

Spectral Measurement Sparsification for Pose-Graph SLAM

Kevin J. Doherty,¹ David M. Rosen,² and John J. Leonard¹

Abstract—Simultaneous localization and mapping (SLAM) is a critical capability in autonomous navigation, but in order to scale SLAM to the setting of “lifelong” SLAM, particularly under memory or computation constraints, a robot must be able to determine what information should be retained and what can safely be forgotten. In graph-based SLAM, the number of edges (measurements) in a pose graph determines both the memory requirements of storing a robot’s observations and the computational expense of algorithms deployed for performing state estimation using those observations; both of which can grow unbounded during long-term navigation. To address this, we propose a *spectral* approach for pose graph sparsification which maximizes the *algebraic connectivity* of the sparsified measurement graphs, a key quantity which has been shown to control the estimation error of pose graph SLAM solutions. Our algorithm, MAC (for *maximizing algebraic connectivity*), which is based on convex relaxation, is simple and computationally inexpensive, and admits formal *post hoc* performance guarantees on the quality of the solutions it provides. In experiments on benchmark pose-graph SLAM datasets, we show that our approach quickly produces high-quality sparsification results which retain the connectivity of the graph and, in turn, the quality of corresponding SLAM solutions, as compared to a baseline approach which does not consider graph connectivity.

I. INTRODUCTION

The problem of simultaneous localization and mapping (SLAM), in which a robot aims to jointly infer its pose and the location of environmental landmarks, is a critical capability in autonomous navigation. However, as we aim to scale SLAM algorithms to the setting of “lifelong” autonomy, particularly on compute- or memory-limited platforms, a robot must be able to determine *what information should be kept, and what can safely be forgotten* [1]. In particular, in the setting of graph-based SLAM and rotation averaging, the number of edges in a measurement graph determines both the memory required to store a robot’s observations as well as the computation time of algorithms employed for state estimation using this measurement graph.

While there has been substantial work on the topic of measurement pruning (or *sparsification*) in lifelong SLAM (e.g. [2–6]), most existing methods rely on heuristics for sparsification whereby little can be said about the quality of the statistical estimates obtained from the sparsified graph versus the original. Recent work on performance guarantees in the setting of pose-graph SLAM and rotation averaging identified the spectral properties—specifically the *algebraic connectivity* (also known as the *Fiedler value*)—of the measurement graphs encountered in these problems to be central

¹K. Doherty and J. Leonard are with the Massachusetts Institute of Technology (MIT), Boston, MA 02139. ²D. Rosen is with Northeastern University, Boston, MA 02115.

objects of interest, controlling not just the best possible expected performance (per earlier work on Cramér-Rao bounds [7–9]), but also the *worst-case* error of estimators [10–12]. These observations suggest the algebraic connectivity as a natural measure of graph quality for assessing SLAM graphs. This motivates our use of the algebraic connectivity as an *objective* in formulating the graph sparsification problem.

Specifically, we propose a spectral approach to pose graph sparsification which maximizes the algebraic connectivity of the measurement graph subject to a constraint on the number of allowed edges.¹ As we discuss, this corresponds to *E-optimal* design in the setting of pose-graph SLAM [14]. This specific problem turns out to be an instance of the *maximum algebraic connectivity augmentation* problem, which is NP-Hard [15]. To address this, we propose to solve a computationally tractable relaxation and *round* solutions obtained to the relaxed problem to approximate feasible solutions of the original problem. Relaxations of this form have been considered previously; in particular Ghosh and Boyd [16] developed a semidefinite program relaxation to solve problems of the form we consider. However, these techniques do not scale to the size of typical problems encountered in graph-based SLAM. To this end, we propose a first-order optimization approach that we show is practically fast for even quite large SLAM problems. Moreover, we show that the *dual* to our relaxation provides tractable, high-quality bounds on the suboptimality of the solutions we provide *with respect to the original problem*.

In summary, we present an approach for pose graph sparsification by maximizing the *algebraic connectivity* of the measurement graph, a key quantity which has been shown to control the estimation error of pose-graph SLAM solutions. Our method, based on convex relaxation, is *simple* and *computationally inexpensive*, and admits formal *post hoc* performance guarantees on the quality of the solutions it provides. In experiments on several benchmark pose-graph SLAM datasets, we show that our approach quickly produces high-quality sparsification results which retain the connectivity of the graph and better preserve the quality of SLAM solutions compared to a baseline which does not consider graph connectivity.

The remainder of this paper proceeds as follows: In Section II we discuss recent results on the importance of algebraic connectivity in the context of pose-graph SLAM,

¹Our method is related to, but should not be confused with, *spectral sparsification* [13]. Similar to spectral sparsification, we aim to sparsify graphs in a way that preserves their spectral properties. However, our method differs in that we focus only on the algebraic connectivity, whereas traditionally spectral sparsification aims to preserve the entire graph Laplacian spectrum.

previous work on the maximum algebraic connectivity augmentation problem, and prior work on network design with applications in pose graph sparsification. In Section III we give relevant pose-graph SLAM background and a formal description of the maximum algebraic connectivity augmentation problem. In Section IV, we formulate a relaxation of the algebraic connectivity maximization problem, present a first-order optimization approach for solving the relaxation, and describe a simple rounding procedure for obtaining approximate solutions to the pose graph sparsification problem from a solution to the relaxed problem. In Section V we demonstrate our approach on several common pose-graph SLAM benchmark datasets and show that compared to a “topology unaware” baseline, our solutions provide improved graph connectivity and improved accuracy of the maximum-likelihood estimators employed to solve pose-graph SLAM using the sparsified graphs.

II. RELATED WORK

A. Importance of the algebraic connectivity in SLAM

The importance of algebraic connectivity in general has been observed since at least 1973, with the seminal work of Fiedler [17]. In robot perception, the algebraic connectivity has appeared in the context of rotation averaging [7], linear SLAM problems and sensor network localization [9, 18], and pose-graph SLAM [8] as a key quantity controlling estimation performance. In particular, Boumal et al. [7] observed that the inverse of the algebraic connectivity bounds (up to constants) the Cramér-Rao lower bound on the expected mean squared error for rotation averaging. More recently, it appeared as the key quantity controlling the *worst-case* error of estimators applied to measurement graphs in pose-graph SLAM and rotation averaging [10–12] (where *larger* algebraic connectivity is associated with (statistically) lower error).

B. Maximizing the algebraic connectivity

The problem of maximizing the algebraic connectivity subject to cardinality constraints has been considered previously for a number of related applications. Ghosh and Boyd [16] consider a semidefinite program relaxation of the same objective we consider. Alternatively, Nagarajan [19] considered a mixed-integer approach to optimize this objective. While our overall approach can make use of any solution to the relaxation we consider, neither of these methods scales to the types of problems we are considering. To the best of our knowledge, this is the first time an approach has been proposed for pose graph sparsification which makes use of *any* approach to solving a convex (or concave) relaxation.

C. Network design and pose graph sparsification

The theory of optimal experimental design (TOED) [14] gives several optimality criteria applicable to network design. Specifically, A-optimality, T-optimality, E-optimality, and D-optimality are common criteria, each of which corresponds to optimizing a different property of the information matrix

describing the distribution of interest (in SLAM, this is typically the joint distribution over robot and landmark states). Briefly, A-optimal designs minimize the trace of the inverse of the information matrix, D-optimal designs maximize the determinant of the information matrix, E-optimal designs maximize the smallest eigenvalue of the information matrix, and T-optimal designs maximize the trace of the information matrix. Chen et al. [8] discuss the connections between Cramér-Rao bounds for pose-graph SLAM and the A-optimality and T-optimality criteria. Historically, Cramér-Rao bounds and the optimal design metrics arising from them have been popular tools for network design and active SLAM [8, 9], including, e.g. in application to the planning of underwater inspection routes [20].

Khosoussi et al. [18] established many of the first results for optimal graph sparsification (i.e. measurement subset selection) in the setting of SLAM. The convex relaxation they consider is perhaps the closest existing work in the literature to ours. However, in contrast to our approach, they consider the D-optimality criterion, while the results discussed in Section II-A strongly suggest that the quantity of interest with regard to estimation performance is the algebraic connectivity, and therefore the E-optimality criterion.² More practically, the E-optimality criterion is both less computationally expensive to compute *and* to optimize.

Several methods have been proposed to reduce the *number of states* which need to be estimated in a SLAM problem (e.g. [2–4, 21]), typically by marginalizing out state variables. This procedure is usually followed by an edge pruning operation to mitigate the unwanted increase in graph density. Previously considered approaches rely on linearization of measurement models at a particular state estimate in order to compute approximate marginals and perform subsequent pruning. Consequently, little can be said concretely about the quality of the statistical estimates obtained from the sparsified graph compared to the original graph. In contrast, our approach does not require linearization, and provides explicit performance guarantees on the graph algebraic connectivity as compared to the globally optimal algebraic connectivity (which is itself linked to both the *best* and *worst* case performance of estimators applied to the SLAM problem).

III. PROBLEM FORMULATION

We consider graph sparsification in the setting of pose-graph SLAM. Pose-graph SLAM is the problem of estimating n unknown values $x_1, \dots, x_n \in \text{SE}(d)$ given a subset of measurements of their pairwise relative transforms \tilde{x}_{ij} . This problem admits a natural graphical representation $\mathcal{G} \triangleq (\mathcal{V}, \tilde{\mathcal{E}})$ where nodes \mathcal{V} correspond to latent variables $x_i \in \text{SE}(d)$ and edges $(i, j) \in \tilde{\mathcal{E}}$ correspond to noisy measurements of relative poses \tilde{x}_{ij} . We adopt the following generative model for rotation and translation measurements: For each edge $(i, j) \in \tilde{\mathcal{E}}$:

$$\tilde{R}_{ij} = R_{ij} R_{ij}^\epsilon, \quad R_{ij}^\epsilon \sim \text{Langevin}(I_d, \kappa_{ij}) \quad (1a)$$

²Of course, by maximizing the smallest eigenvalue, the E-optimality criterion also selects for information matrices with larger determinant.

$$\tilde{t}_{ij} = t_{ij} + t_{ij}^\epsilon, \quad t_{ij}^\epsilon \sim \mathcal{N}(0, \tau_{ij}^{-1} I_d), \quad (1b)$$

where $x_{ij} = (t_{ij}, R_{ij})$ is the true value of x_{ij} . That is, $x_{ij} = x_i^{-1}x_j$ given the true values of poses x_i and x_j . Under this noise model, the typical nonlinear least-squares formulation of maximum-likelihood estimation (MLE) for $\text{SE}(d)$ synchronization is written as follows:

Problem 1 (MLE for $\text{SE}(d)$ synchronization).

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

Problem 1 also directly captures the problem of rotation averaging under the Langevin noise model simply by taking all $\tau_{ij} = 0$ [22].

In prior work [10–12], we showed that the smallest (nonzero) eigenvalue of the *rotational weight Laplacian* L controls the worst-case error of solutions to Problem 1; this is the algebraic connectivity (or Fiedler value) of the graph with nodes in correspondence with robot poses x_i and edge weights equal to each κ_{ij} . The corresponding eigenvector attaining this value is called the *Fiedler vector*. The rotational weight Laplacian L is the $n \times n$ matrix with i, j -entries:

$$L_{ij} = \begin{cases} \sum_{e \in \delta(i)} \kappa_e, & i = j, \\ -\kappa_{ij}, & \{i, j\} \in \mathcal{E}, \\ 0, & \{i, j\} \notin \mathcal{E}. \end{cases} \quad (3)$$

where $\delta(i)$ denotes the set of edges *incident to* node i . The Laplacian of a graph has several well-known properties that we will use here. The Laplacian L of a graph can be written as a sum of the Laplacians of the subgraphs induced by each of its edges. A Laplacian is always positive-semidefinite, and the “all ones” vector $\mathbf{1}$ of length n is always in its kernel. Finally, a graph has positive algebraic connectivity $\lambda_2(L) > 0$ if and only if it is connected.³

It will be convenient to partition the edges as $\mathcal{E} = \mathcal{E}^o \cup \mathcal{E}^c$, $\mathcal{E}^o \cap \mathcal{E}^c = \emptyset$ into a *fixed* set of edges \mathcal{E}^o and a set of m *candidate* edges \mathcal{E}^c , and where L^o and L^c are the Laplacians of the subgraphs induced by \mathcal{E}^o and \mathcal{E}^c . For our purposes, the subgraph induced by \mathcal{E}^o on \mathcal{V} will typically be constructed from sequential odometric measurements (therefore, $|\mathcal{E}^o| = n-1$), but this is not a requirement of our general approach.⁴ It will be helpful in the subsequent presentation to “overload” the definition of L . Specifically, let $L : \mathbb{R}^m \rightarrow \mathbb{S}_{n \times n}$ be the affine map constructing the total graph Laplacian from a weighted combination of edges in \mathcal{E}^c :

$$L(\omega) \triangleq L^o + \sum_{k=1}^m \omega_k L_k^c, \quad (4)$$

³More specifically, the number of zero eigenvalues of a Laplacian is equal to the number of connected components of its corresponding graph.

⁴In particular, to apply our approach we should select L^o and K to guarantee that the feasible set for Problem 2 contains at least one tree. Then, it is clear that the optimization in Problem 2 will always return a connected graph, since $\lambda_2(L(\omega)) > 0$ if and only if the corresponding graph is connected. Note that this condition is always easy to arrange: for example, we can start with L^o a tree, as we do here, or (even more simply) take L^o to be the zero matrix and simply take $K \geq n-1$.

Algorithm 1 MAC Algorithm

Input: An initial iterate ω

Output: An approximate solution to Problem 2

```

1: function MAC( $\omega$ )
2:    $\omega \leftarrow \text{FRANKWOLFEAC}(\omega)$             $\triangleright$  Solve Problem 3
3:   return  $\Pi(\omega)$                           $\triangleright$  Round solution; eq. (10)
4: end function

```

where L_k^c is the Laplacian of the subgraph induced by edge $e_k = \{i_k, j_k\}$ of \mathcal{E}^c . Our goal in this work will be to identify a subset of $\mathcal{E}^* \subseteq \mathcal{E}^c$ of fixed size $|\mathcal{E}^*| = K$ (equivalently, the edge selection ω), which maximizes the algebraic connectivity $\lambda_2(L(\omega))$. This corresponds to the following optimization problem:

Problem 2 (Algebraic connectivity maximization).

$$p^* = \max_{\substack{\omega \in \{0,1\}^m \\ |\omega| = K}} \lambda_2(L(\omega)) \quad (5)$$

IV. APPROACH

Problem 2 is a variant of the *maximum algebraic connectivity augmentation problem*, which is NP-Hard [15]. The difficulty of Problem 2 stems, in particular, from the integrality constraint on the elements of ω . Consequently, our general approach will be to solve a simpler problem obtained by relaxing the integrality constraints of Problem 2, and, if necessary, *rounding* the solution to the relaxed problem to a solution in the feasible set of Problem 2. In particular, we consider the following *Boolean relaxation* of Problem 2:

Problem 3 (Boolean Relaxation of Problem 2).

$$\max_{\substack{\omega \in \{0,1\}^m \\ \mathbf{1}^\top \omega = K}} \lambda_2(L(\omega)) \quad (6)$$

Relaxing the integrality constraints of Problem 2 dramatically alters the difficulty of the problem. In particular, we know (cf. [16]):

Lemma 1. *The function $F(\omega) = \lambda_2(L(\omega))$ is concave on the set $\omega \in [0, 1]^m, \mathbf{1}^\top \omega = K$.*

Consequently, solving Problem 3, then, amounts to maximizing a concave function over a convex set; this is in fact a convex optimization problem (one can see this by simply considering minimization of the objective $-F(\omega)$) and hence *globally solvable* (see, e.g. [23, 24]). Since a solution to Problem 3 need not be feasible for the original problem, we then *round* solutions to the relaxed problem to their nearest correspondents in the feasible set of Problem 2.

A. Solving the relaxation

There are several methods which could, in principle, be used to solve the relaxation in Problem 3. For example, Ghosh and Boyd [16] consider solving an equivalent semidefinite program. This approach has the advantage of

Algorithm 2 Frank-Wolfe Method for Problem 3

Input: An initial feasible iterate ω

Output: An approximate solution to Problem 3

```

1: function FRANKWOLFEAC( $\omega$ )
2:   for  $t = 0, \dots, T$  do
3:     Compute a Fiedler vector  $y^*$  of  $L(\omega)$ 
4:      $\nabla F(\omega)_k \leftarrow y^{*\top} L_k^c y^*, k = 1, \dots, m$   $\triangleright$  Eq. (8)
5:      $s_t \leftarrow \operatorname{argmax}_s s^\top \nabla F(\omega)$   $\triangleright$  Prob. 4; eq. (9)
6:      $\alpha \leftarrow 2/(2+t)$   $\triangleright$  Compute step size
7:      $\omega \leftarrow \omega + \alpha(s_t - \omega)$ 
8:   end for
9:   return  $\omega$ 
10: end function

```

fast convergence (in terms of the number of iterations required to compute an optimal solution), but can nonetheless be slow for the large problem instances ($m > 1000$) typically encountered in the SLAM setting. Instead, our algorithm for *maximizing algebraic connectivity* (MAC), summarized in Algorithm 1, employs an inexpensive subgradient (more precisely, *supergradient*) approach to solve Problem 3, then rounds its solution to the nearest element of the feasible set for Problem 2.⁵

In particular, MAC uses the *Frank-Wolfe method* (also known as the *conditional gradient method*), a classical approach for solving convex optimization problems of the form in Problem 3 [24]. At each iteration, the Frank-Wolfe method requires (1) linearizing the objective F at a particular ω , (2) maximizing the linearized objective over the (convex) feasible set, and (3) taking a step in the direction of the solution to the linearized problem. The remainder of this section gives a detailed exposition of our adaptation of the Frank-Wolfe method to the problem of algebraic connectivity maximization, which is summarized in Algorithm 2.

The Frank-Wolfe method is particularly advantageous in this setting since the feasible set for Problem 3 is the intersection of the hypercube with the linear subspace determined by $\mathbb{1}^\top \omega = K$ (a linear equality constraint). Consequently this problem amounts to solving a linear program, which can be done easily (and in fact, as we show, admits a simple closed-form solution). In particular, the *direction-finding subproblem* for the Frank-Wolfe method is the following linear program:

Problem 4 (Direction-finding subproblem). *Fix an iterate $\omega \in [0, 1]^m$, $\mathbb{1}^\top \omega = K$. The direction-finding subproblem is to find the point s solving the following linear program:*

$$\max_{s \in [0, 1]^m} s^\top \nabla F(\omega), \quad (7)$$

$$\mathbb{1}^\top s = K.$$

In order to compute the linearized objective in Problem 4 we require a supergradient of the original objective function (in the usual case where F is differentiable at ω , this is

⁵Supergradients are simply the concave analogue of subgradients; i.e., the tangent hyperplane formed by any supergradient of a concave function F must lie *above* F .

simply the gradient of F). It turns out, we can always recover a supergradient of F at a particular ω in terms of a Fiedler vector of $L(\omega)$. Specifically, we have the following theorem (which we prove in Appendix A):

Theorem 2 (Supergradients of $F(\omega)$). *Let $y^*(\omega)$ be any eigenvector of $L(\omega)$ corresponding to $\lambda_2(L(\omega))$. Then:*

$$\nabla F(\omega) = \left[\frac{\partial F}{\partial \omega_1}, \dots, \frac{\partial F}{\partial \omega_m} \right]^\top, \quad (8)$$

$$\frac{\partial F}{\partial \omega_k} = y^{*\top} (\omega)^\top L_k^c y^*(\omega),$$

is a supergradient of F at ω .

Therefore, supergradient computation can be performed by simply recovering an eigenvector of $L(\omega)$ corresponding to $\lambda_2(L(\omega))$.

Problem 4 is a linear program, for which several solution techniques exist [24]. However, in our case, Problem 4 admits a simple, *closed-form* solution s^* attaining its optimal value (which we prove in Appendix B):

Theorem 3 (A closed-form solution to Problem 4). *Let \mathcal{S}^* , $|\mathcal{S}^*| = K$ be the set containing the indices of the K largest elements of $\nabla F(\omega)$, breaking ties arbitrarily where necessary. The vector $s^* \in \mathbb{R}^n$ with element k given by:*

$$s_k^* = \begin{cases} 1, & k \in \mathcal{S}^*, \\ 0, & \text{otherwise,} \end{cases} \quad (9)$$

is an optimizer for Problem 4.

In this work, we use a simple decaying step size α to update ω in each iteration. While in principle, we could instead use a line search method [24, Sec. 2.2]), this would potentially require many evaluations of $F(\omega)$ within each iteration. Since every evaluation of $F(\omega)$ requires an eigenvalue computation, this can become a computational burden for large problems.

In the event that the optimal solution to the relaxed problem is integral, we ensure that we have *also* obtained an optimal solution to the original problem. However, this need not be the case in general. In the (typical) case where integrality does not hold, we *project* the solution to the relaxed problem onto the original constraint set. In this case, an integral solution $\Pi(\omega)$ can be obtained by rounding the largest K components of s to 1, and setting all other components to zero:

$$\Pi(\omega)_k \triangleq \begin{cases} 1, & \text{if } \omega_k \text{ is in the largest } K \text{ elements of } \omega, \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

In general, the Frank-Wolfe algorithm offers *sublinear* (i.e. $\mathcal{O}(1/T)$ after T iterations) convergence to the globally optimal solution in the worst case [25]. However, in this context it has several advantages over alternative approaches. First, we can bound the sparsity of a solution after T iterations. In particular, we know that the solution after T iterations has *at most* KT nonzero entries. Second, the

gradient computation requires only a single computation of the minimal 2 dimensional eigenspace of an $n \times n$ matrix. This can be performed quickly using a variety of methods (e.g. the preconditioned Lanczos method). Finally, as we showed, the direction-finding subproblem in Problem 4 admits a simple *closed-form solution* (as opposed to a projected gradient method which requires projection onto an ℓ_1 -ball). In consequence, despite the fact that gradient-based methods may require many iterations to converge to *globally optimal* solutions, high-quality approximate solutions can be computed fast at the scale necessary for SLAM problems. As we show in the following section, our approach admits *post hoc* suboptimality guarantees even in the event that we terminate optimization prematurely (e.g. when a *fast* but potentially suboptimal solution is required). Critically, these suboptimality guarantees ensure the quality of the solutions of our approach not only with respect to the relaxation, but also with respect to the *original problem*.

B. Post-hoc Suboptimality Guarantees

Algorithm 2 admits several *post hoc* suboptimality guarantees. Let p^* be the optimal value of the original nonconvex maximization in Problem 2. Since Problem 3 is a relaxation of Problem 2, in the event that optimality attains for a vector ω^* , we know:

$$F(\Pi(\omega^*)) \leq p^* \leq F(\omega^*). \quad (11)$$

Therefore, the suboptimality of a rounded solution $\Pi(\omega^*)$ is bounded as follows:

$$p^* - F(\Pi(\omega^*)) \leq F(\omega^*) - F(\Pi(\omega^*)). \quad (12)$$

Consequently, in the event that $F(\omega^*) - F(\Pi(\omega^*)) = 0$, we know that $\Pi(\omega^*)$ *must* correspond to an optimal solution to Problem 2.

The above guarantees apply in the event that we obtain a *maximizer* ω^* of Problem 3. This would seem to pose an issue if we aim to terminate optimization before we obtain a verifiable, globally optimal solution to Problem 3 (e.g. in the presence of real-time constraints). Since these solutions are not necessarily globally optimal in the relaxation, we do not know if their objective value is larger or smaller than the optimal solution to Problem 2. However, we can in fact obtain per-instance suboptimality guarantees of the same kind for *any* estimate $\hat{\omega}$ through the *dual* of our relaxation (cf. Lacoste-Julien et al. [26, Appendix D]). Here, we give a derivation of the dual upper bound which uses only the concavity of F .

Since F is concave, for any $x, y \in [0, 1]^m$, $\mathbf{1}^\top x = \mathbf{1}^\top y = K$ we have:

$$F(y) \leq F(x) + (y - x)^\top \nabla F(x). \quad (13)$$

Consider then the following upper bound:

$$\begin{aligned} F(\omega^*) &\leq F(\hat{\omega}) + (\omega^* - \hat{\omega})^\top \nabla F(\hat{\omega}) \\ &\leq \max_{s \in [0, 1]^m, \mathbf{1}^\top s = K} F(\hat{\omega}) + (s - \hat{\omega})^\top \nabla F(\hat{\omega}) \\ &= F(\hat{\omega}) - \hat{\omega}^\top \nabla F(\hat{\omega}) + \max_{s \in [0, 1]^m, \mathbf{1}^\top s = K} s^\top \nabla F(\hat{\omega}). \end{aligned} \quad (14)$$

Dataset	No. of Nodes	No. of Candidate (Loop Closure) Edges
KITTI 02	4661	43
KITTI 05	2761	66
Intel	1728	785
AIS2Klinik	15115	1614
City10K	10000	10688

TABLE I. Summary of the datasets used in our experiments.

We observe that the solution to the optimization in the last line of (14) is *exactly* the solution to the direction-finding subproblem (Problem 4). Letting \hat{s} be a vector obtained as a solution to Problem 4 at $\hat{\omega}$, we obtain the following *dual* upper bound:

$$F_D(\hat{\omega}) \triangleq F(\hat{\omega}) + \nabla F(\hat{\omega})^\top (\hat{s} - \hat{\omega}). \quad (15)$$

Now, from (14), we have $F_D(\omega) \geq F(\omega^*)$ for any ω in the feasible set. In turn, it is straightforward to verify that the following chain of inequalities hold for any estimator $\hat{\omega}$ in the feasible set of the Boolean relaxation:

$$F(\Pi(\hat{\omega})) \leq p^* \leq F_D(\hat{\omega}), \quad (16)$$

with the corresponding suboptimality guarantee:

$$p^* - F(\Pi(\hat{\omega})) \leq F_D(\hat{\omega}) - F(\Pi(\hat{\omega})). \quad (17)$$

Moreover, we can always recover a suboptimality bound on $\hat{\omega}$ with respect to the optimal value $F(\omega^*)$ to relaxed problem as:

$$F(\omega^*) - F(\hat{\omega}) \leq F_D(\hat{\omega}) - F(\hat{\omega}) \quad (18)$$

The expression appearing on the right-hand side of (18) is the (Fenchel) *duality gap*. Equation (18) also motivates the use of the duality gap as a stopping criterion for Algorithm 2: if the gap is sufficiently close to zero (e.g. to within a certain numerical tolerance), we may conclude that we have reached an optimal solution ω^* to Problem 3.

V. EXPERIMENTAL RESULTS

We implemented the MAC algorithm in Python and all computational experiments were performed on a 2.4 GHz Intel i9-9980HK CPU. For computation of the Fiedler value and the corresponding vector, we use TRACEMIN-Fiedler [27, 28]. In all experiments, we run MAC for a maximum of 20 iterations, or when the duality gap in equation (18) reaches a tolerance of 10^{-8} .

We evaluated our approach using several benchmark pose-graph SLAM datasets. For each dataset, we use odometry edges (between successive poses) to form the base graph and loop closure edges as candidate edges. We consider selection of 10%, 20%, ..., 100% of the candidate loop closure edges in the sparsification problem. We present results on five datasets in this document (summarized in Table I). In particular, we consider here the *Intel* dataset, the *City10K* dataset, KITTI dataset sequences 02, and 05, and the *AIS2Klinik* dataset. The *Intel* dataset and the *AIS2Klinik* dataset are both obtained from real data, while the *City10K* dataset is synthetic. The *City10K* dataset, however, contains far more candidate edges, and therefore serves as a reasonable “stress test” for the computation time of our approach. We

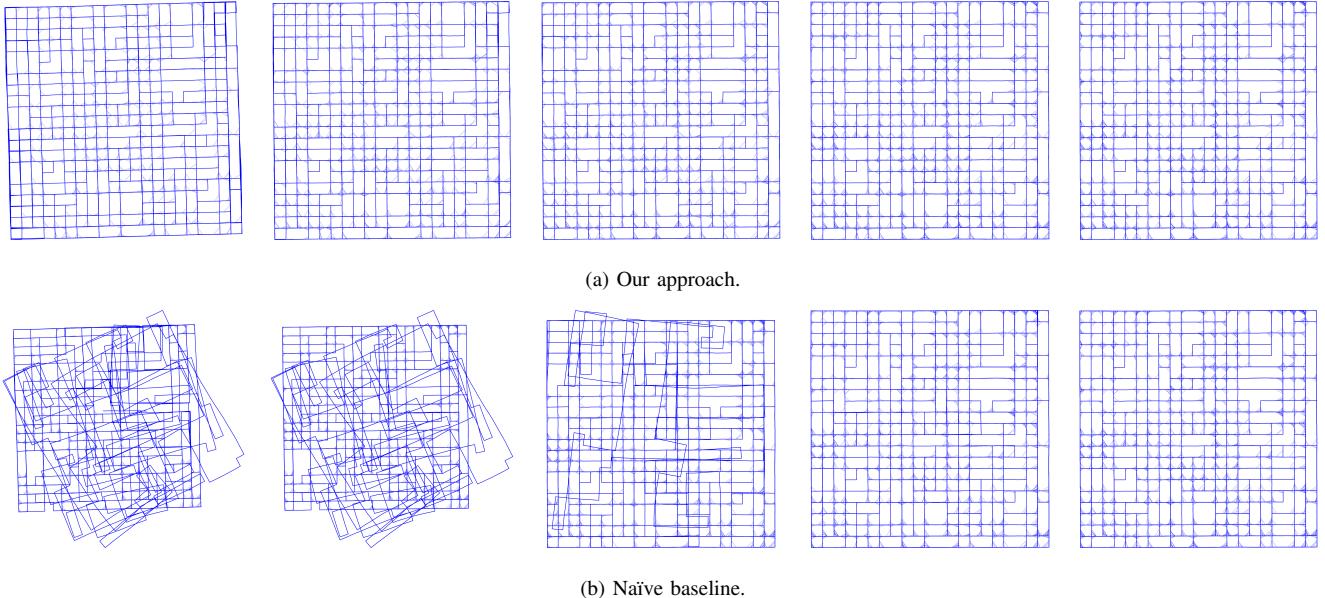


Fig. 1. Qualitative results for pose-graph sparsification. Pose-graph optimization results for the *City10K* dataset with varying degrees of sparsity using (a) our method and (b) a naïve baseline which selects the most certain measurements. Left to right: 20%, 40%, 60%, 80%, and 100% of the candidate edges.

compare our approach to a naïve heuristic method which does not consider graph topology. Specifically, the naïve method selects the edges with the most confident rotation measurements (i.e. the set of K edges $\{i, j\}$ with the largest κ_{ij}). This simple heuristic approach serves two purposes: First, it provides a baseline, topology-agnostic approach to demonstrate the impact of considering graph connectivity in a sparsification procedure; second, we use this method to provide a *sparse* initial estimate to our algorithm. For each method, we compare the graph connectivity (as measured by the Fiedler value) as well as the quality of maximum-likelihood estimators for pose-graph optimization (i.e. solutions to Problem 1) under the edge sets selected by each method. We use SE-Sync [10] to compute the globally optimal estimate of robot poses in each case.⁶

Figure 1 gives a qualitative comparison of the results from our approach as compared with the baseline on the *City10K* dataset across a range of candidate loop closures allowed. We observe that even retaining 60% of the candidate edges, the quality of the results provided by the baseline method degrade significantly compared to those of the full set of loop closures. In contrast, our sparsification approach leads to high-quality estimates even with a *significant* reduction in the number of edges.

For a quantitative comparison of each method, we report three performance measures: (1) the algebraic connectivity $\lambda_2(L(\omega))$ of the graphs determined by each edge selection ω , (2) the “full” objective value from Problem 1 (i.e. keeping 100% of the edges) attained by globally optimal solutions to the *sparsified* problems, and (3) the $\text{SO}(d)$ -orbit distance

between the rotational states of a maximum-likelihood estimator for the sparsified problem and those of a maximum-likelihood estimator for the original (full) objective. The $\text{SO}(d)$ -orbit distance between two rotational state estimates is defined as:

$$d_S(X, Y) \triangleq \min_{G \in \text{SO}(d)} \|X - GY\|_F, \quad (19)$$

$$X, Y \in \text{SO}(d)^n,$$

which can be computed in closed form by means of a singular value decomposition (see Rosen et al. [10, Theorem 5]). The “full” objective value attained by solutions to the sparsified problems serves as one indicator of “how close” solutions to the sparsified problem are to the MLE for the “full” problem. If the “full” objective value attained by the MLE for a sparsified graph is close to that of the MLE computed using 100% of the candidate edges, the MLE for the sparsified graph is likely also a high-quality solution under the full objective. The $\text{SO}(d)$ -orbit distance quantifies the actual deviation (up to global symmetry) between the estimated rotational states in each solution. Since the translational states are recovered analytically (per [10]), this serves as a useful measure, independent of the global scale of the translational states, of the degradation in solution quality from the “full” MLE as we sparsify the graph.

Figure 2 summarizes our quantitative results on each benchmark dataset. Our approach consistently achieves better connected graphs (as measured by the algebraic connectivity). In most cases, a maximum of 20 iterations was enough to achieve solutions to the relaxation with algebraic connectivity very close to the dual upper bound (and therefore nearly globally optimal). Moreover, maximum-likelihood estimators for Problem 1 computed using the

⁶In all of our experiments, SE-Sync returned *certifiably-optimal* solutions to Problem 1.

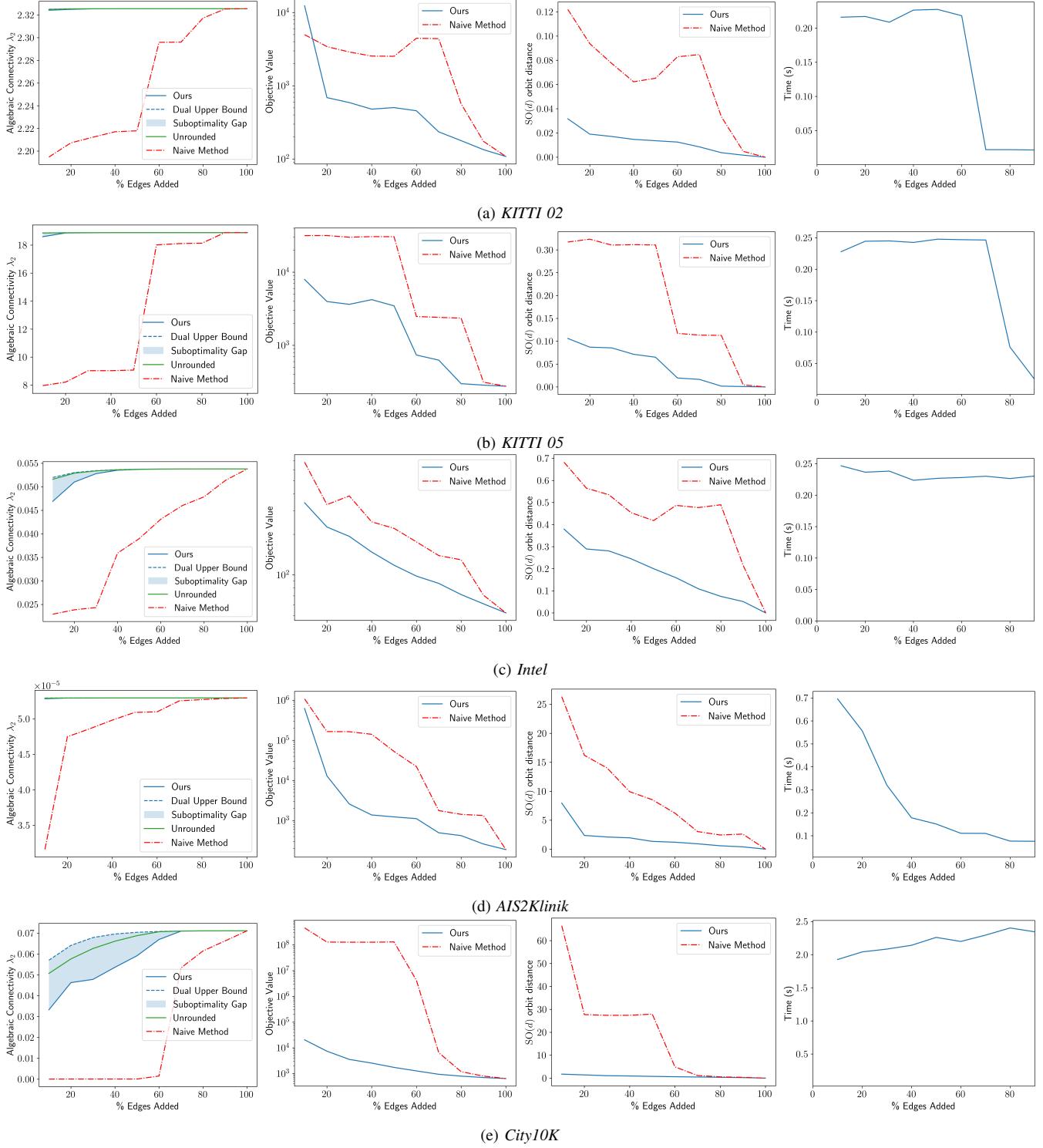


Fig. 2. Quantitative results for pose-graph sparsification. Pose-graph optimization results for (a) the *KITTI 02* dataset, (b) the *KITTI 05* dataset, (c) the *Intel* dataset, (d) the *AIS2Klinik* dataset, and (e) the *City10K* dataset with varying degrees of sparsity (as percent of candidate edges added). Left to right: The algebraic connectivity of the graphs obtained by our method versus the naïve baseline (larger is better), the objective value of the maximum-likelihood estimator for each sparsified problem under the *original* objective, i.e. with all edges retained (smaller is better; note the log-scale), the $\text{SO}(d)$ orbit distance between a maximum-likelihood estimator computing using the sparsified graph and a corresponding maximum-likelihood estimator computed for the graph containing *all* of the candidate edges (smaller is better), and the computation time for our approach. For 100% loop closures, both algorithms return immediately, so no computation time is reported.

sparsified measurement graphs from our method perform significantly better in terms of their “full” objective value and their deviation from a MLE computed using all of the measurement edges.

Beyond providing high-quality sparse measurement graphs, our approach is also fast. For the *Intel* dataset, all solutions were obtained in less than 250 milliseconds. Sparsifying the (larger) *AIS2Klinik* dataset required up to 700 ms, but only around 100 ms when larger edge selections were allowed, as the duality gap tolerance was reached in fewer than the maximum allowed iterations of Frank-Wolfe method. The largest dataset (in terms of candidate edges) is the *City10K* dataset, with over 10000 loop closure measurements to select from. Despite this, our approach produces near-optimal solutions in just 2 seconds.

With respect to the suboptimality guarantees of our approach, it is interesting to note that on both the *Intel* and *City10K* datasets, the rounding procedure introduces fairly significant degradation in algebraic connectivity - particularly for more aggressive sparsity constraints. In these cases, it seems that the Boolean relaxation we consider leads to fractional optimal solutions, rather than solutions amounting to hard selection of just a few edges. It is not clear in these cases whether the integral solutions obtained by rounding are indeed suboptimal for the Problem 2, or whether this is a consequence of the *integrality gap* between *global* optima of the relaxation and of Problem 2.⁷

At present we do not have access to an implementation of the D-optimal sparsification approaches considered in [18]. However, we evaluate our method on similar (and similarly sized) datasets, and a comparison of the computation times suggests that our approach compares favorably in computation time and (consequently) the scale of problems we can consider. For example, Khosoussi et al. [18, Sec. 9.4] report computation times of “ $\gg 10$ minutes” to solve a convex relaxation of the D-optimal sparsification problem on the *City10K* dataset (versus ≈ 2 seconds per Figure 2e). In light of this fact, and since both the D-optimality criterion and E-optimality criterion are essentially variance-minimizing criteria, in the event that one requires D-optimal designs specifically, an interesting avenue for future work would be to use our E-optimal designs to supply an initial estimate to, for example, the convex relaxation approach of Khosoussi et al. [18]. A detailed empirical comparison of the impact of different optimal design criteria on the quality of SLAM solutions would be tremendously helpful for practitioners, and would also be an interesting area for future work.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed an approach for pose-graph measurement sparsification by maximizing the *algebraic connectivity* of the measurement graphs, a key quantity which has been shown to control the estimation error of pose-graph SLAM solutions. Our algorithm, MAC, is based on a

⁷In general, even simply *verifying* the global optimality of solutions to Problem 2 is NP-Hard [15].

first-order optimization approach for solving a convex relaxation of the maximum algebraic connectivity augmentation problem. The algorithm itself is simple and computationally inexpensive, and, as we showed, admits formal *post hoc* performance guarantees on the quality of the solutions it provides. In experiments on several benchmark pose-graph SLAM datasets, our approach quickly produces high-quality sparsification results which better preserve the connectivity of the graph and, consequently, the quality of SLAM solutions computed using those graphs. An interesting area for future work is the empirical comparison of different optimality criteria for the pose graph sparsification problem. Finally, in this work we consider only the removal of measurement graph *edges*. For lifelong SLAM applications, an important aspect of future work will be to combine these procedures with methods for *node* removal (e.g. [2, 4, 29]).

APPENDIX I

A. Subgradients of the Fiedler value

In this appendix we consider the problem of computing a gradient of $F(\omega) = \lambda_2(L(\omega))$ with respect to ω . Strictly speaking, F need not be differentiable at a particular ω (which occurs specifically when $\lambda_2(L(\omega))$ appears with multiplicity greater than 1, i.e. it is not a *simple* eigenvalue). We say that a vector $g \in \mathbb{R}^m$ is a *supergradient* of a concave function F at ω if, for all y, x in the domain of F :

$$F(y) - F(x) \leq g^\top (y - x). \quad (20)$$

Equation (20) generalizes the notion of differentiability to the scenario where the function F may not be (uniquely) differentiable at a particular point. We call the set of all supergradients at a particular value of ω the *superdifferential* of F at ω , denoted $\partial F(\omega)$ [30].⁸

We aim to prove the statement that $\nabla F(\omega)$ as defined in equation (8) is a supergradient of F at ω .

Proof of Theorem 2. We aim to prove the claim by way of equation (20). Let $u, v \in \mathbb{R}^m$, $\|u\|_2 = \|v\|_2 = 1$ be any normalized eigenvectors of $L(x)$ and $L(y)$ with corresponding eigenvalues $\lambda_2(L(x))$ and $\lambda_2(L(y))$, respectively. By definition, then, u and v are Fiedler vectors of $L(x)$ and $L(y)$, respectively. Then the left-hand side of equation (20) can be written as:

$$\begin{aligned} F(y) - F(x) &= \lambda_2(L(y)) - \lambda_2(L(x)) \\ &= v^\top L(y)v - u^\top L(x)u. \end{aligned} \quad (21)$$

Now, substitution of u for the Fiedler vector into the definition in (8), reveals that the k -th element of $\nabla F(x)$ is:

$$\nabla F(x)_k = u^\top L_k^c u. \quad (22)$$

In turn, the right-hand side of (20) can be written as:

$$\nabla F(x)^\top (y - x) = \sum_{k=1}^m (y_k - x_k) u^\top L_k^c u. \quad (23)$$

⁸Supergradients and superdifferentials are simply the analogues of subgradients and subdifferentials for concave functions, as opposed to convex functions.

Since u and v are minimizers of their respective Rayleigh quotient minimization problems, we know:

$$\begin{aligned} v^\top L(y)v &\leq u^\top L(y)u \\ &= u^\top L^o u + \sum_{k=1}^m y_k u^\top L_k^c u, \end{aligned} \quad (24)$$

where the first line follows from the optimality of v with respect to the Rayleigh quotient for $L(y)$ and the second line follows from the definition of $L(y)$. Consider “adding zero” to each y_k in (24) as $x_k - x_k$ to obtain an equivalent expression:

$$\begin{aligned} \sum_{k=1}^m y_k u^\top L_k^c u &= \sum_{k=1}^m (y_k + x_k - x_k) u^\top L_k^c u, \\ &= \sum_{k=1}^m x_k u^\top L_k^c u + \sum_{k=1}^m (y_k - x_k) u^\top L_k^c u. \end{aligned} \quad (25)$$

Comparison to (23) reveals that the last term in (25) is exactly equal to $\nabla F(x)^\top (y - x)$. In turn, substitution of (23) into (25) gives:

$$\sum_{k=1}^m y_k u^\top L_k^c u = \sum_{k=1}^m x_k u^\top L_k^c u + \nabla F(x)^\top (y - x). \quad (26)$$

Substitution back into (24) gives the bound:

$$v^\top L(y)v \leq u^\top L^o u + \sum_{k=1}^m x_k u^\top L_k^c u + \nabla F(x)^\top (y - x). \quad (27)$$

Finally, from the definition of $L(x)$, we obtain

$$v^\top L(y)v \leq u^\top L(x)u + \nabla F(x)^\top (y - x). \quad (28)$$

Subtracting $u^\top L(x)u$ from both sides and substituting into (21) gives the desired result. \square

B. Solving the direction-finding subproblem

This appendix aims to prove the claim that (9) provides an optimal solution to the linear program in Problem 4.

Proof of Theorem 3. Rewriting the objective from Problem 4 in terms of the elements of s and $\nabla F(\omega)$, we have:

$$\begin{aligned} s^\top \nabla F(\omega) &= \sum_{k=1}^m s_k \nabla F(\omega)_k \\ &= \sum_{k=1}^m s_k y^*(\omega)^\top L_k^c y^*(\omega), \end{aligned} \quad (29)$$

where in the last line we have used the definition of $\nabla F(\omega)_k$ in (8). Now, since each $L_k^c \succeq 0$, every component of the gradient must always be nonnegative, i.e. $\nabla F(\omega)_k \geq 0$. Further, since $0 \leq s_k \leq 1$, the objective in (29) is itself a sum of nonnegative terms. From this, it follows directly that the objective in (29) is maximized (subject to the constraint that $\sum_{k=1}^m s_k = K$) specifically by selecting (i.e. by setting $s_k = 1$) each of the K largest components of $\nabla F(\omega)$, giving the result in (9). \square

REFERENCES

- [1] D. M. Rosen, K. J. Doherty, A. T. Espinoza, and J. J. Leonard, “Advances in Inference and Representation for Simultaneous Localization and Mapping,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, 2021.
- [2] N. Carlevaris-Bianco and R. M. Eustice, “Long-term simultaneous localization and mapping with generic linear constraint node removal,” in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, Nov. 2013.
- [3] ———, “Conservative edge sparsification for graph SLAM node removal,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 854–860.
- [4] H. Johannsson, M. Kaess, M. Fallon, and J. Leonard, “Temporally scalable visual SLAM using a reduced pose graph,” in *RSS Workshop on Long-term Operation of Autonomous Robotic Systems in Changing Environments*, Sydney, Australia, Jul. 2012, available as MIT CSAIL Technical Report MIT-CSAIL-TR-2012-013.
- [5] G. Kurz, M. Holoch, and P. Biber, “Geometry-based graph pruning for lifelong SLAM,” *arXiv preprint arXiv:2110.01286*, 2021.
- [6] H. Kretzschmar and C. Stachniss, “Information-theoretic compression of pose graphs for laser-based SLAM,” *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1219–1230, 2012.
- [7] N. Boumal, A. Singer, P.-A. Absil, and V. D. Blondel, “Cramér–Rao bounds for synchronization of rotations,” *Information and Inference: A Journal of the IMA*, vol. 3, no. 1, pp. 1–39, 2014.
- [8] Y. Chen, S. Huang, L. Zhao, and G. Dissanayake, “Cramér–Rao bounds and optimal design metrics for pose-graph SLAM,” *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 627–641, 2021.
- [9] K. Khosoussi, S. Huang, and G. Dissanayake, “Novel insights into the impact of graph structure on SLAM,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 2707–2714.
- [10] D. M. Rosen, L. Carlone, A. S. Bandeira, and J. J. Leonard, “SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group,” *The International Journal of Robotics Research*, vol. 38, no. 2–3, pp. 95–125, 2019.
- [11] K. J. Doherty, D. M. Rosen, and J. J. Leonard, “Performance Guarantees for Spectral Initialization in Rotation Averaging and Pose-Graph SLAM,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.
- [12] ———, “Technical Report: Performance Guarantees for Spectral Initialization in Rotation Averaging and Pose-Graph SLAM,” *arXiv preprint arXiv:2201.03773*, 2022.
- [13] D. A. Spielman and S.-H. Teng, “Spectral sparsification of graphs,” *SIAM Journal on Computing*, vol. 40, no. 4, pp. 981–1025, 2011.
- [14] F. Pukelsheim, *Optimal Design of Experiments*. SIAM, 2006.
- [15] D. Mosk-Aoyama, “Maximum algebraic connectivity augmentation is NP-hard,” *Operations Research Letters*, vol. 36, no. 6, pp. 677–679, 2008.
- [16] A. Ghosh and S. Boyd, “Growing well-connected graphs,” in *Proceedings of the 45th IEEE Conference on Decision and Control*. IEEE, 2006, pp. 6605–6611.
- [17] M. Fiedler, “Algebraic connectivity of graphs,” *Czechoslovak mathematical journal*, vol. 23, no. 2, pp. 298–305, 1973.
- [18] K. Khosoussi, M. Giamou, G. S. Sukhatme, S. Huang, G. Dissanayake, and J. P. How, “Reliable graphs for SLAM,” *The International Journal of Robotics Research*, vol. 38, no. 2–3, pp. 260–298, 2019.
- [19] H. Nagarajan, “On maximizing weighted algebraic connectivity for synthesizing robust networks,” *arXiv preprint arXiv:1805.07825*, 2018.
- [20] A. Kim and R. Eustice, “Toward AUV survey design for optimal coverage and localization using the Cramér Rao lower bound,” in *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*, 2009.
- [21] G. Huang, M. Kaess, and J. Leonard, “Consistent sparsification for graph optimization,” in *Proc. of European Conference on Mobile Robots (ECMR)*, Barcelona, Spain, Sep. 25–27, 2013, pp. 150–157.
- [22] F. Dellaert, D. M. Rosen, J. Wu, R. Mahony, and L. Carlone, “Shanon rotation averaging: Global optimality by surfing $\text{SO}(p)^N$,” in *European Conference on Computer Vision*. Springer, 2020, pp. 292–308.
- [23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [24] D. Bertsekas, *Nonlinear Programming*. Athena Scientific, 2016, vol. 4.
- [25] J. C. Dunn and S. Harshbarger, “Conditional gradient algorithms with open loop step size rules,” *Journal of Mathematical Analysis and Applications*, vol. 62, no. 2, pp. 432–444, 1978.

- [26] S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher, “Block-coordinate frank-wolfe optimization for structural svms,” in *International Conference on Machine Learning*. PMLR, 2013, pp. 53–61.
- [27] M. Manguoglu, E. Cox, F. Saied, and A. Sameh, “TRACEMIN-Fiedler: A parallel algorithm for computing the Fiedler vector,” in *International Conference on High Performance Computing for Computational Science*. Springer, 2010, pp. 449–455.
- [28] A. H. Sameh and J. A. Wisniewski, “A trace minimization algorithm for the generalized eigenvalue problem,” *SIAM Journal on Numerical Analysis*, vol. 19, no. 6, pp. 1243–1259, 1982.
- [29] L. Carlone, Z. Kira, C. Beall, V. Indelman, and F. Dellaert, “Eliminating conditionally independent sets in factor graphs: A unifying perspective based on smart factors,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4290–4297.
- [30] R. T. Rockafellar, *Convex analysis*. Princeton university press, 2015.