotone submodular if and only if $f : 2^{\mathcal{W}} \to \mathbb{R}$ is monotone submodular.*

- ***Non-negative linear combination**: If $f_i : 2^{\mathcal{W}} \to \mathbb{R}$ 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.*

- ***Restriction**: If $f : 2^{\mathcal{W}} \to \mathbb{R}$ is monotone submodular, then for any $\mathcal{M} \subseteq \mathcal{W}$, $g : \mathcal{A} \mapsto f(\mathcal{A} \cap \mathcal{M})$ is monotone submodular.*

- ***Conditioning**: If $f : 2^{\mathcal{W}} \to \mathbb{R}$ is monotone submodular, then for any $\mathcal{M} \subseteq \mathcal{W}$, $g : \mathcal{A} \mapsto f(\mathcal{A} \cup \mathcal{M})$ is monotone submodular.*

A.4 Graph Theory

Definitions

Consider a *simple* undirected graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$. $\mathcal{V}$ is the set of vertices and $\mathcal{E} \subseteq \binom{\mathcal{V}}{2}$ is the edge set of $\mathcal{G}$. With a little abuse of notation, for any graph $\mathcal{G}$, $\mathcal{V}(\mathcal{G})$ and $\mathcal{E}(\mathcal{G})$ refer to, respectively, the vertex set and edge set of graph $\mathcal{G}$. Similarly, we define $n(\mathcal{G}) \triangleq |\mathcal{V}(\mathcal{G})|$ 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 $v \in \mathcal{V}$, $\mathcal{N}(v) \subseteq \mathcal{V}$ 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 $\vec{\mathcal{E}} \subseteq \mathcal{V} \times \mathcal{V}$. 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 $\mathcal{S} \subseteq \mathcal{V}$ and $\mathcal{F} \subseteq \mathcal{E}$. $\mathcal{H}$ 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

$$\mathcal{F} = \Big\{\{u,v\} : u,v \in \mathcal{S}, \{u,v\} \in \mathcal{E}\Big\}. \tag{A.12}$$

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 $n_{\mathrm{c}}(\mathcal{G})$. The *degree* of vertex $v \in \mathcal{V}$, $\deg(v)$, is defined as the number of edges connected to v, i.e., $\deg(v) \triangleq |\mathcal{N}(v)|$. 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 in which 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.

Lemma A.4.1 (Handshaking Lemma)**.** *For any simple undirected graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$,*

$$\sum_{v\in\mathcal{V}} \deg(v) = 2\,|\mathcal{E}|. \tag{A.13}$$

Proof. Note that in $\sum_{v\in\mathcal{V}}\deg(v)$, the edge $\{u,v\}\in\mathcal{E}$ is counted twice; once in $\deg(v)$ and once in $\deg(u)$. □

Graph Matrices

Any graph can be naturally represented by several matrices. These matrices play a key role in the algebraic and spectral graph theory where graphs are studied through such matrices and their spectra [59]. In this section, we briefly review some of these matrices. To simply our notation, we assume $n(\mathcal{G})=n$ and $m(\mathcal{G})=m$.

1. **Degree matrix**: The degree matrix $\mathbf{D}\in\mathbb{R}^{n\times n}$ is defined as $\mathbf{D}\triangleq\operatorname{diag}\left(\deg(v_1),\ldots,\deg(v_n)\right)$ where $\mathcal{V}(\mathcal{G})=\{v_1,\ldots,v_n\}$.

2. **Adjacency matrix**: The adjacency matrix $\mathbf{Adj}\in\{0,1\}^{n\times n}$ is defined as follows:

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

3. **Incidence matrix**: 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,

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

   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 the graph.

4. **Laplacian matrix**: The (graph) Laplacian (or, Kirchhoff) matrix $\mathbf{L}_\circ\in\mathbb{R}^{n\times n}$ is defined as $\mathbf{L}_\circ\triangleq\mathbf{A}_\circ\mathbf{A}_\circ^\top=\mathbf{D}-\mathbf{A}$

Deleting a row/column from the incidence and Laplacian matrices results in the following matrices:

- **Reduced incidence matrix**: 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.

- **Reduced Laplacian matrix**: 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. As mentioned above, the vertex associated to the deleted column and row is usually referred to as the *anchored* or *grounded* vertex. Note that it is possible to have multiple anchors. 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 A.4.1. *Let $\lambda_1(\mathbf{L}_\circ) \leq \lambda_2(\mathbf{L}_\circ) \leq \cdots \leq \lambda_n(\mathbf{L}_\circ)$ 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., $\lambda_i(\mathbf{L}_\circ) \geq 0$ for all $i \in [n]$.*
2. *The Laplacian matrix has at least one zero eigenvalue associated to the $\mathbf{1}_n$ eigenvector; i.e., $\mathbf{L}_\circ \mathbf{1}_n = 0$.*
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 A.4.2. *The* reduced incidence matrix $\mathbf{A}$ *is full row rank if and only if the corresponding graph is weakly connected.*

The following theorem directly follows Theorem A.4.2 and the fact that $\mathbf{L} = \mathbf{A}\mathbf{A}^\top$.

Corollary A.4.3. *The* reduced Laplacian matrix $\mathbf{L}$ *is positive definite if and only if the corresponding graph is connected.*

Theorem A.4.4 (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. $t(\mathcal{G}) = \frac{1}{n}\prod_{i=2}^{n} \lambda_i(\mathbf{L}_\circ)$.[2]

Theorem A.4.5 (Cayley's Formula)**.** *The number of spanning trees in $\mathcal{K}_n$ is given by $t(\mathcal{K}_n) = n^{n-2}$.*

Definition A.4.1 (Tree Value Function)**.** Suppose $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ is a weighted graph with a non-negative weight function. Let $\mathbb{T}_\mathcal{G}$ be the set of all spanning trees of $\mathcal{G}$. The *value* of each spanning

[2] Recall that the Laplacian matrix of any connected graph has a zero eigenvalue with multiplicity one (see, e.g., [59]).

tree of $\mathcal{G}$ is measured by the following function,

$$\mathbb{V}_w : \mathbb{T}_{\mathcal{G}} \to \mathbb{R}_{\geq 0} \tag{A.14}$$

$$\mathcal{T} \mapsto \prod_{e \in \mathcal{E}(\mathcal{T})} w(e). \tag{A.15}$$

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

Theorem A.4.6 (Weighted Matrix-Tree Theorem)**.** *For every simple weighted graph $\mathcal{G} = (\mathcal{V},\mathcal{E},w)$ with $w : \mathcal{E} \to \mathbb{R}_{>0}$ we have $t_w(\mathcal{G}) = \det \mathbf{AWA}^\top$ where $\mathbf{W} \triangleq \operatorname{diag}(w(e_1),\ldots,w(e_m))$.*

Theorem A.4.7. *Let $\mathcal{G}^+$ 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,*

$$t_w(\mathcal{G}^+) = t_w(\mathcal{G}) \cdot (1 + w_{uv}\Delta_{uv}^{\mathcal{G}}), \tag{A.16}$$

where $\Delta_{uv}^{\mathcal{G}} \triangleq \mathbf{a}_{uv}^\top \mathbf{L}^{-1} \mathbf{a}_{uv}$ is the effective resistance between u and v [58].

Proof. The reduced Laplacian matrix of $\mathcal{G}^+$ can be written as $\mathbf{L}^+ = \mathbf{L} + w_{uv}\, \mathbf{a}_{uv}{\mathbf{a}_{uv}}^\top$. Taking the determinant and applying Lemma A.1.3 concludes the proof. □

Remark 12. *In practice, we compute the effective resistance in sparse graphs using Algorithm 12.*

Lemma A.4.2. *Let $\mathcal{G}_1$ be a spanning subgraph of $\mathcal{G}_2$. For any $w : \mathcal{E}(K) \to \mathbb{R}_{\geq 0}$, $\mathbf{L}_{\mathcal{G}_2}^w \succeq \mathbf{L}_{\mathcal{G}_1}^w$ in which $\mathbf{L}_{\mathcal{G}}^w$ is the reduced weighted Laplacian matrix of $\mathcal{G}$ when its edges are weighted by w.*

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

$$\mathbf{L}_{\mathcal{G}_2}^w - \mathbf{L}_{\mathcal{G}_1}^w = \sum_{\{u,v\} \in \mathcal{E}(\mathcal{G}_2) \setminus \mathcal{E}(\mathcal{G}_1)} w_{uv}\ \mathbf{a}_{uv}\mathbf{a}_{uv}^\top \succeq \mathbf{0}. \tag{A.17}$$

□

Algorithm 12 Effective Resistance Between u and v

function Reff(u,v,$\mathbf{L}$) ▷ $u,v \in \mathcal{V} = \{0,1,\ldots,n-1\}$, $\mathbf{L}$: reduced Laplacian
  // Column of the reduced incidence matrix
  $\mathbf{a}_{uv} \leftarrow \mathbf{e}_u - \mathbf{e}_v$
  // Choose a fill-reducing permutation heuristic $\mathbf{P}$
  $\mathbf{P} \leftarrow$ COLAMD($\mathbf{L}$) ▷ e.g., column approximate minimum degree
  // Compute the sparse Cholesky factor $\mathbf{C}$ s.t. $\mathbf{PLP}^\top = \mathbf{CC}^\top$
  $\mathbf{C} \leftarrow$ SparseCholesky($\mathbf{PLP}^\top$)
  // solve $\mathbf{Cx}_{uv} = \mathbf{Pa}_{uv}$
  $\mathbf{x}_{uv} \leftarrow$ ForwardSolver($\mathbf{C}$,$\mathbf{Pa}_{uv}$) ▷ Lower Triangular
  $\Delta_{uv} \leftarrow \|\mathbf{x}_{uv}\|^2$
  return Δ_{uv}
end function

APPENDIX B

Proofs

B.1 Chapter 3

Proof of Theorem 3.3.1. The jth column of $\mathbf{J}_{\text{vp}}$ is given by

$$\begin{aligned}
[\mathbf{J}_{\text{vp}}]_{\cdot,j} = \frac{\partial \mathbf{r}_{\text{vp}}}{\partial \theta_j} &= \frac{\partial}{\partial \theta_j}[\mathbf{P}_{\boldsymbol{\theta}}^{\perp}\,(\tilde{\mathbf{z}} - \tilde{\mathbf{H}}_2\boldsymbol{\theta})] \\
&= \frac{\partial \mathbf{P}_{\boldsymbol{\theta}}^{\perp}}{\partial \theta_j}\,(\tilde{\mathbf{z}} - \tilde{\mathbf{H}}_2\boldsymbol{\theta}) \;+ \mathbf{P}_{\boldsymbol{\theta}}^{\perp}\frac{\partial(\tilde{\mathbf{z}} - \tilde{\mathbf{H}}_2\boldsymbol{\theta})}{\partial \theta_j} \\
&= \frac{\partial \mathbf{P}_{\boldsymbol{\theta}}^{\perp}}{\partial \theta_j}\,(\tilde{\mathbf{z}} - \tilde{\mathbf{H}}_2\boldsymbol{\theta}) \;- \mathbf{P}_{\boldsymbol{\theta}}^{\perp}[\tilde{\mathbf{H}}_2]_{\cdot,j} \\
&= -\frac{\partial \mathbf{P}_{\boldsymbol{\theta}}}{\partial \theta_j}\,(\tilde{\mathbf{z}} - \tilde{\mathbf{H}}_2\boldsymbol{\theta}) \;- \mathbf{P}_{\boldsymbol{\theta}}^{\perp}[\tilde{\mathbf{H}}_2]_{\cdot,j}
\end{aligned} \tag{B.1}$$

Now we only need to compute $\partial \mathbf{P}_{\boldsymbol{\theta}}/\partial \theta_j$. This term was first computed by Golub and Pereyra. We briefly mention their proof as stated in [135]. First note that $\mathbf{P}_{\boldsymbol{\theta}}$ is (i) idempotent, i.e., $(\mathbf{P}_{\boldsymbol{\theta}})^2 = \mathbf{P}_{\boldsymbol{\theta}}$, and (ii) symmetric. Also note that $\mathbf{P}_{\boldsymbol{\theta}}\tilde{\mathbf{H}}_1 = \tilde{\mathbf{H}}_1$. Therefore we have

$$\frac{\partial \mathbf{P}_{\boldsymbol{\theta}}\tilde{\mathbf{H}}_1}{\partial \theta_j} = \frac{\partial \tilde{\mathbf{H}}_1}{\partial \theta_j} = \frac{\partial \mathbf{P}_{\boldsymbol{\theta}}}{\partial \theta_j}\tilde{\mathbf{H}}_1 + \mathbf{P}_{\boldsymbol{\theta}}\frac{\partial \tilde{\mathbf{H}}_1}{\partial \theta_j} \tag{B.2}$$

Therefore,

$$\frac{\partial \mathbf{P}_{\boldsymbol{\theta}}}{\partial \theta_j}\tilde{\mathbf{H}}_1 = (\mathbf{I} - \mathbf{P}_{\boldsymbol{\theta}})\frac{\partial \tilde{\mathbf{H}}_1}{\partial \theta_j} = \mathbf{P}_{\boldsymbol{\theta}}^{\perp}\frac{\partial \tilde{\mathbf{H}}_1}{\partial \theta_j} \tag{B.3}$$

Multiplying both sides by $\tilde{\mathbf{H}}_1^{\dagger}$ from right we get

$$\frac{\partial \mathbf{P}_{\boldsymbol{\theta}}}{\partial \theta_j}\mathbf{P}_{\boldsymbol{\theta}} = \mathbf{P}_{\boldsymbol{\theta}}^{\perp}\frac{\partial \tilde{\mathbf{H}}_1}{\partial \theta_j}\tilde{\mathbf{H}}_1^{\dagger}. \tag{B.4}$$

Also note that

$$(\mathbf{P}_{\boldsymbol{\theta}} \frac{\partial \mathbf{P}_{\boldsymbol{\theta}}}{\partial \theta_j})^\top = \frac{\partial \mathbf{P}_{\boldsymbol{\theta}}}{\partial \theta_j} \mathbf{P}_{\boldsymbol{\theta}}. \tag{B.5}$$

Now we use the properties of $\mathbf{P}_{\boldsymbol{\theta}}$ as an orthogonal projection.

$$\frac{\partial \mathbf{P}_{\boldsymbol{\theta}}}{\partial \theta_j} = \frac{\partial \mathbf{P}_{\boldsymbol{\theta}}^2}{\partial \theta_j} = \frac{\partial \mathbf{P}_{\boldsymbol{\theta}}}{\partial \theta_j} \mathbf{P}_{\boldsymbol{\theta}} + \mathbf{P}_{\boldsymbol{\theta}} \frac{\partial \mathbf{P}_{\boldsymbol{\theta}}}{\partial \theta_j}, \tag{B.6}$$

which can be simplified using (B.4) and (B.5),

$$\frac{\partial \mathbf{P}_{\boldsymbol{\theta}}}{\partial \theta_j} = \mathbf{P}_{\boldsymbol{\theta}}^{\perp} \frac{\partial \tilde{\mathbf{H}}_1}{\partial \theta_j} \tilde{\mathbf{H}}_1^{\dagger} + (\mathbf{P}_{\boldsymbol{\theta}}^{\perp} \frac{\partial \tilde{\mathbf{H}}_1}{\partial \theta_j} \tilde{\mathbf{H}}_1^{\dagger})^\top \tag{B.7}$$

Plugging (B.7) into (B.1) completes the proof, i.e.,

$$\begin{aligned}[\mathbf{J}_{\text{vp}}]_{\cdot,j} = &- \left[\mathbf{P}_{\boldsymbol{\theta}}^{\perp} \frac{\partial \tilde{\mathbf{H}}_1}{\partial \theta_j} \tilde{\mathbf{H}}_1^{\dagger} + (\mathbf{P}_{\boldsymbol{\theta}}^{\perp} \frac{\partial \tilde{\mathbf{H}}_1}{\partial \theta_j} \tilde{\mathbf{H}}_1^{\dagger})^\top \right] (\tilde{\mathbf{z}} - \tilde{\mathbf{H}}_2 \boldsymbol{\theta}) \\ &- \mathbf{P}_{\boldsymbol{\theta}}^{\perp} [\tilde{\mathbf{H}}_2]_{\cdot,j}. \end{aligned} \tag{B.8}$$

□

Derivation of $p(\mathbf{p}|\boldsymbol{\theta},\mathbf{z})$**.**

$$\begin{aligned} p(\mathbf{p}|\boldsymbol{\theta}, \mathbf{z}) &= \frac{p(\mathbf{p},\boldsymbol{\theta},\mathbf{z})}{p(\boldsymbol{\theta},\mathbf{z})} \\ &= \frac{p(\mathbf{z}|\mathbf{x})\, p(\mathbf{x})}{p(\boldsymbol{\theta},\mathbf{z})} \\ &= \frac{p(\mathbf{z}|\mathbf{x})\, p(\mathbf{p}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})}{p(\mathbf{z}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})} \\ &= \frac{p(\mathbf{z}|\mathbf{x})\, p(\mathbf{p}|\boldsymbol{\theta})}{p(\mathbf{z}|\boldsymbol{\theta})} \end{aligned} \tag{B.9}$$

□

B.2 Chapter 4

Proof of Lemma 4.2.1. Applying Schur's determinant formula (Lemma A.1.1) on the top-left block of (4.28) and using

$$\log\det(\mathbf{L}_{w_p} \otimes \mathbf{I}_2) = 2 \cdot \log\det \mathbf{L}_{w_p} \tag{B.10}$$

$$= 2 \cdot \tau_{w_p}(\mathcal{G}) \tag{B.11}$$

concludes the proof. □

Proof of Proposition 4.2.5. First note that $\mathbf{P}_{w_p} \triangleq \mathbf{I} - \mathbf{P}^{\perp}_{w_p} \succeq \mathbf{0}$ and $\mathbf{P}^{\perp}_{w_p} \succeq \mathbf{0}$ since they are orthogonal projection matrices and therefore their spectra consist of zeros and ones. For the lower bound we start from (4.65) and apply Lemma A.1.4,

$$\log\det \mathbb{I}(\mathbf{x}) = 2 \cdot \tau_{w_p}(\mathcal{G}) + \log\det(\mathbf{L}_{w_\theta} + \mathbf{E}) \tag{B.12}$$

$$\geq 2 \cdot \tau_{w_p}(\mathcal{G}) + \log\det \mathbf{L}_{w_\theta} \tag{B.13}$$

$$= 2 \cdot \tau_{w_p}(\mathcal{G}) + \tau_{w_\theta}(\mathcal{G}) \tag{B.14}$$

where $\mathbf{E} \triangleq \mathbf{\Delta}^{\top}_{w_p} \mathbf{P}^{\perp}_{w_p} \mathbf{\Delta}_{w_p}$. Define $\delta \triangleq \|\mathbf{\Delta}^{\top}_{w_p} \mathbf{\Delta}_{w_p}\|_{\infty}$. The upper bound also results from Lemma A.1.4 as shown below.

$$\begin{aligned}
\log\det \mathbb{I}(\mathbf{x}) &= 2 \cdot \tau_{w_p}(\mathcal{G}) + \log\det(\mathbf{L}_{w_\theta} + \mathbf{E}) \\
&\leq 2 \cdot \tau_{w_p}(\mathcal{G}) + \log\det(\mathbf{L}_{w_\theta} + \mathbf{\Delta}^{\top}_{w_p} \mathbf{\Delta}_{w_p}) \\
&\leq 2 \cdot \tau_{w_p}(\mathcal{G}) + \log\det(\mathbf{L}_{w_\theta} + \delta \mathbf{I}) \\
&= 2 \cdot \tau_{w_p}(\mathcal{G}) + \sum_{i=1}^{n} \log(\lambda_i(\mathbf{L}_{w_\theta}) + \delta).
\end{aligned}$$

The second and third lines above follow from applying Lemma A.1.4 on

$$\log\det(\underbrace{\mathbf{L}_{w_\theta} + \mathbf{\Delta}^{\top}_{w_p} \mathbf{P}^{\perp}_{w_p} \mathbf{\Delta}_{w_p}}_{\succ \mathbf{0}} + \underbrace{\mathbf{\Delta}^{\top}_{w_p} \mathbf{P}_{w_p} \mathbf{\Delta}_{w_p}}_{\succeq \mathbf{0}}) \tag{B.15}$$

and

$$\log\det(\underbrace{\mathbf{L}_{w_\theta} + \mathbf{\Delta}^{\top}_{w_p} \mathbf{\Delta}_{w_p}}_{\succ \mathbf{0}} + \underbrace{\delta \mathbf{I} - \mathbf{\Delta}^{\top}_{w_p} \mathbf{\Delta}_{w_p}}_{\succeq \mathbf{0}}), \tag{B.16}$$

respectively; see (4.29). □

Proof of Theorem 4.2.6. This result follows from Proposition 4.2.5 and the Squeeze Theorem, since:

$$\lim_{\delta \to 0^+} \mathcal{U} = \mathcal{L}. \tag{B.17}$$

where $\mathcal{U}$ and $\mathcal{L}$ are the upper and lower bounds defined in Proposition 4.2.5. □

B.3 Chapter 5

Proof of Theorem 5.3.1. First note that $\mathbb{V}_w(\mathcal{T})$ is positive for any $\mathcal{T}$ (Definition A.4.1).

1. Normalized: $t_{n,w}(\varnothing) = 0$ by definition.

2. Monotone: Let $\mathcal{G} \triangleq (\mathcal{V},\mathcal{E} \cup \{e\})$. Denote by $\mathbb{T}_{\mathcal{G}}^e$ the set of spanning trees of $\mathcal{G}$ that contain e.

$$t_{n,w}(\mathcal{E} \cup \{e\}) = \sum_{\mathcal{T} \in \mathbb{T}_{\mathcal{G}}} \mathbb{V}_w(\mathcal{T}) = \sum_{\mathcal{T} \in \mathbb{T}_{\mathcal{G}}^e} \mathbb{V}_w(\mathcal{T}) + \sum_{\mathcal{T} \notin \mathbb{T}_{\mathcal{G}}^e} \mathbb{V}_w(\mathcal{T}) \tag{B.18}$$

$$= \sum_{\mathcal{T} \in \mathbb{T}_{\mathcal{G}}^e} \mathbb{V}_w(\mathcal{T}) + t_{n,w}(\mathcal{E}) \geq t_{n,w}(\mathcal{E}). \tag{B.19}$$

3. Supermodular: $t_{n,w}$ is supermodular iff for all $\mathcal{E}_1 \subseteq \mathcal{E}_2 \subseteq \mathcal{E}(\mathcal{K}_n)$ and all $e \in \mathcal{E}(\mathcal{K}_n) \setminus \mathcal{E}_2$,

$$t_{n,w}(\mathcal{E}_2 \cup \{e\}) - t_{n,w}(\mathcal{E}_2) \geq t_{n,w}(\mathcal{E}_1 \cup \{e\}) - t_{n,w}(\mathcal{E}_1). \tag{B.20}$$

   Define $\mathcal{G}_1 \triangleq (\mathcal{V},\mathcal{E}_1)$ and $\mathcal{G}_2 \triangleq (\mathcal{V},\mathcal{E}_2)$. As we showed in (B.19),

$$t_{n,w}(\mathcal{E}_1 \cup \{e\}) - t_{n,w}(\mathcal{E}_1) = \sum_{\mathcal{T} \in \mathbb{T}_{\mathcal{G}_1}^e} \mathbb{V}_w(\mathcal{T}), \tag{B.21}$$

$$t_{n,w}(\mathcal{E}_2 \cup \{e\}) - t_{n,w}(\mathcal{E}_2) = \sum_{\mathcal{T} \in \mathbb{T}_{\mathcal{G}_2}^e} \mathbb{V}_w(\mathcal{T}). \tag{B.22}$$

   Now it suffices to show that $\sum_{\mathcal{T} \in \mathbb{T}_{\mathcal{G}_2}^e} \mathbb{V}_w(\mathcal{T}) \geq \sum_{\mathcal{T} \in \mathbb{T}_{\mathcal{G}_1}^e} \mathbb{V}_w(\mathcal{T})$. This inequality holds since $\mathbb{T}_{\mathcal{G}_1}^e \subseteq \mathbb{T}_{\mathcal{G}_2}^e$.

□

Proof of Theorem 5.3.2.

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 $\Phi_w(\mathcal{E} \cup \{e\}) \geq \Phi_w(\mathcal{E})$. This is equivalent to showing that,

$$\tau_{n,w}(\mathcal{E}_{\text{init}} \cup \mathcal{E} \cup \{e\}) \geq \tau_{n,w}(\mathcal{E}_{\text{init}} \cup \mathcal{E}). \tag{B.23}$$

   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 Theorem A.4.7 on the LHS of (B.23); i.e.,

$$\tau_{n,w}(\mathcal{E}_{\text{init}} \cup \mathcal{E} \cup \{e\}) = \tau_{n,w}(\mathcal{E}_{\text{init}} \cup \mathcal{E}) + \log(1 + w_e\Delta_e). \tag{B.24}$$

   Therefore it suffices 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 $\mathcal{E}_1 \subseteq \mathcal{E}_2 \subseteq \mathcal{E}(\mathcal{K}_n)$ and all $e \in \mathcal{E}(\mathcal{K}_n) \setminus \mathcal{E}_2$,

$$\Phi_w(\mathcal{E}_1 \cup \{e\}) - \Phi_w(\mathcal{E}_1) \geq \Phi_w(\mathcal{E}_2 \cup \{e\}) - \Phi_w(\mathcal{E}_2). \tag{B.25}$$

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

$$\tau_{n,w}(\mathcal{E}_1 \cup \mathcal{E}_{\text{init}} \cup \{e\}) - \tau_{n,w}(\mathcal{E}_1 \cup \mathcal{E}_{\text{init}}) \geq \tau_{n,w}(\mathcal{E}_2 \cup \mathcal{E}_{\text{init}} \cup \{e\}) - \tau_{n,w}(\mathcal{E}_2 \cup \mathcal{E}_{\text{init}}). \tag{B.26}$$

   If $e \in \mathcal{E}_{\text{init}}$, both sides of (B.26) become zero. Hence, we can safely assume that $e \notin \mathcal{E}_{\text{init}}$. For convenience let us define $\mathcal{E}_i^* \triangleq \mathcal{E}_i \cup \mathcal{E}_{\text{init}}$ for $i = 1,2$. Therefore (B.26) can be rewritten as,

$$\tau_{n,w}(\mathcal{E}_1^* \cup \{e\}) - \tau_{n,w}(\mathcal{E}_1^*) \geq \tau_{n,w}(\mathcal{E}_2^* \cup \{e\}) - \tau_{n,w}(\mathcal{E}_2^*). \tag{B.27}$$

   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 Theorem A.4.7 on both sides of (B.27). After doing so we have to show that

$$\log(1 + w_e\Delta_e^{\mathcal{G}_1}) \geq \log(1 + w_e\Delta_e^{\mathcal{G}_2}) \tag{B.28}$$

   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 (B.28) holds iff $\Delta_e^{\mathcal{G}_1} \geq \Delta_e^{\mathcal{G}_2}$. Now

note that

$$\Delta_e^{\mathcal{G}_1} - \Delta_e^{\mathcal{G}_2} = \mathbf{a}_e^\top (\mathbf{L}_{\mathcal{G}_1}^{-1} - \mathbf{L}_{\mathcal{G}_2}^{-1})\, \mathbf{a}_e \geq 0 \tag{B.29}$$

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 A.1.2, $\mathbf{L}_{\mathcal{G}_1}^{-1} \succeq \mathbf{L}_{\mathcal{G}_2}^{-1}$.

□

Proof of Theorem 5.3.4. Define the following indicator function,

$$\mathbb{1}_{\mathbb{T}_\mathcal{G}}(\mathcal{T}) \triangleq \begin{cases} 1 & \mathcal{T} \in \mathbb{T}_\mathcal{G}, \\ 0 & \mathcal{T} \notin \mathbb{T}_\mathcal{G}, \end{cases} \tag{B.30}$$

in which $\mathbb{T}_\mathcal{G}$ denotes the set of spanning trees of $\mathcal{G}$. Now note that,

$$\begin{aligned}
\mathbb{E}_{\mathcal{G}\sim\mathbb{G}(\mathcal{G}_\bullet,\mathbf{p})}\big[t_w(\mathcal{G})\big] &= \mathbb{E}_{\mathcal{G}\sim\mathbb{G}(\mathcal{G}_\bullet,\mathbf{p})}\Big[\sum_{\mathcal{T}\in\mathbb{T}_{\mathcal{G}_\bullet}} \mathbb{1}_{\mathbb{T}_\mathcal{G}}(\mathcal{T})\mathbb{V}_w(\mathcal{T})\Big] && \text{(B.31)}\\
&= \sum_{\mathcal{T}\in\mathbb{T}_{\mathcal{G}_\bullet}} \mathbb{E}_{\mathcal{G}\sim\mathbb{G}(\mathcal{G}_\bullet,\mathbf{p})}\Big[\mathbb{1}_{\mathbb{T}_\mathcal{G}}(\mathcal{T})\mathbb{V}_w(\mathcal{T})\Big] && \text{(B.32)}\\
&= \sum_{\mathcal{T}\in\mathbb{T}_{\mathcal{G}_\bullet}} \mathbb{P}\Big[\mathcal{T} \in \mathbb{T}_\mathcal{G}\Big]\mathbb{V}_w(T) && \text{(B.33)}\\
&= \sum_{\mathcal{T}\in\mathbb{T}_{\mathcal{G}_\bullet}} \mathbb{V}_p(\mathcal{T})\mathbb{V}_w(\mathcal{T}) && \text{(B.34)}\\
&= \sum_{\mathcal{T}\in\mathbb{T}_{\mathcal{G}_\bullet}} \mathbb{V}_{\bar{w}}(\mathcal{T}) && \text{(B.35)}\\
&= t_{\bar{w}}(\mathcal{G}_\bullet). && \text{(B.36)}
\end{aligned}$$

Here we have used the fact the $\mathbb{P}[T \in \mathbb{T}_\mathcal{G}]$ is equal to the probability of existence of every edge of T in $\mathcal{G}$, which is equal to $\mathbb{V}_p(\mathcal{T})$. □

Proof of Theorem 5.3.5. First, note that (5.18) readily follows from Theorem 5.3.4. To see why (5.17)

holds note that $\pi_i \sim \mathrm{Bern}(p_i)$. Therefore,

$$\mathbb{E}[k^*] = \mathbb{E}\left[\sum_{i=1}^{c} \pi_i\right] \tag{B.37}$$

$$= \sum_{i=1}^{c} \mathbb{E}[\pi_i] \tag{B.38}$$

$$= \sum_{i=1}^{c} p_i. \tag{B.39}$$

□

Proof of Theorem 5.3.7. This theorem is a direct application of Chernoff bounds for Poisson trials of independently sampling edges from $\mathcal{C}^+$ with probabilities specified by $\boldsymbol{\pi}^\star$. □

Proof of Theorem 5.3.10. The proof outline is as follows:

Step 1. First, the Cauchy-Binet formula is used to expand the determinant as a sum over $\binom{m}{n}$ terms.

Step 2. The expected value of each of the $\binom{m}{n}$ terms can be easily computed.

Step 3. Finally, the Cauchy-Binet formula is applied again to shrink the sum.

Now we present the proof. We begin by applying the Cauchy-Binet formula:

$$\mathbb{E}_{\boldsymbol{\pi}}\left[\det\left(\sum_{i=1}^{m} \pi_i \mathbf{u}_i \mathbf{v}_i^\top\right)\right] = \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{\mathcal{Q}\in\binom{[m]}{n}} \det\left(\sum_{k\in\mathcal{Q}} \pi_k \mathbf{u}_k \mathbf{v}_k^\top\right)\right] \tag{B.40}$$

$$= \sum_{\mathcal{Q}\in\binom{[m]}{n}} \mathbb{E}_{\boldsymbol{\pi}}\left[\det\left(\sum_{k\in\mathcal{Q}} \pi_k \mathbf{u}_k \mathbf{v}_k^\top\right)\right]. \tag{B.41}$$

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

$$\mathrm{rank}\left(\sum_{k\in\mathcal{Q}} \pi_k \mathbf{u}_k \mathbf{v}_k^\top\right) = \begin{cases} n & \text{iff } \pi_k = 1 \text{ for all } k \in \mathcal{Q}, \\ \gamma < n & \text{otherwise.} \end{cases} \tag{B.42}$$

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

$$\det\left(\sum_{k\in\mathcal{Q}} \pi_k \mathbf{u}_k \mathbf{v}_k^\top\right) = \begin{cases} \det\left(\sum_{k\in\mathcal{Q}} \mathbf{u}_k \mathbf{v}_k^\top\right) & \text{iff } \pi_k = 1 \text{ for all } k \in \mathcal{Q}, \\ 0 & \text{otherwise.} \end{cases} \tag{B.43}$$

But from the independence assumption we know that,

$$\mathbb{P}\Big[\bigwedge_{k\in\mathcal{Q}} \pi_k = 1\Big] = \prod_{k\in\mathcal{Q}} p_k. \tag{B.44}$$

Each individual expectation in (B.41) can be computed as follows.

$$\mathbb{E}_{\boldsymbol{\pi}}\Big[\det\Big(\sum_{k\in\mathcal{Q}} \pi_k \mathbf{u}_k \mathbf{v}_k^\top\Big)\Big] = \det\Big(\sum_{k\in\mathcal{Q}} \mathbf{u}_k \mathbf{v}_k^\top\Big) \prod_{k\in\mathcal{Q}} p_k \tag{B.45}$$

$$= \det\Big(\sum_{k\in\mathcal{Q}} p_k \mathbf{u}_k \mathbf{v}_k^\top\Big). \tag{B.46}$$

Plugging (B.46) back into (B.41) yields,

$$\mathbb{E}_{\boldsymbol{\pi}}\Big[\det\Big(\sum_{i=1}^{m} \pi_i \mathbf{u}_i \mathbf{v}_i^\top\Big)\Big] = \sum_{\mathcal{Q}\in\binom{[m]}{n}} \det\Big(\sum_{k\in\mathcal{Q}} p_k \mathbf{u}_k \mathbf{v}_k^\top\Big). \tag{B.47}$$

Note that (B.47) is nothing but the Cauchy-Binet expansion of $\det\Big(\sum_{i=1}^{m} p_i \mathbf{u}_i \mathbf{v}_i^\top\Big)$. This concludes the proof. □

Bibliography

[1] Pratik Agarwal, Gian Diego Tipaldi, Luciano Spinello, Cyrill Stachniss, and Wolfram Burgard. Robust map optimization using dynamic covariance scaling. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2013.

[2] Brian D.O. Anderson and John B. Moore. Optimal filtering. 1979.

[3] Rosemary A Bailey and Peter J Cameron. Combinatorics of optimal designs. *Surveys in Combinatorics*, 365:19–73, 2009.

[4] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *Robotics & Automation Magazine, IEEE*, 13(3):108–117, 2006.

[5] Tim Bailey, Juan Nieto, Jose Guivant, Michael Stevens, and Eduardo Nebot. Consistency of the EKF-SLAM algorithm. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3562–3568. IEEE, 2006. ISBN 1424402581.

[6] Timothy D Barfoot and Paul T Furgale. Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Transactions on Robotics*, 30(3):679–693, 2014.

[7] RH Barham and Wanzer Drane. An algorithm for least squares estimation of nonlinear parameters when some of the parameters are linear. *Technometrics*, 14(3):757–766, 1972.

[8] Prabir Barooah and Joao P Hespanha. Estimation on graphs from relative measurements. *Control Systems, IEEE*, 27(4):57–74, 2007.

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

[10] Ake Björck. *Numerical methods for least squares problems*. SIAM, 1996.

[11] Francis T Boesch. Synthesis of reliable networks: A survey. *IEEE Transactions on Reliability*, 35 (3):240–246, 1986.

[12] Francis T Boesch. On unreliability polynomials and graph connectivity in reliable network synthesis. *Journal of Graph Theory*, 10(3):339–352, 1986.

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

[14] Michael Bosse, Paul Newman, John Leonard, Martin Soika, Wendelin Feiten, and Seth Teller. An atlas framework for scalable mapping. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 1899–1906. IEEE, 2003.

[15] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[16] Terrence JN Brown, Roger B Mallion, Paul Pollak, and Arnd Roth. Some methods for counting the spanning trees in labelled molecular graphs, examined in relation to certain fullerenes. *Discrete Applied Mathematics*, 67(1):51–66, 1996.

[17] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, Jose Neira, Ian D Reid, and John J Leonard. Simultaneous localization and mapping: Present, future, and the robust-perception age. *arXiv preprint arXiv:1606.05830*, 2016.

[18] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

[19] Nicholas Carlevaris-Bianco, Michael Kaess, and Ryan M. Eustice. Generic node removal for factor-graph SLAM. *IEEE Transactions on Robotics*, 30(6):1371–1385, 2014.

[20] Luca Carlone. Convergence analysis of pose graph optimization via gauss-newton methods. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2013.

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

[22] Luca Carlone and Frank Dellaert. Duality-based verification techniques for 2d SLAM. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4589–4596. IEEE, 2015.

[23] Luca Carlone, Rosario Aragues, Jose Castellanos, and Basilio Bona. A linear approximation for graph-based simultaneous localization and mapping. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2011.

[24] Luca Carlone, Rosario Aragues, José A Castellanos, and Basilio Bona. A fast and accurate approximation for planar pose graph optimization. *The International Journal of Robotics Research*, pages 965–987, 2014.

[25] Luca Carlone, Andrea Censi, and Frank Dellaert. Selecting good measurements via l1 relaxation: A convex approach for robust estimation over graphs. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2667–2674. IEEE, 2014.

[26] Luca Carlone, Giuseppe C Calafiore, Carlo Tommolillo, and Frank Dellaert. Planar pose graph optimization: Duality, optimal solutions, and verification. *IEEE Transactions on Robotics*, 32(3): 545–565, 2016.

[27] Wai-Kai Chen. *Applied graph theory*, volume 13. North Holland Publishing Company, 1971.

[28] Yanqing Chen, Timothy A Davis, William W Hager, and Sivasankaran Rajamanickam. Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. *ACM Transactions on Mathematical Software (TOMS)*, 35(3):22, 2008.

[29] Ching-Shui Cheng. 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, 1981.

[30] C Chow and C Liu. Approximating discrete probability distributions with dependence trees. *IEEE transactions on Information Theory*, 14(3):462–467, 1968.

[31] Joel E Cohen. Connectivity of finite anisotropic random graphs and directed graphs. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 99, pages 315–330. Cambridge Univ Press, 1986.

[32] Thomas H Cormen. *Introduction to algorithms*. MIT press, 2009.

[33] Michael Csorba. *Simultaneous localisation and map building*. PhD thesis, University of Oxford, 1997.

[34] Timothy A Davis. Algorithm 915, suitesparseqr: Multifrontal multithreaded rank-revealing sparse QR factorization. *ACM Transactions on Mathematical Software (TOMS)*, 38(1):8, 2011.

[35] Frank Dellaert and Michael Kaess. Square root SAM: Simultaneous localization and mapping via square root information smoothing. *The International Journal of Robotics Research*, 25(12): 1181–1203, 2006.

[36] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte carlo localization for mobile robots. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1322–1328. IEEE, 1999.

[37] Frank Dellaert, Justin Carlson, Viorela Ila, Kai Ni, and Charles E Thorpe. Subgraph-preconditioned conjugate gradients for large scale SLAM. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2566–2571. IEEE, 2010.

[38] Gamini Dissanayake, Paul Newman, Steve Clark, Hugh Durrant-Whyte, and Michael Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *Robotics and Automation, IEEE Transactions on*, 17(3):229–241, 2001.

[39] Arnaud Doucet. On sequential simulation-based methods for Bayesian filtering. Technical Report CUED/F-INFENG/TR. 310, Cambridge University Department of Engineering, 1998.

[40] Arnaud Doucet. On sequential simulation-based methods for bayesian filtering. Technical Report CUED/F-INFENG/TR. 310, Cambridge University Department of Engineering, 1998.

[41] Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12:656–704, 2009.

[42] Arnaud Doucet, Nando De Freitas, Kevin Murphy, and Stuart Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 176–183. Morgan Kaufmann Publishers Inc., 2000.

[43] Arnaud Doucet, Nando De Freitas, and Neil Gordon. *Sequential Monte Carlo methods in practice*. Springer Verlag, 2001.

[44] Tom Duckett, Stephen Marsland, and Jonathan Shapiro. Learning globally consistent maps by relaxation. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 4, pages 3841–3846. IEEE, 2000.

[45] Tom Duckett, Stephen Marsland, and Jonathan Shapiro. Fast, on-line learning of globally consistent maps. *Autonomous Robots*, 12(3):287–300, 2002.

[46] Hugh Durrant-Whyte. Uncertain geometry in robotics. *IEEE Journal on Robotics and Automation*, 4(1):23–31, 1988.

[47] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localisation and mapping (SLAM): Part I the essential algorithms. *Robotics and Automation Magazine*, 13(2):99–110, 2006.

[48] Austin I. Eliazar and Ronald Parr. Dp-slam: Fast, robust simultaneous localization and mapping without predetermined landmarks. In *IJCAI*, pages 1135–1142, 2003.

[49] Carlos Estrada, Jose Neira, and Juan D. Tardós. Hierarchical SLAM: Real-time accurate mapping of large environments. *Robotics, IEEE Transactions on*, 21(4):588–596, 2005.

[50] Ryan M. Eustice, Hanumant Singh, and John J. Leonard. Exactly sparse delayed-state filters. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2417–2424. IEEE, 2005.

[51] Ryan M Eustice, Hanumant Singh, and John J Leonard. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Transactions on Robotics*, 22(6):1100–1114, 2006.

[52] Marshall L Fisher, George L Nemhauser, and Laurence A Wolsey. *An analysis of approximations for maximizing submodular set functions—II*. Springer, 1978.

[53] John Folkesson and Henrik Christensen. Graphical SLAM-a self-correcting map. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, pages 383–390. IEEE, 2004.

[54] Udo Frese. A proof for the approximate sparsity of SLAM information matrices. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 329–335. IEEE, 2005.

[55] Udo Frese and Gerd Hirzinger. Simultaneous localization and mapping-a discussion. In *Proceedings of the IJCAI Workshop on Reasoning with Uncertainty in Robotics*, pages 17–26. Seattle, 2001.

[56] Udo Frese, Per Larsson, and Tom Duckett. A multilevel relaxation algorithm for simultaneous localization and mapping. *IEEE Transactions on Robotics*, 21(2):196–207, 2005.

[57] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[58] Arpita Ghosh, Stephen Boyd, and Amin Saberi. Minimizing effective resistance of a graph. *SIAM review*, 50(1):37–66, 2008.

[59] Chris Godsil and Gordon Royle. *Algebraic graph theory*. Graduate Texts in Mathematics Series. Springer London, Limited, 2001. ISBN 9780387952413.

[60] Gene Golub and Victor Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on numerical analysis*, 10(2):413–432, 1973.

[61] Gene Golub and Victor Pereyra. Separable nonlinear least squares: the variable projection method and its applications. *Inverse problems*, 19(2):R1, 2003.

[62] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *Robotics, IEEE Transactions on*, 23(1):34–46, 2007.

[63] Giorgio Grisetti, Cyrill Stachniss, Slawomir Grzonka, and Wolfram Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.

[64] Giorgio Grisetti, Rainer Kummerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based SLAM. *Intelligent Transportation Systems Magazine, IEEE*, 2(4):31–43, 2010.

[65] Jose E Guivant. *Efficient simultaneous localization and mapping in large environments*. PhD thesis, The University of Sydney, 2002.

[66] Jose E Guivant and Eduardo Mario Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *Robotics and Automation, IEEE Transactions on*, 17(3):242–257, 2001.

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

[68] Jens-Steffen Gutmann and Kurt Konolige. Incremental mapping of large cyclic environments. In *Computational Intelligence in Robotics and Automation, 1999. CIRA'99. Proceedings. 1999 IEEE International Symposium on*, pages 318–325. IEEE, 1999.

[69] Dirk Hahnel, Wolfram Burgard, Dieter Fox, and Sebastian Thrun. An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS 2003)*, volume 1, pages 206–211, 2003. doi: 10.1109/IROS.2003.1250629.

[70] Christoph Hertzberg. A framework for sparse, non-linear least squares problems on manifolds. In *UNIVERSITÄT BREMEN*. Citeseer, 2008.

[71] Christoph Hertzberg, René Wagner, Udo Frese, and Lutz Schröder. Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion*, 14(1):57–77, 2013.

[72] Dorit S Hochbaum. *Approximation algorithms for NP-hard problems*. PWS Publishing Co., 1996.

[73] Gibson Hu, Kasra Khosoussi, and Shoudong Huang. Towards a reliable SLAM back-end. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 37–43. IEEE, 2013.

[74] Guoquan Huang, Michael Kaess, and John J Leonard. Consistent sparsification for graph optimization. In *Mobile Robots (ECMR), 2013 European Conference on*, pages 150–157. IEEE, 2013.

[75] Guoquan P Huang, Anastasios I Mourikis, and Stergios I Roumeliotis. Observability-based rules for designing consistent EKF SLAM estimators. *The International Journal of Robotics Research*, 29(5):502–528, 2010.

[76] Shoudong Huang and Gamini Dissanayake. Convergence and consistency analysis for extended kalman filter based SLAM. *Robotics, IEEE Transactions on*, 23(5):1036–1049, 2007.

[77] Shoudong Huang, Zhan Wang, and Gamini Dissanayake. Sparse local submap joining filter for building large-scale maps. *IEEE Transactions on Robotics*, 24(5):1121–1130, 2008.

[78] Shoudong Huang, Yingwu Lai, Udo Frese, and Gamini Dissanayake. How far is SLAM from a linear least squares problem? In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3011–3016. IEEE, 2010.

[79] Shoudong Huang, Heng Wang, Udo Frese, and Gamini Dissanayake. On the number of local minima to the point feature based SLAM problem. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2074–2079. IEEE, 2012.

[80] Viorela Ila, Josep M Porta, and Juan Andrade-Cetto. Information-based compact pose SLAM. *Robotics, IEEE Transactions on*, 26(1):78–93, 2010.

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

[82] Simon J Julier and Jeffrey K Uhlmann. A counter example to the theory of simultaneous localization and map building. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 4, pages 4238–4243. IEEE, 2001.

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

[84] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Trans. on Robotics (TRO)*, 24(6):1365–1378, December 2008.

[85] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. iSAM2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012.

[86] Linda Kaufman. A variable projection method for solving separable nonlinear least squares problems. *BIT Numerical Mathematics*, 15(1):49–57, 1975.

[87] Alexander K Kelmans. Connectivity of probabilistic networks. *Automation and Remote Control*, (3):98–116, 1967.

[88] Alexander K Kelmans. On graphs with randomly deleted edges. *Acta Mathematica Hungarica*, 37(1-3):77–88, 1981.

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

[90] Kasra Khosoussi, Shoudong Huang, and Gamini Dissanayake. Novel insights into the impact of graph structure on SLAM. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2014*, pages 2707–2714, 2014.

[91] Kasra Khosoussi, Shoudong Huang, and Gamini Dissanayake. Exploiting the separable structure of SLAM. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

[92] Kasra Khosoussi, Shoudong Huang, and Gamini Dissanayake. Good, bad and ugly graphs for SLAM. *RSS Workshop on the problem of mobile sensors*, 2015.

[93] Kasra Khosoussi, Shoudong Huang, and Gamini Dissanayake. A sparse separable SLAM back-end. *IEEE Transactions on Robotics*, 32(6):1536–1549, 2016.

[94] Kasra Khosoussi, Shoudong Huang, and Gamini Dissanayake. Tree-connectivity: Evaluating the graphical structure of SLAM. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1316–1322. IEEE, 2016.

[95] Kasra Khosoussi, Gaurav S. Sukhatme, Shoudong Huang, and Gamini Dissanayake. Maximizing the weighted number of spanning trees: Near-t-optimal graphs. *CoRR*, abs/1604.01116, 2016. URL `http://arxiv.org/abs/1604.01116`.

[96] Kasra Khosoussi, Gaurav S. Sukhatme, Shoudong Huang, and Gamini Dissanayake. Designing sparse reliable pose-graph SLAM: A graph-theoretic approach. *International Workshop on the Algorithmic Foundations of Robotics*, 2016.

[97] Namhee Kim, Louis Petingi, and Tamar Schlick. Network theory tools for RNA modeling. *WSEAS transactions on mathematics*, 9(12):941, 2013.

[98] Kurt Konolige, Giorgio Grisetti, Rainer Kummerle, Wolfram Burgard, Benson Limketkai, and Regis Vincent. Efficient sparse pose adjustment for 2D mapping. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 22–29. IEEE, 2010.

[99] Henrik Kretzschmar and Cyrill Stachniss. Information-theoretic compression of pose graphs for laser-based slam. *The International Journal of Robotics Research (IJRR)*, 31:1219–1230, 2012. doi: 10.1177/0278364912455072. URL `http://ijr.sagepub.com/content/31/11/1219`.

[100] Henrik Kretzschmar, Cyrill Stachniss, and Giorgio Grisetti. Efficient information-theoretic graph pruning for graph-based SLAM with laser range finders. In *Proc. of the IEEE/RSJ*

International Conference on Intelligent Robots and Systems (IROS), pages 865–871, San Francisco, CA, USA, 2011. URL `http://www.informatik.uni-freiburg.de/~kretzsch/pdf/kretzschmar11iros.pdf`.

[101] Joseph B Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

[102] Rainer Kuemmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g2o: A general framework for graph optimization. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[103] Yasir Latif, César Cadena, and José Neira. Robust loop closing over time for pose graph SLAM. *The International Journal of Robotics Research*, 32(14):1611–1626, 2013.

[104] John J. Leonard and Hugh Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Intelligent Robots and Systems' 91.'Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on*, pages 1442–1447. Ieee, 1991.

[105] John J. Leonard and Hans Jacob S Feder. Decoupled stochastic mapping for mobile robot & AUV navigation. *IEEE Journal of Oceanic Engineering*, 26(4):561–571, 2001.

[106] Minjie Liu, Shoudong Huang, G. Dissanayake, and Heng Wang. A convex optimization based approach for pose SLAM problems. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1898–1903, 2012. doi: 10.1109/IROS.2012.6385742.

[107] Johan Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. URL `http://users.isy.liu.se/johanl/yalmip`.

[108] Feng Lu and Evangelos Milios. Globally consistent range scan alignment for environment mapping. *Autonomous robots*, 4(4):333–349, 1997.

[109] Todd Lupton and Salah Sukkarieh. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *Robotics, IEEE Transactions on*, 28(1):61–76, 2012.

[110] Mladen Mazuran, Wolfram Burgard, and Gian Diego Tipaldi. Nonlinear factor recovery for long-term slam. *The International Journal of Robotics Research*, 35(1-3):50–72, 2016.

[111] Carl D Meyer. *Matrix analysis and applied linear algebra*. Siam, 2000.

[112] Michael Montemerlo and Sebastian Thrun. Simultaneous localization and mapping with unknown data association using FastSLAM. In *Proc. IEEE Int. Conf. Robotics and Automation ICRA '03*, volume 2, pages 1985–1991, 2003. doi: 10.1109/ROBOT.2003.1241885.

[113] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the National conference on Artificial Intelligence*, pages 593–598. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2002.

[114] Philippe Moutarlier and Raja Chatila. An experimental system for incremental environment modelling by an autonomous mobile robot. In *Experimental Robotics I*, pages 327–346. Springer, 1990.

[115] Kevin Murphy. Bayesian map learning in dynamic environments. *Advances in Neural Information Processing Systems*, 12:1015–1021, 2000.

[116] Wendy Myrvold. Reliable network synthesis: Some recent developments. In *Proceedings of International Conference on Graph Theory, Combinatorics, Algorithms, and Applications*, 1996.

[117] Wendy Myrvold, Kim H Cheung, Lavon B Page, and Jo Ellen Perry. Uniformly-most reliable networks do not always exist. *Networks*, 21(4):417–419, 1991.

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

[119] Kai Ni, Drew Steedly, and Frank Dellaert. Tectonic sam: Exact, out-of-core, submap-based SLAM. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 1678–1685. IEEE, 2007.

[120] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[121] Edwin Olson. *Robust and Efficient Robotic Mapping*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, June 2008.

[122] Edwin Olson and Pratik Agarwal. Inference on networks of mixtures for robust robot mapping. *Proceedings of Robotics: Science and Systems (RSS), Sydney, Australia*, 2012.

[123] Edwin Olson and Michael Kaess. Evaluating the performance of map optimization algorithms. In *RSS Workshop on Good Experimental Methodology in Robotics*, page 40, 2009.

[124] Edwin Olson, John Leonard, and Seth Teller. Fast iterative optimization of pose graphs with poor initial estimates. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2262–2269, 2006.

[125] Teresa Anne Parks. *Reducible nonlinear programming problems (separable least squares)*. PhD thesis, Rice University, 1985.

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

[127] Michael J.D. Powell. *A new algorithm for unconstrained optimization*. UKAEA, 1970.

[128] Robert Clay Prim. Shortest connection networks and some generalizations. *Bell system technical journal*, 36(6):1389–1401, 1957.

[129] J Scott Provan and Michael O Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM Journal on Computing*, 12(4):777–788, 1983.

[130] Friedrich Pukelsheim. *Optimal design of experiments*, volume 50. SIAM, 1993.

[131] David M. Rosen, Michael Kaess, and John J. Leonard. RISE: An incremental trust-region method for robust online sparse least-squares estimation. *IEEE Trans. on Robotics, TRO*, 30 (5):1091–1108, Oct 2014.

[132] David M Rosen, Charles DuHadway, and John J Leonard. A convex relaxation for approximate global optimization in simultaneous localization and mapping. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5822–5829. IEEE, 2015.

[133] Havard Rue and Leonhard Held. *Gaussian Markov random fields: theory and applications*. CRC Press, 2005.

[134] Axel Ruhe and Per Åke Wedin. Algorithms for separable nonlinear least squares problems. *SIAM Review*, 22(3):318–337, 1980.

[135] George A. F. Seber and C. J. Wild. *Nonlinear Regression*. Wiley-Interscience, 1989.

[136] Manohar Shamaiah, Siddhartha Banerjee, and Haris Vikalo. Greedy sensor selection: Leveraging submodularity. In *49th IEEE conference on decision and control (CDC)*, pages 2572–2577. IEEE, 2010.

[137] DR Shier. 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, 1974.

[138] Randall Smith and Peter Cheeseman. On the representation and estimation of spatial uncertainty. *The international journal of Robotics Research*, 5(4):56, 1986.

[139] Randall Smith, Matthew Self, and Peter Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous robot vehicles*, 1:167–193, 1990.

[140] H.W. Sorenson. *Parameter estimation: principles and problems*. Control and systems theory. M. Dekker, 1980. ISBN 9780824769871.

[141] Hauke Strasdat. *Local accuracy and global consistency for efficient visual SLAM*. PhD thesis, Imperial College London, 2012.

[142] Hauke Strasdat, JMM Montiel, and Andrew J Davison. Real-time monocular SLAM: Why filter? In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2657–2664. IEEE, 2010.

[143] Hauke Strasdat, José MM Montiel, and Andrew J Davison. Visual SLAM: why filter? *Image and Vision Computing*, 30(2):65–77, 2012.

[144] Niko Sunderhauf and Peter Protzel. Switchable constraints for robust pose graph SLAM. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1879–1884, Oct. doi: 10.1109/IROS.2012.6385590.

[145] Juan D Tardós, José Neira, Paul M Newman, and John J Leonard. Robust mapping and localization in indoor environments using sonar data. *The International Journal of Robotics Research*, 21(4):311–330, 2002.

[146] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[147] Sebastian Thrun and Michael Montemerlo. The graph SLAM algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research*, 25(5-6): 403, 2006.

[148] Sebastian Thrun, Yufeng Liu, Daphne Koller, Andrew Y. Ng, Zoubin Ghahramani, and Hugh Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *The International Journal of Robotics Research*, 23(7-8):693, 2004.

[149] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.

[150] Reha H Tütüncü, Kim C Toh, and Michael J Todd. Solving semidefinite-quadratic-linear programs using sdpt3. *Mathematical programming*, 95(2):189–217, 2003.

[151] Lieven Vandenberghe, Stephen Boyd, and Shao-Po Wu. Determinant maximization with linear matrix inequality constraints. *SIAM journal on matrix analysis and applications*, 19(2):499–533, 1998.

[152] John Vial, Hugh Durrant-Whyte, and Tim Bailey. Conservative sparsification for efficient and consistent approximate estimation. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 886–893. IEEE, 2011.

[153] Guifang Wang. A proof of Boesch's conjecture. *Networks*, 24(5):277–284, 1994.

[154] Heng Wang, Gibson Hu, Shoudong Huang, and Gamini Dissanayake. On the structure of nonlinearities in pose graph SLAM. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2012.

[155] Heng Wang, Shoudong Huang, Udo Frese, and Gamini Dissanayake. The nonlinearity structure of point feature SLAM problems with spherical covariance matrices. *Automatica*, 49(10): 3112–3119, 2013.

[156] Heng Wang, Shoudong Huang, Kasra Khosoussi, Udo Frese, Gamini Dissanayake, and Bingbing Liu. Dimensionality reduction for point feature SLAM problems with spherical covariance matrices. *Automatica*, 51(0):149 – 157, 2015.

[157] Stefan B Williams, Gamini Dissanayake, and Hugh Durrant-Whyte. An efficient approach to the simultaneous localisation and mapping problem. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 1, pages 406–411. IEEE, 2002.

[158] Laurence A Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.

[159] Nikos Zikos and Vassilios Petridis. L-SLAM: Reduced dimensionality FastSLAM with unknown data association. In *Robotics and Automation (ICRA), International Conference on*, pages 4074–4079. IEEE, 2011.

[160] Nikos Zikos and Vassilios Petridis. 6-DoF low dimensionality SLAM (L-SLAM). *Journal of Intelligent & Robotic Systems*, pages 1–18, 2014.