IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

Spectral Trade-off for Measurement Sparsification of Pose-graph SLAM

Jiyeon Nam[1], Soojeong Hyeon[1], Youngjun Joo[2], DongKi Noh[3], and Hyungbo Shim[1]

***Abstract*—In this paper, we propose a trade-off optimization algorithm to compute an appropriate number of edges for measurement (edge) sparsification in pose-graph SLAM. The greater the amount of measurement data, the larger is the computational burden. To reduce computational burden, one can remove a portion of measurements. However, reliable data, such as odometric measurements, can be lost if measurements are removed without any principle. To remove measurements which is redundant, we propose a trade-off optimization algorithm between maximization of the Fiedler value and minimization of the largest eigenvalue of adjacency matrix for measurement graph. This problem formulation gives virtues twofold. First, it is scalable. For any dataset, when a weight for trade-off is given, this algorithm determines the appropriate number of edges since this is a trade-off optimization problem. Second, the edges of the measurement graph can be distributed evenly. The algorithm considers the minimization of the largest eigenvalue of the adjacency matrix, so it suppresses the upper bound of the maximum degree of the measurement graph. It removes the redundant information concentrated on a few nodes, and improves the estimation accuracy of the sparsified graph. To validate the performance of the proposed trade-off optimization algorithm, we apply our approach to CSAIL, Intel, and Manhattan datasets.**

***Index Terms*—Mapping, Optimization and Optimal Control, SLAM.**

I. Introduction

In graph-based simultaneous localization and mapping (SLAM), computational expenses depend on the number of edges of the graph. Especially for "lifelong" SLAM, analogous measurement data provide similar information as a robot repeatedly navigates the same place. The redundant edges should be removed for resource-constrained pose-graph SLAM problem [1]–[3].

In estimation theory, Cramér-Rao bound provides a lower bound on the variance of any unbiased estimator [4]. The second smallest eigenvalue (also known as the Fiedler value or algebraic connectivity) of the information weighted measurement graph determines an upper-bound of Cramér-Rao bound. A lower Cramér-Rao bound can be obtained by increasing the Fiedler value of the measurement graph. Moreover, the worst-case error of estimators applied to measurement graphs is also controlled by the Fiedler value [5], [6]. The larger Fiedler value is associated with a lower error.

The Fiedler value increases with an increase in the number of edges, but at the same time, the amount of redundant information also increases. Based on these facts, the authors of [7] propose a method that use the Fiedler value for sparsifying (pruning) measurements of pose-graph SLAM. To this end, they generate an initial graph using only odometric measurements. For the rest of measurements, they remove the redundant edges, except for a pre-determined number of edges. Finally, they constitute a final graph from the initial graph and the remaining edges after removal while maximizing the Fiedler value. They obtain a measurement graph with fewer edges but a maximized Fiedler value in comparison to the original graph. This method is formulated as a specific optimization problem for constructing a measurement graph, nonetheless the number of reducing edges needs to be pre-determined by relying on heuristics.

In this paper, we present a systematic method to obtain a measurement graph with an appropriate number of edges. Although adding an edge increases the Fiedler value of the graph Laplacian, it also increases the largest eigenvalue of adjacency matrix [8]. Therefore, we formulate a trade-off optimization problem between maximizing the Fiedler value of the graph Laplacian and minimizing the largest eigenvalue of adjacency matrix. Since this problem formulation is based on trade-off optimization, it is not necessary to pre-determine the number of remaining edges. We only need to decide on a trade-off weight. Therefore, the proposed algorithm is scalable to any environment. Even if the dataset changes, it is not necessary to specify the number of remaining edges.

Further, the proposed optimization method yields a graph with evenly distributed edges by considering the largest eigenvalue of the adjacency matrix. The square root of the maximum degree of a graph is upper-bounded by the largest eigenvalue of adjacency matrix [9]. The maximum degree of the graph tends to decrease with a decrease in the largest eigenvalue of the adjacency matrix. Essentially, the smaller the maximum degree, the higher evenly distributed edges on the graph. The evenly distributed edges mean evenly distributed loop closures. So, our algorithm, which considers the largest eigenvalue of adjacency matrix, effectively removes the concentration of information on a few nodes.

To summarize, we propose a trade-off optimization problem for deciding the number of edges in pose-graph SLAM. Our

Manuscript received: July, 5, 2023; Revised October, 3, 2023; Accepted November, 6, 2023.

This paper was recommended for publication by Editor S. Behnke upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Advanced Robotics Lab., CTO Division, LG Electronics Inc.

[1]Jiyeon Nam, Soojeong Hyeon, and Hyungbo Shim are with ASRI, Department of Electrical and Computer Engineering, Seoul National University, Republic of Korea {jynam, hsj9537}@cdsl.kr, hshim@snu.ac.kr

[2]Youngjun Joo is with Department of Electrical Engineering and Institute of Advanced Materials and Systems, Sookmyung Women's University, Republic of Korea youngjun.joo@sookmyung.ac.kr

[3]DongKi Noh is with the Advanced Robotics Lab., CTO Division, LG Electronics Inc., Republic of Korea dongki.noh@lge.com

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

problem is scalable to any environment because it systematically determines the number of edges when a weight for trade-off is determined. The weight can be the same for different datasets. Further, it considers not only maximizing the Fiedler value, but also minimizing the largest eigenvalue of adjacency matrix. Consequently, we can obtain an efficient graph with evenly distributed edges while minimizing information loss, as the maximum degree of the graph decreases with a decrease in the largest eigenvalue of adjacency matrix.

Structure of the paper. The rest of this paper is organized as follows. Section 2 reviews the related works. In Section 3, we gives a formal description of spectral trade-off optimization problem, which is a concave maximization problem with an integer constraint. Section 4 presents a relaxation of the problem in Section 3 and an approach to solve the optimization problem based on the Frank-Wolfe method. Section 5 presents the results of using our optimization problem on CSAIL, Intel, and Manhattan datasets and compares the sparsified graph obtained using the proposed method with that reported in [7]. Finally, in Section 6, we offer concluding remarks.

Notation. An (weighted) undirected graph is denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \cdots, N\}$ is a finite nonempty set of nodes and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is an edge set of ordered pairs of nodes. The (symmetric) Laplacian matrix $\mathcal{L} \in \mathbb{S}^{N\times N}$ of an undirected graph is defined as $\mathcal{L} := \mathcal{D} - \mathcal{A}$, where $\mathbb{S}^{N\times N}$ means the set of symmetric $N \times N$ matrices, $\mathcal{D}$ is the degree matrix and $\mathcal{A}$ is the adjacency matrix of the graph. Let $\lambda_i(\cdot)$ for $i = 1, \cdots, N$, be the eigenvalues of a $N \times N$ matrix. Without loss of generality, for the $N \times N$ matrices whose eigenvalues are real, let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$. The operation defined by the symbol $\otimes$ is Kronecker product. We let $\mathrm{vec}(\cdot)$ denote vectorization of a matrix. $\|\cdot\|_2$ and $\|\cdot\|_F$ denote 2-norm of a vector and Frobenius norm of a matrix.

II. RELATED WORKS

A. Design of a graph in SLAM

In this subsection, we introduce related works proposing important criteria for sparsifying the pose-graph. The authors of [10] analyze SLAM problem from the perspective of graph theory. They conclude that the optimization algorithm can perturb some nodes in the graph in order to better fit the noise in the edges when the average degree of graph is small. The works in [11]–[14] theoretically justify the observation in [10] and reveal the fact that E-optimality (minimizing the largest eigenvalue of the covariance matrix) improves with an increase in the Fiedler value. Further, it is shown that the weighted number of spanning trees, as a graph connectivity metric, is closely related to D-optimality (the determinant of Cramér-Rao lower bound). They suggest that these metrics can be efficiently computed for large graphs by exploiting the sparse structure of underlying estimation problems. The general relationship between traditional optimality criteria and the terms of underlying pose-graph are presented in [15]–[17]. These optimality criteria are applicable to network design.

As discussed above, a graph can be designed for specific purposes based on specific criteria. In [18], they introduce "switchable constraints" for loop closure edges. This switchable constraint assigns activation (1) or deactivation (0) on a loop closure edge. Through this process, some loop closures are removed and the graph is redesigned. To maintain the sparsity of graph, the Chow-Liu tree algorithm has been employed after node removal [19]–[21]. In [22], the authors use the trace of the information matrix as a metric for removing certain edges, i.e., edges with little information are pruned. In [7], the authors fix the number of available edges for graph and then construct the final graph as maximizing the Fiedler value.

B. Optimization of the Fiedler value or the largest eigenvalue of adjacency matrix

This subsection introduces related works from the perspective of optimization for the Fiedler value or the largest eigenvalue of adjacency matrix. The authors of [23] reveal that the Fiedler value is a measure of how well-connected the graph is. Then they study the problem of adding edges (from a set of candidate edges) to a graph so as to maximize its Fiedler value. The airport transportation network is designed to maximize the performance (i.e., the Fiedler value) with the constraints of a fixed number of non-stop flight route additions or deletions inside a given candidate route set [24]. In [25], they design the network topology, which is both fast and secure for distributed computation. For fast convergence, they consider the Fiedler value, and for security, they consider the maximum expected portion of infected agents in the network when an agent is initially attacked. A trade-off problem is proposed to consider both properties. The authors of [8] propose a trade-off problem between performance and security, as reported in [25].

III. PROBLEM FORMULATION

In this section, we consider the sparsification (pruning) of measurement in the setting of pose-graph SLAM. Pose-graph SLAM is the problem of estimating N unknown values $x_1, \cdots, x_N \in \mathrm{SE}(d)$ given a subset of measurement of their pairwise relative transforms. This problem is formulated as a nonlinear least-squares problem (maximum-likelihood estimation for $\mathrm{SE}(d)$ synchronization) [5]. Then the nonlinear least-squares problem is simplified to quadratic programming (QP) form. The special Euclidean group $\mathrm{SE}(d)$ comprises an arbitrary combination of rotation and translation. For a fixed value of rotational term, this QP form reduces to the unconstrained minimization of a quadratic form in the translational variable so that a closed-form solution can be easily obtained. For ease of explanation, we begin with Problem 1 in [7] without considering translation because there is a closed-form solution for translation [5]:

Problem 1 (Synchronization problem).

$$\min_{R_i \in \mathrm{SO}(d)} \sum_{(i,j)\in\mathcal{E}} \kappa_{ij} \|R_j - R_i \tilde{R}_{ij}\|_F^2, \tag{1}$$

where $\kappa_{ij} > 0$ is rotational weight, $\tilde{R}_{ij} = \underline{R}_{ij} R_{ij}^{\epsilon}$ is noisy measurement, $\underline{R}_{ij}$ is true value, and $R_{ij}^{\epsilon} \sim \mathrm{Langevin}(I_d, \kappa_{ij})$ is measurement noise. ■

Let the rotational weight Laplacian $\mathcal{L}$ be

$$\mathcal{L} = \begin{cases} \sum_{j\in\mathcal{N}_i} \kappa_{ij}, & i = j, \\ -\kappa_{ij}, & \{i,j\} \in \mathcal{E}, \\ 0, & \{i,j\} \notin \mathcal{E}. \end{cases} \tag{2}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

Its corresponding adjacency matrix $\mathcal{A}$ is

$$\mathcal{A} = \begin{cases} \kappa_{ij}, & \{i,j\} \in \mathcal{E}, \\ 0, & i = j \text{ or } \{i,j\} \notin \mathcal{E}. \end{cases} \tag{3}$$

Let us explain Problem 1 with an example. Suppose that Laplacian matrix and adjacency matrix are given as follows:

$$\mathcal{L} = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}, \quad \mathcal{A} = [\alpha_{ij}] = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

For $i = 1, 2, 3$,

$$R_i = \begin{bmatrix} R_{i1} & R_{i2} \\ R_{i3} & R_{i4} \end{bmatrix} \in \mathbb{R}^{2\times 2}.$$

Let $r_i := \text{vec}(R_i) \in \mathbb{R}^4$ and

$$r := \left[r_1^\top, r_2^\top, r_3^\top\right]^\top \in \mathbb{R}^{12}.$$

We want to show that

$$\frac{1}{2} \sum_{(i,j)\in\mathcal{E}} \alpha_{ij} \|R_j - R_i\|_F^2 = r^\top (\mathcal{L} \otimes I_4) r \tag{4}$$

where I_4 is the identity matrix in $\mathbb{R}^{4\times 4}$. Since the Frobenius norm of arbitrary matrix R is equivalent to the 2-norm of vectorized matrix $\text{vec}(R)$, the left-hand side in (4) is

$$\begin{aligned} &\frac{1}{2} \sum_{(i,j)\in\mathcal{E}} \alpha_{ij} \|R_j - R_i\|_F^2 \\ &= \frac{1}{2} \sum_{(i,j)\in\mathcal{E}} \alpha_{ij} \|\text{vec}(R_j) - \text{vec}(R_i)\|_2^2 \\ &= 2r_1^\top r_1 + r_2^\top r_2 + r_3^\top r_3 - 2r_1^\top r_2 - 2r_1^\top r_3. \end{aligned}$$

The right-hand side in (4) is

$$r^\top (\mathcal{L} \otimes I_4) r = 2r_1^\top r_1 + r_2^\top r_2 + r_3^\top r_3 - 2r_1^\top r_2 - 2r_1^\top r_3.$$

Thus (4) holds. The general objective function of synchronization problem, which is formulated as optimization, is the form of $r^\top \mathcal{L} r$, where r is an argument vector of optimization problem. Since (4) holds, Problem 1 is an element-wise synchronization problem.

In pose-graph SLAM, the second smallest eigenvalue (Fiedler value or algebraic connectivity) of rotational weight Laplacian $\mathcal{L}$ controls the worst-case estimation error of solutions to Problem 1 [5], [6]. We obtain a greater Fiedler value and a lower worst-case estimation error by increasing the number of edges. Meanwhile, increasing edges can be a computational burden on the back-end, which is responsible for solving the optimization problem [26]. To achieve smaller estimation error and lower computational burden simultaneously, we adopt a trade-off optimization problem.

Let partition the edge set $\mathcal{E}$ as $\mathcal{E} = \mathcal{E}^0 \cup \mathcal{E}^c$ such that $\mathcal{E}^0 \cap \mathcal{E}^c = \emptyset$, where $\mathcal{E}^0$ is a fixed set of edges and $\mathcal{E}^c$ is a set of candidate edges. To guarantee the connectivity of the graph, we will construct the set $\mathcal{E}^0$ from sequential odometric measurements. The weighted (undirected) Laplacian $\mathcal{L}$ can be represented as a sum of the Laplacians of the subgraphs induced by each of its edges. Specifically, let $L : \mathbb{R}^m \to \mathbb{S}_{N\times N}$ be the affine map constructing the total graph Laplacian from a weighted combination of edges in $\mathcal{E}^c$:

$$L(\omega) := \mathcal{L}^o + \sum_{k=1}^{m} \omega_k \mathcal{L}_k^c, \tag{5}$$

where $\mathcal{L}^0$ is the Laplacian of the subgraph induced by $\mathcal{E}^0$, $\mathcal{L}_k^c$ is the Laplacian of the subgraph induced by edge $e_k = \{i_k, j_k\}$ of $\mathcal{E}^c$, and $\omega_k \in \{0, 1\}$ is a Boolean element. Also, we consider the affine map $A : \mathbb{R}^m \to \mathbb{S}_{N\times N}$ such that

$$A(\omega) := \mathcal{A}^o + \sum_{k=1}^{m} \omega_k \mathcal{A}_k^c, \tag{6}$$

where $\mathcal{A}^0$ and $\mathcal{A}_k^c$ are adjacency matrices defined analogously as described above.

In a graph, adding an edge increases the Fiedler value $\lambda_2(\mathcal{L})$ but at the same time also increases the largest eigenvalue $\lambda_N(\mathcal{A})$ of adjacency matrix [8]. Therefore, there is trade-off between $\lambda_2(\mathcal{L})$ and $-\lambda_N(\mathcal{A})$. To compute an appropriate number of edges, we formulate the following optimization problem:

Problem 2 (Spectral trade-off problem).

$$\max_{\omega\in\{0,1\}^m} (1-\beta)\lambda_2(L(\omega)) - \beta\lambda_N(A(\omega)) \tag{7}$$

where $\omega := [\omega_1, \cdots, \omega_m]^\top$ and $\beta \in [0, 1]$ is a weight which represents the trade-off between $\lambda_2(L)$ and $-\lambda_N(A)$. ■

We use a soft penalty $(\max -\lambda_N(A))$ rather than a hard constraint on the number of edges as in [7]. Further, there is an appropriate number of edges for Problem 2 above since maximizing $\lambda_2(L)$ increases the number of edges and minimizing $\lambda_N(A)$ decreases the number of edges. Therefore, the number of the remaining edges need not be pre-determined. All we have to do is to decide a trade-off weight β. In addition, the upper bound of the maximum degree of measurement graph is suppressed by $\lambda_N(A)$. By considering the minimization of $\lambda_N(A)$, we obtain a graph with evenly distributed edges. It prevents the concentration of the edges to a few nodes. The redundant information of measurement graph can be removed efficiently, especially in the case of "lifelong" SLAM.

IV. MAIN RESULTS

In this section, we tailor Problem 2 to make it easier. The optimization approach in our work is similar to [7] with the primary difference being the addition of the cost term related to the largest eigenvalue of adjacency matrix in place of an explicit constraint on the number of edges. Since there is an integer constraint on the element of ω, Problem 2 is difficult to solve. We adopt Boolean relaxation [27]:

Problem 3 (Boolean relaxation of Problem 2).

$$\max_{\omega\in[0,1]^m} (1-\beta)\lambda_2(L(\omega)) - \beta\lambda_N(A(\omega)). \tag{8}$$

■

As in [27], the largest eigenvalue of symmetric adjacency matrix $\lambda_N(A(\omega))$ can be represented as the pointwise supremum of a family of linear functions of ω:

$$\lambda_N(A(\omega)) = \sup\{y^\top A(\omega) y : \|y\| = 1, y \in \mathbb{R}^N\}, \tag{9}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

so it is a convex function of ω. And $\lambda_2(L(\omega))$ is concave on $\omega \in [0,1]^m$ [23]. Therefore, maximization of the sum of two concave functions ($(1-\beta)\lambda_2(L(\omega))$ and $-\beta\lambda_N(A(\omega))$) on the linear constraint of ω as (8) is a convex optimization problem.

To solve Problem 3, we employ the subgradient method. The subgradient (it is supergradient for concave optimization) method can be used for big problems such as $m > 1000$ [23]. In particular, we use the Frank-Wolfe method, which is useful for solving a problem of the form in Problem 3. The Frank-Wolfe method is one of the feasible direction methods. So, it is necessary to have an initial feasible point [28]. Finding the initial feasible point may be difficult if the constraint set is specified by nonlinear inequality constraints. However, in Problem 3, it is not difficult to satisfy since the constraint is relaxed to $\omega \in [0,1]^m$.

Let

$$F(\omega) := (1-\beta)\lambda_2(L(\omega)) - \beta\lambda_N(A(\omega)). \tag{10}$$

The most simple way to generate a feasible direction is to solve the following problem:

Problem 4 (Direction-finding subproblem). Let us consider any initial condition for $\omega \in [0,1]^m$. The direction-finding subproblem is to find the point s such that

$$\max_{s\in[0,1]^m} s^\top \mathsf{F}_\nabla(\omega), \tag{11}$$

where $\mathsf{F}_\nabla(\omega)$ is a supergradient of $F(\omega)$. ■

As in Problem 4, the subproblem is a linear program, that is easy to solve [28]. To solve Problem 4, it is necessary to compute a supergradient of $F(\omega)$. As in (10), $F(\omega)$ is a weighted sum of $\lambda_2(L(\omega))$ and $-\lambda_N(A(\omega))$. The authors of [7] provide a supergradient of $\lambda_2(L(\omega))$ in their Theorem 2. In the following lemma, we propose a supergradient of $-\lambda_N(A(\omega))$ in terms of an eigenvector corresponding to the largest eigenvalue of adjacency matrix. In the subsequent theorem, we show that a supergradient of $F(\omega)$ can be represented as a weighted sum of the supergradient of $\lambda_2(L(\omega))$ and the supergradient of $-\lambda_N(A(\omega))$.

Lemma 1: (Supergradients of $-\lambda_N(A(\omega))$). Let $G(\omega) := -\lambda_N(A(\omega))$ and $p(\omega)$ be any normalized eigenvector of $A(\omega)$ corresponding to $\lambda_N(A(\omega))$. Then,

$$\begin{aligned}\mathsf{G}_\nabla(\omega) &= \begin{bmatrix}\frac{\partial G}{\partial \omega_1} & \cdots & \frac{\partial G}{\partial \omega_m}\end{bmatrix}^\top,\\ \frac{\partial G}{\partial \omega_k} &= -p(\omega)^\top \mathcal{A}_k^c p(\omega),\end{aligned} \tag{12}$$

is a supergradient of G at ω. ■

proof: Let

$$G(x) := -\lambda_N(A(x)), \quad G(y) := -\lambda_N(A(y)). \tag{13}$$

Then, we have

$$G(y) - G(x) = -\lambda_N(A(y)) + \lambda_N(A(x)). \tag{14}$$

Let p and q represent any normalized eigenvectors corresponding to $\lambda_N(A(x))$, and $\lambda_N(A(y))$, respectively. From the definitions of p and q, (14) becomes

$$G(y) - G(x) = -q^\top A(y)q + p^\top A(x)p. \tag{15}$$

From the definition of $A(\cdot)$ in (6), for $k = 1, \cdots, m$,

$$\mathsf{G}_\nabla(x)_k := \frac{\partial G}{\partial x_k} = -p^\top \mathcal{A}_k^c p. \tag{16}$$

To show that $\mathsf{G}_\nabla(x) = [\mathsf{G}_\nabla(x)_1, \cdots, \mathsf{G}_\nabla(x)_m]^\top$ is a supergradient of G at x, let us compute $\mathsf{G}_\nabla(x)^\top(y-x)$ as follows:

$$\begin{aligned}\mathsf{G}_\nabla(x)^\top(y-x) &= \begin{bmatrix}-p^\top\mathcal{A}_1^c p & \cdots & -p^\top \mathcal{A}_m^c p\end{bmatrix}\begin{bmatrix} y_1 - x_1 \\ \vdots \\ y_m - x_m\end{bmatrix}\\ &= -\sum_{k=1}^m (y_k - x_k)p^\top \mathcal{A}_k^c p.\end{aligned} \tag{17}$$

By Rayleigh quotient for $A(y)$, we have

$$q^\top A(y) q \geq p^\top A(y) p, \tag{18}$$

since q is an eigenvector corresponding to $\lambda_N(A(y))$. From (6),

$$p^\top A(y)p = p^\top \mathcal{A}^o p + \sum_{k=1}^m y_k p^\top \mathcal{A}_k^c p. \tag{19}$$

For the second term in the right-hand side of (19),

$$\begin{aligned}\sum_{k=1}^m y_k p^\top \mathcal{A}_k^c p &= \sum_{k=1}^m (y_k + x_k - x_k)p^\top \mathcal{A}_k^c p\\ &= \sum_{k=1}^m x_k p^\top \mathcal{A}_k^c p + \sum_{k=1}^m (y_k - x_k) p^\top \mathcal{A}_k^c p.\end{aligned} \tag{20}$$

By applying (17) to the left-hand side of (20), we obtain

$$\sum_{k=1}^m y_k p^\top \mathcal{A}_k^c p = \sum_{k=1}^m x_k p^\top \mathcal{A}_k^c p - \mathsf{G}_\nabla(x)^\top (y-x). \tag{21}$$

From (18) and (19),

$$q^\top A(y) q \geq p^\top A(y) p = p^\top \mathcal{A}^o p + \sum_{k=1}^m y_k p^\top \mathcal{A}_k^c p. \tag{22}$$

By applying (21) to the most right-hand side of (22),

$$\begin{aligned} q^\top A(y) q &\geq p^\top A(y) p\\ &= p^\top \mathcal{A}^o p + \sum_{k=1}^m x_k p^\top \mathcal{A}_k^c p - \mathsf{G}_\nabla(x)^\top(y-x)\\ &= p^\top A(x) p - \mathsf{G}_\nabla(x)^\top (y-x)\end{aligned} \tag{23}$$

Since from (23),

$$-q^\top A(y) q + p^\top A(x) p \leq \mathsf{G}_\nabla(x)^\top (y-x),$$

therefore, by $G(y)$ and $G(x)$ in (15), $G(y) - G(x) \leq \mathsf{G}_\nabla(x)^\top(y-x)$. The m elements of $\mathsf{G}_\nabla(x)$ in (16) is a supergradient of G at x. This completes the proof. □

In the next theorem, we provide a supergradient of $F(\omega)$.

Theorem 2: (Supergradients of $F(\omega)$) Let $H(\omega) := \lambda_2(L(\omega))$ and $G(\omega) := -\lambda_N(A(\omega))$. Then $F(\omega) = (1-\beta)H(\omega) + \beta G(\omega)$ and a supergradient of $F(\omega)$ is given as

$$\begin{aligned}\mathsf{F}_\nabla(\omega) &= (1-\beta)\mathsf{H}_\nabla(\omega) + \beta \mathsf{G}_\nabla(\omega)\\ &= \begin{bmatrix}\frac{\partial F}{\partial \omega_1} & \cdots & \frac{\partial F}{\partial \omega_m}\end{bmatrix}^\top,\\ \frac{\partial F}{\partial \omega_k} &= (1-\beta)u(\omega)^\top \mathcal{L}_k^c u(\omega) - \beta p(\omega)^\top \mathcal{A}_k^c p(\omega),\end{aligned} \tag{24}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

where $u(\omega)$ is any normalized eigenvector corresponding to $\lambda_2(L(\omega))$ and $p(\omega)$ is any normalized eigenvector corresponding to $\lambda_N(A(\omega))$. ■

proof: Let a supergradient of H at point x be h and a supergradient of G at point x be g. By the definition of supergradient, for every y,

$$H(y) - H(x) \leq h^\top (y - x), \tag{25}$$

$$G(y) - G(x) \leq g^\top (y - x). \tag{26}$$

By multiplying $(1-\beta)$ with (25) and β with (26), we obtain

$$(1-\beta)(H(y) - H(x)) \leq (1-\beta)(h^\top (y - x)), \tag{27}$$

$$\beta(G(y) - G(x)) \leq \beta(g^\top (y - x)). \tag{28}$$

The sum of left-hand sides of (27) and (28) becomes

$$(1-\beta)(H(y) - H(x)) + \beta(G(y) - G(x)) = F(y) - F(x), \tag{29}$$

and it is upper-bounded by

$$\begin{aligned} F(y) - F(x) &\leq (1-\beta)(h^\top (y - x)) + \beta(g^\top (y - x)) \\ &= ((1-\beta)h + \beta g)^\top (y - x). \end{aligned} \tag{30}$$

From this inequality, we conclude that $(1-\beta)h + \beta g$ is a supergradient of F at x. By Theorem 2 in [7], $h(\omega) = u(\omega)^\top \mathcal{L}_k^c u(\omega)$ and by Lemma 1, $g(\omega) = -p(\omega)^\top \mathcal{A}_k^c p(\omega)$. Therefore, $(1-\beta)u(\omega)^\top \mathcal{L}_k^c u(\omega) - \beta p(\omega)^\top \mathcal{A}_k^c p(\omega)$ is a supergradient of $F(\omega)$. This completes the proof. □

In Theorem 2, we compute a supergradient $\mathsf{F}_\nabla(\omega)$ to solve Problem 4, which turns into a simple linear program. Now, we provide a closed-form solution for Problem 4 by the following theorem.

Theorem 3: (A closed-form solution to Problem 4). Let $\mathcal{S}^\star$ be the set containing the indices of positive elements of $\mathsf{F}_\nabla(\omega)$. The vector $s^\star \in \mathbb{R}^m$ with element k given by

$$s_k^\star = \begin{cases} 1, & k \in \mathcal{S}^\star, \\ 0, & \text{otherwise}, \end{cases}$$

is an optimizer for Problem 4. ■

proof: Problem 4 is reformulated as

$$\max_{s\in[0,1]^m} \sum_{k=1}^{m} s_k \Big((1-\beta)u(\omega)^\top \mathcal{L}_k^c u(\omega) - \beta p(\omega)^\top \mathcal{A}_k^c p(\omega) \Big). \tag{31}$$

Then, (31) is a maximization problem. Thus, if $\big((1-\beta)u(\omega)^\top \mathcal{L}_k^c u(\omega) - \beta p(\omega)^\top \mathcal{A}_k^c p(\omega)\big)$ is positive, then assign $s_k = 1$, otherwise, $s_k = 0$. This completes the proof. □

V. EXPERIMENTS

In this section, we conduct experiments to verify the performance of the proposed algorithm on three benchmark datasets, CSAIL, Intel, and Manhattan in MATLAB, and all experiments were performed on a 3.6 GHz Intel i7-12700K CPU. Two are real-world examples (CSAIL and Intel) and the remainder is synthetic (Manhattan). The properties of datasets are summarized in Table I. Here, "# nodes", "# measurements", and "# loop closure" denote the number of nodes, measurements, and loop closures, respectively.

Dataset	Type	# nodes	# measurements	# loop closure
CSAIL	real	1045	1172	128
Intel	real	1728	2512	785
Manhattan	synthetic	3500	5453	1954

TABLE I: Summary of datasets.

We compare the proposed algorithm with that proposed in [7]. The algorithm in [7] considers only the Fiedler value, whereas our algorithm is a trade-off between the Fiedler value and the largest eigenvalue of adjacency matrix. So as we have discussed so far, our algorithm systematically determines the number of edges for sparsification, when β is determined. We obtain evenly distributed edges for measurement graph by considering the largest eigenvalue of adjacency matrix, thus redundant information concentrated on a few nodes is removed efficiently.

The procedure of our experiments is as follows. Here, we employ SE-Sync to perform pose-graph optimization [5], [29]–[33]. First, we run our algorithm to obtain appropriate number of (loop closure) edges. Second, we apply the same number of edges obtained in the first step to the algorithm of [7]. Finally, we compare our algorithm with the algorithm of [7] in terms of the sum of translational and rotational errors. Here, the error indicates the difference of "certain value" to be described below between the graph with all existing measurements and the graph with selected measurements.

As stated in Section III, the special Euclidean group $\mathrm{SE}(d)$ comprises an arbitrary combination of translation and rotation. In the case of translational error, we simply use the norm of the difference between two translational state estimates. To compare the two rotational state estimates $X, Y \in \mathrm{SO}(d)^N$, we employ the $\mathrm{SO}(d)$ orbit distance in [5]:

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

For more details on computing the orbit distance, please refer to Theorem 5 in [5]. For given X, $Y \in \mathrm{SO}(d)^N$, let

$$XY^\top = U\Sigma V^\top$$

be a singular value decomposition of $XY^\top$ with $\Sigma = \mathrm{diag}(\sigma_1, \cdots, \sigma_d)$ and $\sigma_1 \geq \cdots \geq \sigma_d \geq 0$. Then the orbit distance $d_S(X,Y)$ is given by

$$d_S(X,Y) = \sqrt{2dN - 2\mathrm{tr}(\Xi\Sigma)}$$

where Ξ is the matrix

$$\Xi = \mathrm{diag}(1, \cdots, 1, \det(UV^\top)) \in \mathbb{R}^{d\times d}.$$

Here, we use two-dimensional data so $d = 2$ and N is given the number of nodes as presented in Table I.

In Problem 2, we propose trade-off problem between $\lambda_2(L)$ and $\lambda_N(A)$. When a graph becomes bigger (i.e., the number of nodes of a graph becomes bigger), the difference between $\lambda_2(L)$ and $\lambda_N(A)$ becomes larger. The trade-off optimization might not work well if the orders of magnitude of $\lambda_2(L)$ and $\lambda_N(A)$ are very different. So for a large graph, we adopt a normalization process for experiments. Let $\lambda_{2,\text{full}}$ be the Fiedler value of measurement graph with all existing measurements

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

Dataset	Item	Ours	[7]
CSAIL	# selected edges	16	16
	error	4.4862	13.2455
	maximum degree	5	5
	normalized F	0.0925	-
Intel	# selected edges	12	12
	error	6.9420	9.3631
	maximum degree	4	4
	normalized F	-0.2172	-
Manhattan	# selected edges	60	60
	error	533.5353	651.6766
	maximum degree	5	4
	normalized F	-0.2590	-

TABLE II: Experimental results when $\beta = 0.5$.

and $\lambda_{N,\text{full}}$ be the largest eigenvalue of adjacency matrix of measurement graph with all existing measurements. Then let us modify Problem 2 slightly as below:

$$\max_{\omega \in \{0,1\}^m} (1-\beta)\frac{\lambda_2(L(\omega))}{\lambda_{2,\text{full}}} - \beta \frac{\lambda_N(A(\omega))}{\lambda_{N,\text{full}}}. \tag{32}$$

We use (32) for experiments since the differences between $\lambda_2(L)$ and $\lambda_N(A)$ are relatively large.

In experiments in Table II, we run our algorithm for 100 iterations and fix $\beta = 0.5$. In items, "# selected edges" means the number of edges obtained from our algorithm, "error" means the sum of translational error and rotational error, "maximum degree" is the maximum degree of obtained graph, and "normalized F" is

$$F(\omega) = (1-\beta)\frac{\lambda_2(L(\omega))}{\lambda_{2,\text{full}}} - \beta \frac{\lambda_N(A(\omega))}{\lambda_{N,\text{full}}}, \tag{33}$$

as in (32).

As item "error" in Table II, our algorithm shows a lower error than [7]. By considering the largest eigenvalue of adjacency matrix in objective function of optimization problem, we suppress the upper bound of the maximum degree of measurement graph. So, we obtain a graph with evenly distributed loop closures and it lowers error as shown in Fig. 1. In Fig. 1, the blue graph in the middle (based on the proposed approach) is superimposed almost similar to the original red graph. In contrast, the blue graph in the rightmost (based on [7]) is out of line with the original red one.

For each dataset, the number of selected edges changes along β, as shown in Fig. 2. Especially, in Manhattan dataset, the number of selected edges remarkably decreases with an increase in β. In Table III, we compute error, $\lambda_2(L)$, $\lambda_N(A)$, runtime (time), and runtime for obtaining λ_{full} (time for λ_{full}) along β. Here, the item "$\lambda_2(L)$" and "$\lambda_N(A)$" are the Fiedler value of the obtained measurement graph and the largest eigenvalue of adjacency matrix of the obtained measurement graph. The abbreviation "b.r." means the values before rounding procedure. When β is close to zero (i.e. when weighted to $\lambda_2(L)$), the error tends to lower. However, still our algorithm tends to show lower (10 cases out of 15 cases) or the same (3 cases out of 15 cases) errors compared with the algorithm in [7]. Comparing $\lambda_2(L)$, $\lambda_N(A)$ and $\lambda_2(L)$ b.r., $\lambda_N(A)$ b.r., it can be seen that the rounding procedure causes degradation. These phenomena are also pointed out in [7].

In addition, as in Fig. 3, our maximization problem of $F(\omega)$ in (33) is working well. For 100 iterations, $F(\omega)$ achieves the maximum, and it maintains. Here, "NOE" represents the number of selected edges.

In summary, we apply the proposed algorithm to CSAIL, Intel, and Manhattan datasets. The proposed algorithm has virtues twofold. First, it is scalable. For any dataset, the number of involved edges for measurement graph is decided when β is determined. And trade-off weight $\beta \in [0, 1]$ can be the same on the different datasets. Second, since our algorithm considers the largest eigenvalue of adjacency matrix in objective function, in most cases, it has a lower error than the algorithm in [7] for the same number of measurements.

VI. Conclusion

In this paper, we address the problem of measurement sparsification for pose-graph SLAM. In resource-constrained pose-graph SLAM, a larger number of edges results in a larger computational burden. Meanwhile, the amount of information in the graph decreases with a decrease in the number of edges. Therefore, a balance point should be determined. In this light, we formulate a trade-off optimization problem between the maximization of the second smallest eigenvalue of Laplacian matrix and the minimization of the largest eigenvalue of adjacency matrix because the second smallest eigenvalue of Laplacian matrix and the largest eigenvalue of adjacency matrix increase with an increase in the number of edges. Our algorithm employs a soft penalty term on increasing the number of edges instead of a hard constraint on the number of edges. So it decides the appropriate number of edges for measurement graph and it is unnecessary to pre-determine the remaining number of edges. In addition, considering the minimization of the largest eigenvalue of adjacency matrix, the edges of measurement graph are distributed evenly. To verify the performance of the proposed algorithm, we implement our algorithm in MATLAB, using CSAIL, Intel, and Manhattan datasets.

References

[1] M. Zhao, X. Guo, L. Song, B. Qin, X. Shi, G. H. Lee, and G. Sun, "A general framework for lifelong localization and mapping in charging environment," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 3305–3312.

[2] Y. Tazaki, "A spanning tree-based multi-resolution approach for pose-graph optimization," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 10033–10040, 2022.

[3] C. E. Denniston, Y. Chang, A. Reinke, K. Ebadi, G. S. Sukhatme, L. Carlone, B. Morrell, and A. A. Agha-mohammadi, "Loop closure prioritization for efficient and scalable multi-robot SLAM," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 9651–9658, 2022.

[4] N. Boumal, A. Singer, P.-A. Absil and V. D. Blondel, "Cramér-Rao bounds for synchronization of rotations," *Inf. Inference: A J. IMA*, vol. 3, no. 1, pp. 1–39, 2014.

[5] 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," *Int. J. Robot. Res.*, vol. 38, no. 2-3, pp. 95–125, 2019.

[6] K. J. Doherty, D. M. Rosen, and J. J. Leonard, "Technical report: Performance guarantees for spectral initialization in rotation averaging and pose-graph SLAM," *arXiv:2201.03773*, 2022.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

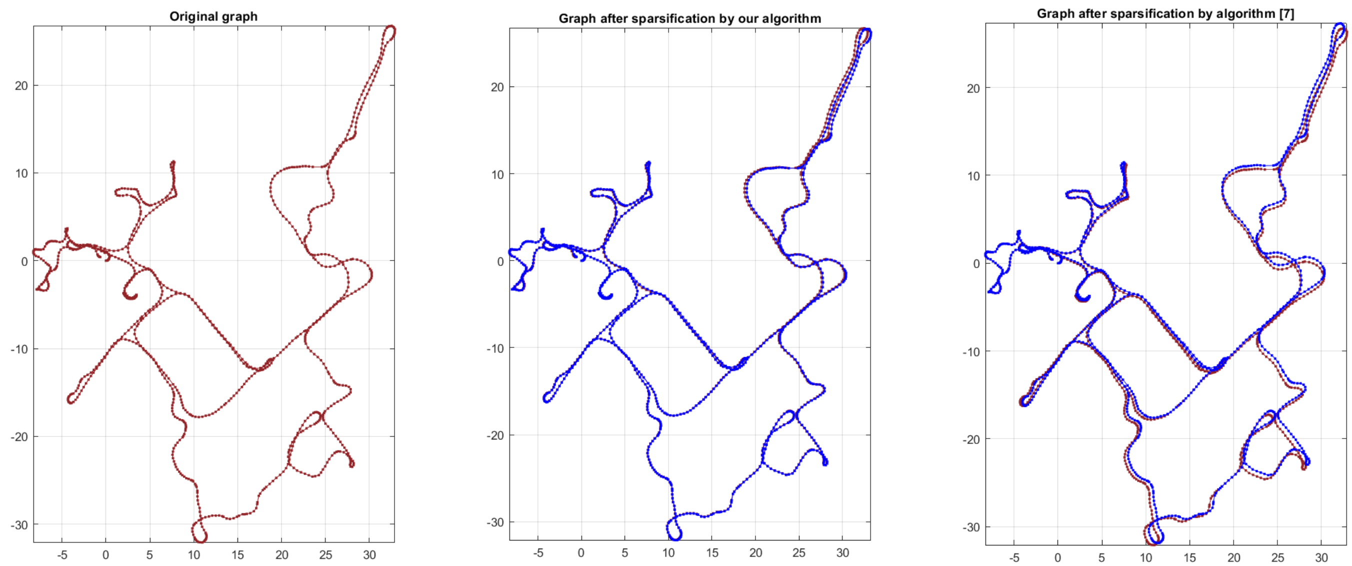

Fig. 1: Comparison of our approach with [7]. Pose-graph optimization results for CSAIL dataset. The leftmost red graph is a graph with all existing measurements. The middle graph is a superimposed graph of the red graph and a blue graph based on our approach. The rightmost graph is a superimposed graph of the red graph and a blue graph based on [7].

Dataset	Item	Ours	[7]	Ours	[7]	Ours	[7]	Ours	[7]	Ours	[7]
	β	0.1		0.3		0.5		0.7		0.9	
CSAIL	error	2.6603	6.7967	2.2108	2.2108	4.4862	13.2455	13.1221	13.3595	9.8064	9.8064
	$\lambda_2(L)$	0.7018	0.7368	0.7002	0.7002	0.6759	0.5348	0.5234	0.5247	0.4911	0.4911
	$\lambda_2(L)$ b.r.	0.7501	0.7481	0.7477	0.7444	0.7434	0.7428	0.7355	0.7275	0.7100	0.7032
	$\lambda_N(A)$	19357	-	19326	-	19326	-	19326	-	19326	-
	$\lambda_N(A)$ b.r.	19338	-	19334	-	19328	-	19317	-	19313	-
	time	6.341s	3.125s	5.893s	3.045s	6.461s	3.242s	6.628s	3.141s	6.687s	3.446s
	time for λ_{full}	0.1014s	-	0.1014s	-	0.1014s	-	0.1014s	-	0.1014s	-
Intel	error	4.3865	7.2876	4.6018	6.9093	6.9420	9.3631	13.8753	26.0634	45.1714	45.1714
	$\lambda_2(L)$	0.0147	0.0185	0.0144	0.0169	0.0102	0.0050	0.0006	0.0005	0.0005	0.0005
	$\lambda_2(L)$ b.r.	0.0502	0.0486	0.0498	0.0479	0.0490	0.0471	0.0478	0.0414	0.0450	0.0005
	$\lambda_N(A)$	434.1	-	424.0	-	447.5	-	424.0	-	424.0	-
	$\lambda_N(A)$ b.r.	425.4	-	425.1	-	424.7	-	424.1	-	424.0	-
	time	29.97s	12.11s	29.76s	12.14s	29.96s	12.17s	29.41s	12.11s	29.15s	7.263s
	time for λ_{full}	0.2307s	-	0.2307s	-	0.2307s	-	0.2307s	-	0.2307s	-
Manhattan	error	499.2	549.3	476.0	576.5	535.5	651.7	644.1	612.9	1432.8	1377.7
	$\lambda_2(L)$	0.0492	0.0487	0.0323	0.0465	0.0263	0.0445	0.0224	0.0254	0.0044	0.0045
	$\lambda_2(L)$ b.r.	0.3540	0.3401	0.3458	0.3285	0.3328	0.3096	0.3062	0.2670	0.2226	0.1578
	$\lambda_N(A)$	20056	-	20056	-	19648	-	18999	-	18987	-
	$\lambda_N(A)$ b.r.	18812	-	18778	-	18773	-	18770	-	18767	-
	time	287.8s	74.46s	280.6s	74.32s	270.7s	74.01s	260.6s	74.29s	216.3s	74.09s
	time for λ_{full}	1.2374s	-	1.2374s	-	1.2374s	-	1.2374s	-	1.2374s	-

TABLE III: The changes in experimental results when β changes.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

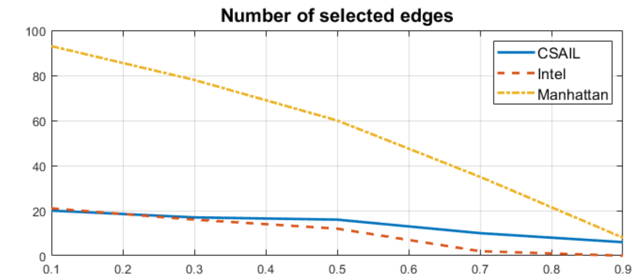

Fig. 2: The changes in the number of edges when β changes.

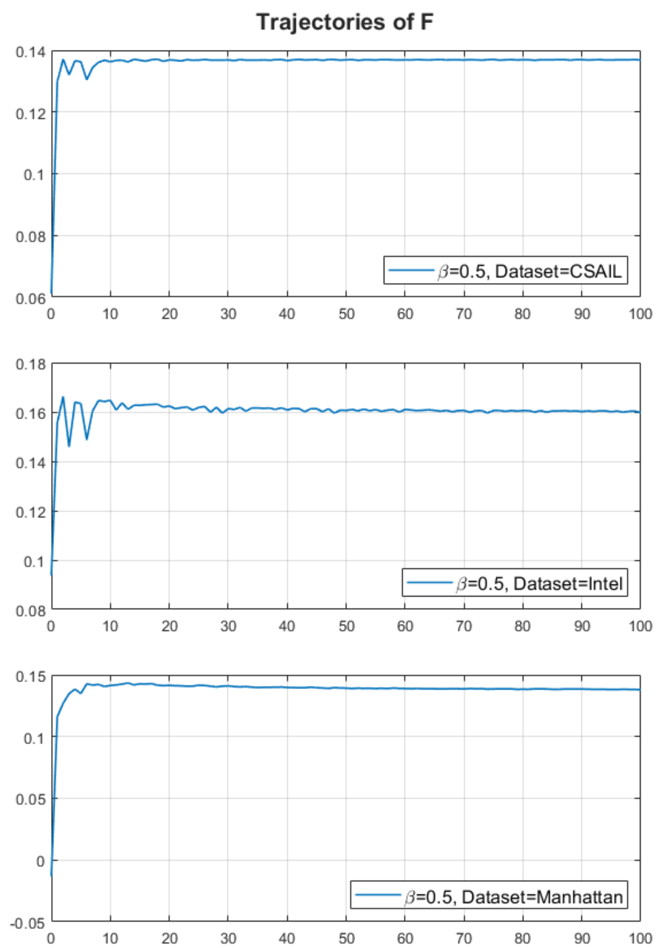

Fig. 3: Maximization of $F(\omega)$ in (33) by solving our trade-off optimization problem. The uppermost, middle, and lowermost plots are based on CSAIL dataset (NOE $= 16$), Intel dataset (NOE $= 12$), and Manhattan dataset (NOE $= 60$), respectively.

[7] K. J. Doherty, D. M. Rosen, and J. J. Leonard, "Spectral measurement sparsification for pose-graph SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 1–8.

[8] A. Gusrialdi and Z. Qu, "Growing connected networks under privacy constraint: achieving trade-off between performance and security," in *Proc. IEEE Conf. Decision and Control*, 2015, pp. 312–317.

[9] D. Stevanović, "Bounding the largest eigenvalue of trees in terms of the largest vertex degree," *Linear Algebra and its Applications*, vol. 360, pp. 35–42, 2003.

[10] E. Olson and M. Kaess, "Evaluating the performance of map optimization algorithms," in *RSS Workshop on Good Experimental Methodology in Robotics*, vol. 15, 2009.

[11] K. Khosoussi, S. Huang, and G. Dissanayake, "Novel insights into the impact of graph structure on SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 2707–2714.

[12] K. Khosoussi, S. Huang, and G. Dissanayake, "Tree-connectivity: Evaluating the graphical structure of SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 1316–1322.

[13] Y. Chen, L. Zhao, K. M. B. Lee, C. Yoo, S. Huang, and R. Fitch, "Broadcast your weaknesses: cooperative active pose-graph SLAM for multiple robots," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 2200-2207, 2020.

[14] Y. Chen, S. Huang, L. Zhao, and G. Dissanayake, "Cramér-Rao bounds and optimal design metrics for pose-graph SLAM," *IEEE Trans. Robot.*, vol. 37, no. 2, pp. 627–641, Apr. 2021.

[15] H. Carrillo, P. Dames, V. Kumar, and J. A. Catellanos, "Autonomous robotic exploration using a utility function based on Rényi's general theory of entropy," *Auton. Robots*, vol. 42, no. 2, pp. 235–256, 2018.

[16] J. A. Placed and J. A. Castellanos, "Fast autonomous robotic exploration using the underlying graph structure," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 6672–6679.

[17] J. A. Placed and J. A. Castellanos, "A general relationship between optimality criteria and connectivity indices for active graph-SLAM," *IEEE Robot. Automat. Lett.*, vol. 8, no. 2, pp. 816–823, 2023.

[18] N. Sünderhauf and P. Protzel, "Switchable constraints for robust pose graph SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 1879–1884.

[19] H. Kretzschmar and C. Stachniss, "Information-theoretic compression of pose graphs for laser-based SLAM," *Int. J. Robot. Res.*, vol. 31, no. 11, pp. 1219–1230, 2012.

[20] N. Carlevaris-Bianco, M. Kaess, and R. M. Eustice, "Generic node removal for factor-graph SLAM," *IEEE Trans. Robot.*, vol. 30, no. 6, pp. 1371–1385, Dec. 2014.

[21] J. Vallvé, J. Solà, and J. Andrade-Cetto, "Graph SLAM sparsification with populated topologies using factor descent optimization," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 1322-1329, 2018.

[22] G. Kurz, M. Holoch, and P. Biber, "Geometry-based graph pruning for lifelong SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 3313–3320.

[23] A. Ghosh and S. Boyd, "Growing well-connected graphs," in *Proc. IEEE Conf. Decision and Control*, 2006, pp.6605–6611.

[24] P. Wei and D. Sun, "Weighted algebraic connectivity: An application to airport transportation network," in *Proc. 18th IFAC World Congress*, 2011, pp.13864–13869.

[25] J. Liu and T. Başar, "Toward optimal network topology design for fast and secure distributed computation," in *Proc. Decision and Game Theory for Security*, 2014, pp. 234–245.

[26] K. Khosoussi, M. Giamou, G. S. Sukhatme, S. Huang, G. Dissanayake, and J. P. How, "Reliable graphs for SLAM," *Int. J. Robot. Res.*, vol. 38, no. 2-3, pp. 260–298, 2019.

[27] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[28] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 2016.

[29] P. A. Absil, C. G. Baker and K. A. Gallivan, "Trust-region methods on Riemannian manifolds," *Found. of Comput. Math.*, vol. 7, pp. 303–330, 2007.

[30] D. M. Rosen, L. Carlone, A. S. Bandeira, and J. J. Leonard, "A Certifiably Correct Algorithm for Synchronization over the Special Euclidean Group," *Proc. Int. Workshop Algorithmic Found. Robot.*, 2016.

[31] D. M. Rosen and L. Carlone, "Computational Enhancements for Certifiably Correct SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017.

[32] D. M. Rosen, "Accelerating Certifiable Estimation with Preconditioned Eigensolvers," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 12507–12514, 2022.

[33] N. Boumal, B. Mishra, P. A. Absil, and R. Sepulchre, "Manopt, a MATLAB Toolbox for Optimization on Manifolds," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1455–1459, 2014.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.