2016 IEEE International Conference on Robotics and Automation (ICRA)
Stockholm, Sweden, May 16-21, 2016

A Unified Resource-Constrained Framework for Graph SLAM

Liam Paull, Guoquan Huang, and John J. Leonard

Abstract— **Graphical methods have proven an extremely useful tool employed by the mobile robotics community to frame estimation problems. Incremental solvers are able to process incoming sensor data and produce maximum a posteriori (MAP) estimates in realtime by exploiting the natural sparsity within the graph for reasonable-sized problems. However, to enable truly longterm operation in prior unknown environments requires algorithms whose computation, memory, and bandwidth (in the case of distributed systems) requirements scale constantly with time and environment size. Some recent approaches have addressed this problem through a two-step process - first the variables selected for removal are marginalized which induces density, and then the result is sparsified to maintain computational efficiency. Previous literature generally addresses only one of these two components.**

In this work, we attempt to explicitly connect all of the aforementioned resource constraint requirements by considering the node removal and sparsification pipeline in its entirety. We formulate the node selection problem as a minimization problem over the penalty to be paid in the resulting sparsification. As a result, we produce node subset selection strategies that are optimal in terms of minimizing the impact, in terms of Kullback-Liebler divergence (KLD), of approximating the dense distribution by a sparse one. We then show that one instantiation of this problem yields a computationally tractable formulation. Finally, we evaluate the method on standard datasets and show that the KLD is minimized as compared to other commonly-used heuristic node selection techniques.

I. Introduction

In many realistic application scenarios, robots are required to navigate over long time periods in unknown and uncertain environments by performing simultaneous localization and mapping (SLAM). For example, a team of autonomous underwater vehicles (AUVs) is often used to cooperatively collect data in the ocean (e.g., for seabed mapping). Recently, graph-based approaches have emerged as one of the most popular approaches to SLAM [1]. In this formulation, each new sensor measurement adds a new edge (constraint) between two nodes (states) into the graph. The most likely configuration of the states can be efficiently found by exploiting the sparse structure of the system, and an incremental method (e.g. [2]) can be further utilized in order to achieve real-time performance for "medium-sized" problems.

This work was partially supported by the Office of Naval Research under grants N00014-10-1-0936, N00014-11-1-0688 and N00014-13-1-0588 and by the National Science Foundation under grant IIS-1318392, which we gratefully acknowledge.

L. Paull and J. Leonard are with the Computer Science and Artificial Intelligence Laboratory (CSAIL), MIT, Cambridge, MA 02139, USA. Email: {lpaull,jleonard}@mit.edu

G. Huang is with the Dept. of Mechanical Engineering, University of Delaware, Newark, DE 19716, USA. Email: ghuang@udel.edu

*The authors would like to thank Mladen Mazuran for helpful discussions during the preparation of the manuscript

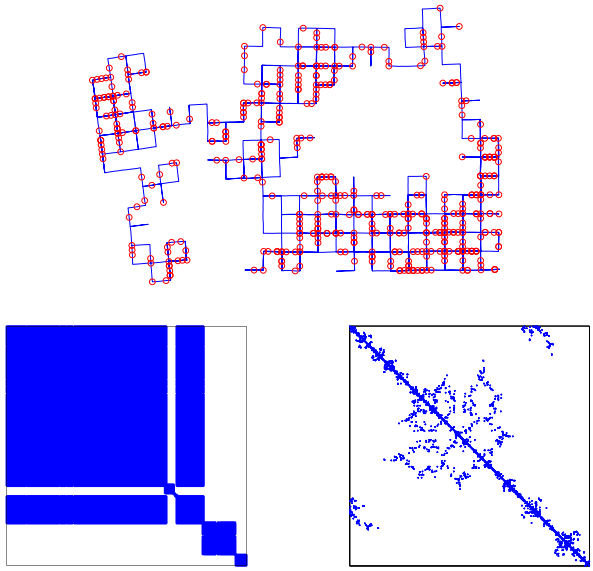

Fig. 1: In the absence of node reduction strategies, the size of the SLAM graph will grow without bound. **Top:** The Manhattan 3500 dataset [3] with 3500 poses in $SE(2)$ and 5599 constraints. The blue edges denote the entire dataset and the red circles are 500 poses that have been subselected. **Bottom Left:** The removal of the 3000 poses induces density in the graph. Here we show the sparsity structure of the resulting information information matrix. **Bottom Right:** In order to maintain computational efficiency, the dense representation is approximated by a sparse one. In this work we propose to choose the nodes to remove so that the sparse representation is as close as possible (in the KL divergence sense) to the dense one.

Notwithstanding the efficiency of these graph-SLAM approaches, they are not directly applicable to large-scale problems because resources such as memory, computation, and communication in multi-robot systems do not scale worse than constantly. For example, communication bandwidth through acoustics available to AUVs is typically very limited [4]. As a result, without removing states from the system, these graph-based methods will lead to resource constraint violation as mission duration and operational area increase.

To address this issue, recent work has focused either on how to select which nodes to remove, or on how to maintain sparse connectivity between nodes, but rarely *both*. In particular, marginalization is often used to remove nodes, which is achieved through Schur complement on the Hessian (information) matrix. Note that marginalization enforces a "node constraint" on the total number of variables. How-

ever, this process induces density and significantly increases the overhead of communication bandwidth or computation complexity, which in effect motivates edge sparsification [5], [6], [7]. Specifically, a Kullback-Liebler divergence (KLD) minimization is formulated to find a sparse information matrix to approximate the original dense one. Note that sparsification allows us to satisfy an "edge constraint" or "density constraint", for example, for computational efficiency and/or bandwidth considerations. However, this process is approximate and the penalty that we pay can be quantified in terms of KLD between the dense true distribution and its sparse approximation.

For a given application, a problem-specific approach to select and remove nodes may exist. For instance, one recent work proposed removing the nodes at which there is a minimum probability of collision with the environment for a navigation objective [8]. However, we argue that in the absence of problem-specific node selection strategies (i.e., every node is an equal candidate for removal), the optimal choice of nodes to remove through *marginalization* in order to satisfy node number constraints are the ones that will incur the minimum penalty in the subsequent *sparsification* to meet the edge constraints.

These two operations, marginalization and sparsification, are usually treated as distinct. The algorithms of node removal via marginalization do not consider the attainable performance of the subsequent sparsification, and conversely, the sparsification approaches are agnostic to the node selection used to choose which nodes to marginalize. In this work, we tightly couple these two processes into a single unified optimization framework, whose objective is to minimize the information loss due to graph reduction while being constrained by limited resources available. In particular, our proposed unified framework aims to optimally account for the resource constraints of *computation*, *memory*, and *communication bandwidth*. To prove this concept, after formulating the problem in the general sense, we provide one tractable solution instance. To validate this solution, we compare against the node selection strategies available in the literature and show that in our case the constraints are met with less penalty in terms of KLD between the dense distribution and the sparse approximation.

II. Related Work

Graph reduction algorithms can be categorized into two classes: (i) selecting which measurements and/or variables to discard, and (ii) marginalizing variables and then sparsifying measurements. In what follows, we briefly review these two clusters of literature.

A. Measurement/variable selection

1) Measurement selection: The basic idea of most measurement selection approaches is to evaluate the relative "informativeness" of measurements themselves and then discard the least useful. In particular, Kretzschmar et al. [9] introduced pose-graph compression for laser-based SLAM, in which nodes are selected for removal based on the amount of new information provided by their respective laser scans. The less informative scans are removed and then the associated poses are marginalized followed by a Chow-Liu tree (CLT)-based [10] approximation to regain sparsity. Similarly, Ila et al. [11] provided a relative information metric to evaluate whether edges should even be added to the pose graph in the first place, as well as to remove uninformative loop closure constraints.

2) Variable selection: The question of how to select a subset of variables to better support localization and/or mapping operation has been investigated. In [12], a Euclidean distance criterion is employed for node removal to ensure that the size of the state vector grows only with the size of the mapped environment. However, this approach does not bound the number of measurements. Similarly, downsampling features based on a visual-saliency measure in vision-based navigation systems has also been explored in order to improve loop closing [13], [14], [15]. In appearance-based visual SLAM approaches a similar problem is framed as "dictionary learning" where the size of the dictionary must be reduced. For example, in a online "sparsity-cognizant" approach to dictionary learning was proposed by Latif et. al. [16]. Other work considers variable selection to support the objective of navigation. For example, Strasdat et al. [17] introduced a reinforcement learning based landmark selection policy to minimize the robot position error at the goal. Lerner et al. [18] considered single camera frame based landmark selection in terms of a "severity function." And Sala et al. [19] chose the minimal set of landmarks that are viewable from every point in the configuration space. Moreover, Mu et al. [8] recently proposed a single framework for both landmark and measurement selection to support navigation. Other landmark and measurement selection techniques are also available but task-specific, e.g., uniform landmark selection [20] and entropy-based landmark selection [21] for the localization and mapping objective, as well as an incremental approach [22]. By contrast, in this work, we address the variable selection strategy by explicitly considering the resource constraints to be satisfied and choose the variables whose subsequent removal (through marginalization and sparsification) will incur the least information loss.

B. Node marginalization and edge sparsfication

Since marginalization induces dense connectivity across the Markov blanket of the marginalized node, recent research efforts have been devoted to further reduce the edges of the graph. In particular, Vial et al. [23] are among the first to formulate this sparsification problem as a convex optimization that minimizes the KLD between the dense distribution and its sparse approximation. In our prior work [5], we further regularize this formulation with ℓ_1-norm, which is appealing in its flexibility as it does not commit to any sparse graph structure. However, one challenge with this approach is that direct control over the structure of the resulting sparsified information matrix is lost. To mitigate this issue, Carlevaris and Eustice [24] introduced generic linear constraints (GLCs)

to approximate the dense factors induced by marginalization based on the CLT approximation. Most recently, Mazuran et al. [7], [25] improve the previous results by allowing nonlinear measurements to approximate the dense factors with "virtual" measurements which can be defined arbitrarily and then insightfully formulating the convex optimization over the *measurement*, rather than *state*, information matrix and proving that it remains convex. However, designing these virtual measurements is nontrivial and task specific.

It is important to note that none of these approaches provides any insight into how nodes should be selected to be marginalized, although this choice of nodes has a significant impact on the optimal KLD that is attainable in the sparsification stage. Moreover, they do not explicitly consider constraints other than computation, which clearly is not adequate for real robotic systems since other key resource constraints such as memory and communication cannot be ignored. In our recent work [4], communication constraints were taken into account in building multi-AUV SLAM systems. Specifically, marginalization is performed over the robot poses so that only maps are communicated to save communication throughput, and then the dense map connectivity is sparsified using a convex optimization similar to [7]. In this work, building upon our prior work [5], [4], we propose a unified framework to incorporate both edge (bandwidth or computation) and node (memory) constraints.

III. Problem Formulation

Let $\mathbf{X} = [\mathbf{x}_1^T, \cdots, \mathbf{x}_N^T]^T$ be the set of states (robot poses and/or landmark positions) that we seek to estimate, and $\mathbf{Z} = [\mathbf{z}_1^T, \cdots, \mathbf{z}_M^T]^T$ be the set of conditionally independent measurements. By assuming that $p(\mathbf{z}_j|\mathbf{X}) = p(\mathbf{z}_j|\mathcal{X}_j)$, i.e., $\mathcal{X}_j$ is the subset of states that are constrained by measurement $\mathbf{z}_j$, we can write the measurement model as follows:

$$\mathbf{z}_j = h_j(\mathcal{X}_j) + \eta_j \ , \quad \eta_j \sim \mathcal{N}(0, D_j^{-1}) \tag{1}$$

where we assume additive white Gaussian noise.

In graph SLAM, we aim to to find the most likely configuration of the states $\mathbf{X}$ given the measurements that we have made (i.e., maximum likelihood estimation or MLE). This problem can be shown to be equivalent to the following nonlinear least-squares (NLS) problem [1]:

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \sum_{j=1}^{M} ||\mathbf{z}_j - h_j(\mathcal{X}_j)||^2_{D_j^{-1}} \tag{2}$$

where $||\mathbf{e}||_\Sigma$ denotes the Mahalanobis distance (energy norm) and $D_j \in \mathcal{R}^{|\mathbf{z}_j| \times |\mathbf{z}_j|}$ is the measurement noise information matrix. To solve (2), due to the nonlinearity of measurement model (1), an iterative algorithm such as Gauss-Newton is often employed. Specifically, starting from an initial guess $\hat{\mathbf{X}}^{(0)}$, we iteratively solve for the (locally) optimal error state (increment) which is then used to update the state estimate:

$$\delta\mathbf{X}^{(k+1)} = \underset{\delta\mathbf{X}}{\operatorname{argmin}} \sum_{j=1}^{M} ||\mathbf{z}_j - h(\hat{\mathcal{X}}_j^{(k)}) - H_j^{(k)} \delta\mathbf{X}||^2_{D_k^{-1}} \tag{3}$$

$$\hat{\mathbf{X}}^{(k+1)} = \hat{\mathbf{X}}^{(k)} + \delta\mathbf{X}^{(k+1)} \tag{4}$$

where

$$H_j^{(k)} = \frac{\partial h(\mathcal{X}_j)}{\partial \mathbf{X}}|_{\mathbf{X}=\hat{\mathbf{X}}^{(k)}} \ \in \mathcal{R}^{|z_j| \times N}$$

is the measurement Jacobian evaluated at the current state estimate in the k-th iteration. In solving (3), an information (Hessian) matrix is typically required, which is given by:

$$\mathcal{I} = \sum_{j=1}^{M} H_j^T D_j H_j = H^T D H$$
$$\text{with} \quad H \triangleq \begin{bmatrix} H_1 \\ \vdots \\ H_M \end{bmatrix} , \ D \triangleq \begin{bmatrix} D_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & D_M \end{bmatrix} \tag{5}$$

It is important to note that the sparsity pattern of $\mathcal{I}$ corresponds exactly to the connectivity in the graph, i.e., the information matrix encodes the conditional dependence:

$$\mathcal{I}_{[i,j]} \begin{cases} \neq 0 & \exists \mathbf{z}_k | (\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}_k) \\ = 0 & \text{otherwise} \end{cases} \tag{6}$$

Consequently, we can evaluate the number of distinct non-zeros that will appear in $\mathcal{I}$ without ever having to calculate it using the following iterative equation:

$$||\mathcal{I}||_0 = \sum_{j=1}^{M} |\mathcal{P}_2(\mathcal{X}_j)| - |\mathcal{P}_2(\mathcal{X}_j) \bigcap \{ \bigcup_{i=1}^{j-1} \mathcal{P}_2(\mathcal{X}_i) \}| \tag{7}$$

where $\mathcal{P}_2$ is the subset of power set of cardinality at most 2.

It is clear that the size of the NLS problem (3)-(4) grows as new robot poses and/or landmark positions are added into the graph. The graph will eventually become prohibitively large for real-time performance, thus necessitating graph reduction.

A. Marginalization of nodes

To reduce the graph, marginalization is often used to remove nodes (i.e., reduce the size of the state space N). To this end, we partition all states into two subsets: the states we wish to keep, $\mathbf{X}_R$, and the states we wish to remove, $\mathbf{X}_M$. Marginalization over the canonical parametrization of the Gaussian distribution is performed via Schur complement:

$$\mathcal{I}_d = \mathcal{I}_{RR} - \mathcal{I}_{RM} \mathcal{I}_{MM}^{-1} \mathcal{I}_{MR} \tag{8}$$

$$\mathcal{I} = \begin{bmatrix} \mathcal{I}_{RR} & \mathcal{I}_{MR} \\ \mathcal{I}_{RM} & \mathcal{I}_{MM} \end{bmatrix} \tag{9}$$

where $\mathcal{I}_d$ in general becomes more dense than the original block matrix $\mathcal{I}_{RR}$. For subsequent optimization, a new set of *dense* factors can be generated using this dense information matrix $\mathcal{I}_d$ as well as the current estimate of the removed states $\hat{\mathbf{X}}_R$ [5], [24], [7].

B. Sparsification of edges

While marginalization reduces the size of the graph, it adversely increases the density of the graph. To further reduce the graph, one approach is to replace the dense distribution over the Markov blanket (subgraph) with a sparse approximation, for example, using the CLT approximation which is the optimal minimal yet connected distribution [24].

However, this approach does not guarantee consistency (i.e., information might be added to the graph) and the tree structure may not be desirable. Alternatively, one seeks to solve for a sparse approximation based on the convex optimization of minimizing KLD between the original dense distribution and the new sparse one [23]:

$$\begin{aligned} &\min_{\mathcal{I}_s \in \mathcal{S}_{++}} \mathcal{D}_{KL}(\mathcal{N}(\hat{\mathbf{X}}, \mathcal{I}_s^{-1}) || \mathcal{N}(\hat{\mathbf{X}}, \mathcal{I}_d^{-1})) \\ &= \min_{\mathcal{I}_s \in \mathcal{S}_{++}} \text{tr}(\mathcal{I}_s \mathcal{I}_d^{-1}) - \ln(|\mathcal{I}_s|) \end{aligned} \tag{10}$$

This method has the advantage that conservativeness can be enforced through the additional constraint $\mathcal{I}_s \preceq \mathcal{I}_d$. However, it remains open how to select the edges to remove. Options include again to choose the CLT structure [6], to enforce sparsity through sparsity regularization [5] or use problem-specific predefined graph structure [4].

IV. Resource-Constrained Graph SLAM

In this section, we propose a unified optimization framework that seeks to find an optimal reduced graph with respect to both nodes and edges, while meeting all resource requirements by explicitly expressing them as the constraints imposed onto the pertinent optimization variables.

A. Edge constraints

The number of edges (measurements) in the graph can be seen as directly aligning with a bandwidth constraint in the case of multi-robot systems [4], but it also connects to the computation required to solve for the MLE estimate [see (3) and (4)]. To derive the exact computation required as a function of the edge density, or fill-in, is challenging (if not impossible) since it is heavily impacted by other factors such as initial estimate, nonlinearity of measurement functions, and so on. However, in general, the efficiency of NLS solvers largely depends on the fill-in of the information matrix, which impacts the efficiency of back-substitution and covariance recovery [2]. Based on this key observation, we formulate the following reduction problem with graph density as the computation constraint:

Problem 1. ***Graph Density as Computation Constraint***

$$\begin{aligned} &\min_{\mathcal{I}_s \in \mathcal{S}^+} \mathcal{D}_{KL}(\mathcal{N}(\hat{\mathbf{X}}, \mathcal{I}_d^{-1}) || \mathcal{N}(\hat{\mathbf{X}}, \mathcal{I}_s^{-1})) \\ &\quad \textit{s.t.} \ \frac{||\mathcal{I}_s||_0}{N} \leq \kappa_{density} \end{aligned} \tag{11}$$

where $\mathcal{I}_s$ is the sparse information matrix, $\mathcal{I}_d$ is the dense information matrix, $\hat{\mathbf{X}}$ is the most recent MAP estimate, N is the dimensionality (number of nodes times dimension of each node) and $\kappa_{density}$ is the edge density constraint.

In the case of multi-robot distributed systems, we also consider the bandwidth as a finite resource. In this case, it is advantageous to consider a variation on (10) is to formulate the minimization over the measurement information of the new "virtual measurements" D_s [7], which are related to the sparse state information through $\mathcal{I}_s = H_s^T D_s H_s$. As proposed in [4] if $||D_s||_0 \leq ||\mathcal{I}_s||_0$, where $||\mathcal{I}_s||_0$ is incrementally calculated through (5), and the virtual measurement functions are known to all robots (presumably agreed upon at the start) then it is advantageous to only send the non-zero values in D_s. This motivates the following bandwidth-constrained sparsification problem:

Problem 2. ***Edge Number as Bandwidth Constraint***

$$\begin{aligned} &\min_{D_s \in \mathcal{D}} \mathcal{D}_{KL}(\mathcal{N}(\hat{\mathbf{X}}, \mathcal{I}_d^{-1}) || \mathcal{N}(\hat{\mathbf{X}}, (H_s^T D_s H_s)^{-1})) \\ &\quad \textit{s.t.} \ ||D_s||_0 \leq \kappa_{bandwidth} \end{aligned} \tag{12}$$

where $\mathcal{D}$ is the set of block diagonal positive definite matrices that correspond to the block structure of the measurements.

It should be pointed out that the actual Jacobian matrices, H_s can be computed by the receiver, since the structure of the nonlinear measurements is known and the linearization points, $\hat{\mathbf{X}}_R$, are also sent [4]. Consequently, the sparse information matrix can be recovered.

B. Variable constraint

In general, the number of nodes, and thus the size of the NLS problem (2), grows without bound as robot(s) operate. Therefore, it is necessary to remove nodes to retain constant-time scalability and enable long-term operation of mobile robots in unknown environments. To this end, we effectively impose an upper bound, κ_{node}, on the number of nodes that can be contained in the graph.

Problem 3. ***Node Number as Memory Constraint***

$$\begin{aligned} &\min_{\mathbf{X}_R \subset \mathbf{X}} f(\mathbf{X}_R) \\ &\quad \textit{s.t.} \ ||\mathbf{X}_R||_0 \leq \kappa_{node} \end{aligned} \tag{13}$$

where $\mathbf{X}_R$ is a subset of the entire set of variables, and $f(\cdot)$ is an objective function that is a design choice.

The cost function $f(\cdot)$ may be designed based on some application-specific requirements. For example, it can be mutual information [9], functions of euclidean distance [12], mutual information of associated sensor data, or probability of collision with obstacles [8]. We propose an alternative definition for the function $f(\cdot)$ as is detailed in the following subsection.

C. Unified optimization with edge and variable constraints

The central idea behind this work is that in the absence of front-end data considerations or problem-specific node removal strategies, the best way to choose the nodes to remove is *to select the ones that will induce the least penalty in the subsequent sparsification.* Hence, we combine Problems 1, 2 and 3 to formulate one *unified* optimization problem constrained by both edge and variable requirements. In particular, by defining the objective function $f(\cdot)$ in Problem 3 by the output of the KLD minimization of the sparsification problems, we get the following unified formulation:

Problem 4. ***Resource-Constrained Graph Reduction***

$$\min_{\mathbf{X}_R \subset \mathbf{X}} \left\{ \min_{\mathcal{I}_s \in \mathcal{S}^+} \mathcal{D}_{kl}(\mathcal{N}(\hat{\mathbf{X}}, \mathcal{I}_d^{-1}) || \mathcal{N}(\hat{\mathbf{X}}, \mathcal{I}_s^{-1})) \right. \\ \text{s.t. } ||\mathcal{I}_s||_0 \leq \kappa_{density} \\ \left. ||D_s||_0 \leq \kappa_{bandwidth} \right\} \\ \text{s.t. } ||\mathbf{X}_R||_0 \leq \kappa_{node} \tag{14}$$

D_s and $\mathcal{I}_s$ are related through $\mathcal{I}_s = H_s^T D_s H_s$, and the node subset determines the information matrix partitioning in the calculation of $\mathcal{I}_d$ as given by (8)*.*

Both of the edge constraints, $\kappa_{density}$ and $\kappa_{bandwidth}$, and the node constraint, κ_{node}, are enforced in Problem 4. The inequalities in (13) and (14) can be treated as equalities in the case that we wish to enforce that the resources are fully utilized.

V. Solving Problem 4

Due to the combinatorial nature, it is in general computationally intractable to solve Problem 4 analytically. To mitigate this issue, let us first turn our attention to the inner optimization which is convex, and then to the outer optimization which is combinatorial.

A. Solving the inner convex optimization

Given a potential subset of nodes $\mathbf{X}_R$, one standard approach for solving this problem is to enforce sparsity through ℓ_1-regularization on the information matrix. This is done by adding the term $\lambda||\mathcal{I}_s||_1$ to the objective function in Problem 1 and removing the constraint. The ℓ_1 norm is the closest convex relaxation of the ℓ_0 norm but is known to promote sparsity, where the tuning parameter λ determines the level of sparsity. This can be solved by an interior point method or using the alternating direction method of multipliers (ADMM) [5]. While this is an appealing approach, it is still preclusively slow since the optimization will have to iterate to convergence for every node that is evaluated.

Instead, we adopt the formulation for sparsification via minimizing the measurement information matrix as shown in Problem 2. In this case, we have direct control over the design of the Jacobian matrix H_s, and the block structure of the measurements, encoded in $\mathcal{D}_s$ that together will determine the resulting sparsity. As such, we can satisfy the edge constraint by construction through appropriate specification of these matrices. Moreover, it is shown in [25] that in the case that H_s is invertible, the optimal measurement informations are computable in closed form:

$$D_i = (\{H_s \mathcal{I}_d^{-1} H_s^T\}_i)^{-1} \tag{15}$$

where the $\{\cdot\}_i$ selects the ith block of the inner matrix. The resulting information matrix is given by $\mathcal{I}_s = H_s D_s H_S^T$ with

$$D_s = \begin{bmatrix} D_1 & 0 & \cdots & 0 \\ 0 & D_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & D_K \end{bmatrix} \tag{16}$$

We now build the Jacobian matrix that consists of relative pose-pose or pose-landmark measurements over the CLT across the Markov blanket of the node(s) being removed, which is guaranteed to be full rank and square (and hence non-singular). We can then add additional edges as permitted by adding correlations between these measurements in the block measurement information structure [25].

Although not necessarily explicitly stated, current approaches such as [25] marginalize one node at a time, performing a sparsification after each one. We note here that neither the resulting graph topology, nor the approximation are independent of the node elimination order in this case. Moreover, it is impossible to enforce hard global sparsity restrictions this way. Instead sparsity is enforced locally upon removal of each node. In contrast, here, after selecting the nodes designated for removal, we eliminate them all simultaneously. In the case where one node is inside the Markov blanket of another (and necessarily vice-versa) then the Markov blankets should be merged. As a result, the resulting graph topology after marginalization and sparsification is unique and optimal (as a result of the CLT optimality) and actually can be more sparse than the resulting graph after incremental node removal, which imposes a separate local CLT structure on each node as it is removed.

We proceed as follows. We begin by using the known graph topology and the candidate nodes $\mathbf{X}_R$ to generate a set of distinct Markov blankets $\mathbf{X}^{\{p\}} \subset \mathbf{X}_R$, $p = 1, \cdots, P$, with $1 \leq P \leq |\mathbf{X}_M|$ where $|\mathbf{X}_M|$ is the number of nodes to be removed times the dimension of an individual node. We perform a Schur complement to generate $\mathcal{I}_d$ once using (8) but then decompose the result into the individual dense marginal information matrices for each Markov blanket, $\mathcal{I}_d^{\{p\}}$. For each Markov blanket, we perform a CLT decomposition and generate a non-singular Jacobian $H_s^{\{p\}}$ consisting of relative pose-pose or pose-landmark measurements over the CLT evaluated at the current estimates of the nodes, $\hat{\mathbf{X}}^{\{p\}}$. We subsequently solve for the block measurement informations using (15). We can solve for the minimum KLD by computing the sparse information matrix over the Markov blanket and re-inserting it into the KLD objective function (10). Finally, the total KLD for this node combination can be evaluated by summing the individual KLDs since the estimates of the variables in $\mathbf{X}_R$ but not in any Markov blanket will remain constant. As a result, the function $f(\mathbf{X}_R)$ from Problem 3 can be expressed as:

$$f(\mathbf{X}_R) = \sum_{p=1}^{P} \log\det(H_s^{\{p\}} D_s^{\{p\}} (H_s^{\{p\}})^T) \\ + \text{tr}(H_s^{\{p\}} D_s^{\{p\}} (H_s^{\{p\}})^T (\mathcal{I}_d^{\{p\}})^{-1}) \tag{17}$$

where the edge constraint satisfaction is explicitly guaranteed through the design of the Jacobians and block structure of the measurements. Note that we could also optionally guarantee consistency by efficiently projecting into the consistency space using an eigendecomposition of the small local information matrices [4]. The algorithm for evaluating a candidate solution is summarized in Algorithm 1.

Algorithm 1 Solving the inner optimization in Problem 4 for one candidate node subset $\mathbf{X}_R$

Input: $\mathbf{X}_R$ - the set of nodes that should be retained
$\hat{\mathbf{X}}_R$ - the current map estimates of the nodes in $\mathbf{X}_R$
$\mathcal{I}$ - the full information matrix
$\kappa_{density}$, $\kappa_{bandwidth}$ - the edge contraints
Output: $\mathcal{D}^*_{KL}$

$\mathcal{I} = \begin{bmatrix} \mathcal{I}_{RR} & \mathcal{I}_{MR} \\ \mathcal{I}_{RM} & \mathcal{I}_{MM} \end{bmatrix}$
$\mathcal{I}_d = \mathcal{I}_{RR} - \mathcal{I}_{RM}\mathcal{I}_{MM}^{-1}\mathcal{I}_{MR}$
Extract the P separable Markov blankets $\mathbf{X}^{\{p\}}, p = 1..P$ based on nodes to remove $\mathbf{X}_R$ and connectivity encoded in $\mathcal{I}$
$\mathcal{D}^*_{KL} \leftarrow 0$
for all $p = 1, \ldots, P$ **do**
  $\mathcal{I}_d^{\{p\}} \leftarrow$ block information matrix according to $\mathbf{X}^{\{p\}}$
  Find minimum spanning tree (CLT) for Markov blanket p
  $H_s^{\{p\}} \leftarrow$ Jacobians for pairwise measurements over spanning tree evaluated at $\mathbf{X}_R^{\{p\}}$
  (optional) greedily add conditional dependencies to measurements until reach $\kappa_{density}$ or $\kappa_{bandwidth}$
  increment $\mathcal{D}^*_{KL}$ by $\log\det(H_s^{\{p\}} D_s^{\{p\}} (H_s^{\{p\}})^T) + \mathrm{tr}(H_s^{\{p\}} D_s^{\{p\}} (H_s^{\{p\}})^T (\mathcal{I}_d^{\{p\}})^{-1})$
end for

B. Solving the outer combinatorial optimization

We solve the combinatorial outer optimization using a branch and bound method over the partial order of node subsets. Fig. 2 illustrates the process, where there are 8 nodes and 10 edges (unary factors don't count as edges). The corresponding partial order is shown in Fig. 3. Each row, N, in the partial order contains all possible combinations of N nodes. Edges in the partial order (with arrows as shown) correspond to a single node removal. We bias the search in the tree to follow paths minimizing the "edge cost" which is defined as follows:

Definition 1. ***(Edge Cost)*** *The edge cost is the number of edges added to the graph by removing an additional node (moving down one level in the partial order) assuming dense connectivity over the nodes in the Markov blanket*

The edge costs are labeled on the edges in the partial order in Fig. 3. These edge costs can be calculated quickly by looking at the sparsity pattern in the Schur complement sub-blocks, $\mathcal{I}_{RR}$ and $\mathcal{I}_{RM}\mathcal{I}_{MM}^{-1}\mathcal{I}_{MR}$. The matrix $\mathcal{I}_{RR}$ encodes the existing connectivity over the remaining nodes. Every non-zero component in $\mathcal{I}_{RM}\mathcal{I}_{MM}^{-1}\mathcal{I}_{MR}$ without a counterpart in $\mathcal{I}_{RR}$ denotes the addition of a new edge over the Markov blanket that did not previously exist.

One can immediately observe that these edge costs do no necessarily align with the least connected nodes, but instead the nodes whose Markov blankets have the densest connectivity. For example, consider removal of node $\mathbf{X}_7$ as shown in Fig. 2-bottom right. It is connected to nodes $\mathbf{X}_6$ and $\mathbf{X}_5$, but $\mathbf{X}_6$ and $\mathbf{X}_5$ are already connected, therefore the removal of $\mathbf{X}_7$ has an edge cost of -2 (Edges $5-7$ and $6-7$ are removed) which is the lowest even though there are nodes that are more minimally connected in the graph (node 3 is singly connected but its removal imposes an edge cost of only -1). Removal of node $\mathbf{X}_2$, on the other hand, has a connectivity of three but none of the nodes in the Markov blanket are previously connected. Therefore, removal of node $\mathbf{X}_2$ incurs an edge cost of 0, which will serve to increase the edge density since one node has been removed.

To search the tree we greedily explore nodes of the tree with smaller "edge costs" since these are more likely (although not guaranteed) to provide solutions that not have minimum KLD and also are able to meet the edge constraint.

The efficiency of branch and bound in this case is derived from the fact that we can quickly find a "good" solution, even

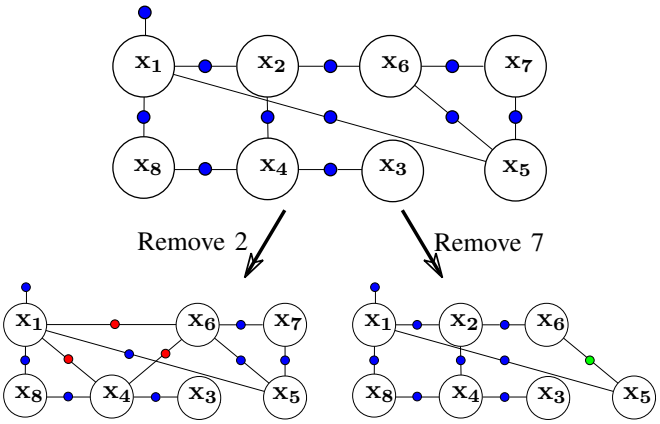

Fig. 2: Example of graph reduction. Large nodes constitute variables to be estimated. Small blue circles are constraints (factors) derived from sensor measurements. **Top:** Original graph with eight nodes and 10 edges. **Bottom Right:** Node $\mathbf{X}_7$ removed. Since the nodes connected to $\mathbf{X}_7$ were already connected, the constraint (shown in green) can be updated resulting in no new edges. Graph now has 7 nodes and 8 edges. **Bottom Left:** Node $\mathbf{X}_2$ is removed. 3 new edges (shown in red) are added over the Markov blanket which was previously not connected. Graph now has 7 nodes but 10 edges.

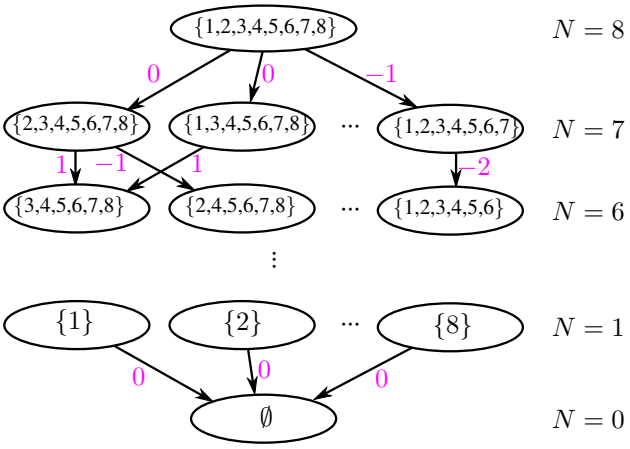

Fig. 3: The partial order over nodes in the graph corresponding to Fig. 2. Labels on edges correspond to edge costs.

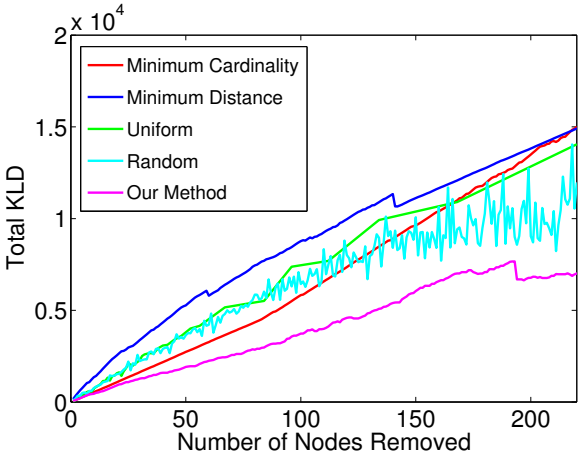

Fig. 4: The Kullback-Liebler divergence over the remaining nodes as a function of the number of nodes removed. The dataset used is a 670 node subset of the Manhattan dataset shown in Fig. 1. Five different node removal strategies are compared.

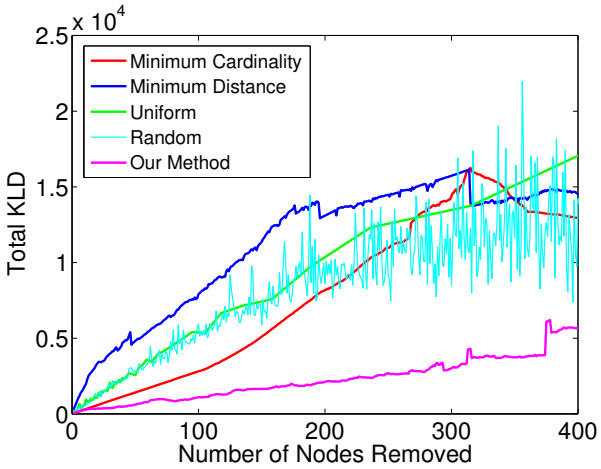

Fig. 5: The Kullback-Liebler divergence over the remaining nodes as a function of the number of nodes removed. The dataset used is the Intel dataset. The full dataset, nodes selected by our method, and the dense and sparse information matrices are shown in Fig. 6.

if it is not the best by using the minimum node cardinality heuristic and then updating when we find a better solution using the greedy strategy. Armed with a strong incumbent, we are able to rapidly prune potential solutions. In this case, if the evaluation of a candidate in the partial order that does not yet meet the node removal requirement incurs a higher minimum KLD as given by (17), then all subsets of this candidate can already be removed. For example, consider that we are tasked with removing two nodes from the graph in Fig. 2, and we have already evaluated that removal of nodes $\mathbf{X}_7$ and $\mathbf{X}_8$ results in a minimum KLD of 10 while meeting the edge requirements. Hypothetically, we then evaluate the minimum KLD for removal of only node $\mathbf{X}_2$ and it induces a minimum KLD of higher than 10, we need not evaluate any further candidates that contain $\mathbf{X}_2$ as a candidate node for removal. Note that monotonicity of the KLD is not guaranteed in this case, however it proves a good bound in practice and is able to rapidly reduce the size of the search space.

VI. Results & Discussion

We evaluate the proposed method on standard SLAM datasets. We compare against four other commonly-used node selection strategies:

- Uniform
- Smallest degree (i.e. the ones with the least connectivity in the graph)
- Most spatially redundant nodes as determined by the Euclidean distance to other nodes
- Random

Fig. 4 plots the KLD values as a function of the number of nodes removed for a subset of the Manhattan 3500 dataset (see Fig. 1) that contains 670 nodes and 1001 constraints. We can see that the minimum Euclidean distance approach induces the highest penalty since it tends to remove nodes that are densely connected in the graph. Uniform and random are roughly equal except when the number of nodes to be removed becomes large at which point uniform becomes highly sub-optimal since it will tend to maximize the number of distinct Markov blankets to be optimized. The minimum node cardinality approach is effective at a low number of nodes but then increases rapidly. It should be stated that we implemented no principled way to break ties in the case of minimum node cardinality. We can see that our approach performs the best in all cases. As an additional note, it was found that our approach would naturally tend to favor selecting connected nodes. In many cases we would select a fully connected set such that there was only one resulting Markov blanket and the resulting impact was minimized. In this case the KLD is roughly constant across all levels of node removal at a value roughly equal to the removal of single node. However, we deemed that practically speaking this is usually not an acceptable approach since large chunks of the pose graph are simply removed. As a result, we explicitly discouraged the selection of connected nodes until it was required due to the number of nodes being removed. Such a restriction is not imposed upon the other node selection strategies. Also note that the approach is guaranteed to maintain one connected graph at all times.

We also show results for the Intel dataset, which has 943 nodes and 1838 constraints. The total KLD as a function of number of nodes removed is shown in Fig. 5 and follows a similar pattern to Fig. 4. Additionally we show the nodes selected for the 400 node removal case on the dense dataset in Fig. 6. We show the dense and sparse approximate information matrices, which show a similar structure, however, the dense one contains less than half the number of non-zero entries.

We would agree that there are certainty other node selection strategies could have an application-specific purpose. However, we have provided a method to evaluate the cost of the particular node selection scheme so that a user may weigh the benefits of the chosen scheme against the penalty being paid compared an optimal node selection. It could also be possible to devise a node selection strategy that is a hybrid of

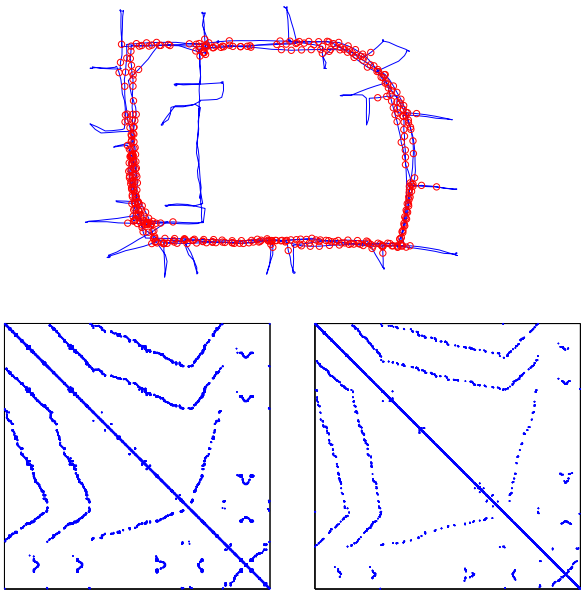

Fig. 6: The intel dataset. **Top:** The dense dataset (blue lines) and 400 selected nodes based on our proposed method (the end point of the Fig. 5). **Bottom Left:** The dense information matrix (39292 non-zero entries) **Bottom Right:** The sparse approximate information matrix (18576 non-zero entries).

the proposed scheme and any other. For example, one could select nodes that minimize KLD biased towards nodes with maximal Euclidean distance separation.

VII. Conclusion

We have presented a unified framework for resource-constrained graph reduction in SLAM. This approach (encapsulated by Problem 4) formulates the node removal decision as a subset selection problem whose objective function is the minimum penalty (in terms of KLD) that is paid to subsequently sparsify the graph after marginalization. As such it is able to encapsulate three major constraints in one single framework: the memory constraint (number of nodes), the computation constraint (density of graph), and the bandwidth constraint (number of edges to be transmitted in distributed system).

The next step for mobile robotics is operation in unknown large environments over extended or indefinite time scales. Achieving this objective requires an approach where resource consumption remains constant on average.

References

[1] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous location and mapping via square root information smoothing," *Int. J. of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.

[2] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *Robotics, IEEE Transactions on*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.

[3] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," in *in Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*, 2006, pp. 2262–2269.

[4] L. Paull, G. Huang, M. Seto, and J. J. Leonard, "Communication-constrained multi-AUV cooperative SLAM," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 6592–6599.

[5] G. Huang, M. Kaess, and J. Leonard, "Consistent sparsification for graph optimization," in *Mobile Robots (ECMR), 2013 European Conference on*, Sept 2013, pp. 150–157.

[6] N. Carlevaris-Bianco and R. M. Eustice, "Conservative edge sparsification for graph SLAM node removal," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 1–8.

[7] M. Mazuran, G. D. Tipaldi, L. Spinello, and W. Burgard, "Nonlinear graph sparsification for SLAM," in *Robotics Science and Systems Conference*, 2014.

[8] B. Mu, A. Agha-mohammadi, L. Paull, M. Graham, J. How, and J. J. Leonard, "Two-stage focused inference for resource-constrained collision-free navigation," in *Robotics Science and Systems*, 2015.

[9] 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.

[10] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[11] V. Ila, J. Porta, and J. Andrade-Cetto, "Information-based compact pose SLAM," *Robotics, IEEE Transactions on*, vol. 26, no. 1, pp. 78–93, Feb 2010.

[12] H. Johannsson, M. Kaess, M. Fallon, and J. Leonard, "Temporally scalable visual SLAM using a reduced pose graph," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 54–61.

[13] S. Frintrop and P. Jensfelt, "Attentional landmarks and active gaze control for visual SLAM," *Robotics, IEEE Transactions on*, vol. 24, no. 5, pp. 1054–1065, Oct 2008.

[14] A. Kim and R. M. Eustice, "Active visual SLAM for robotic area coverage: Theory and experiment," *International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 457–475, 2015.

[15] S. Hochdorfer and C. Schlegel, "Landmark rating and selection according to localization coverage: Addressing the challenge of lifelong operation of SLAM in service robots," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, Oct 2009, pp. 382–387.

[16] Y. Latif, G. Huang, J. Leonard, and J. Neira, "An online sparsity-cognizant loop-closure algorithm for visual navigation," in *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014.

[17] H. Strasdat, C. Stachniss, and W. Burgard, "Which landmark is useful? learning selection policies for navigation in unknown environments," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 1410–1415.

[18] R. Lerner, E. Rivlin, and I. Shimshoni, "Landmark selection for task-oriented navigation," *Robotics, IEEE Transactions on*, vol. 23, no. 3, pp. 494–505, June 2007.

[19] P. Sala, R. Sim, A. Shokoufandeh, and S. Dickinson, "Landmark selection for vision-based navigation," *Robotics, IEEE Transactions on*, vol. 22, no. 2, pp. 334–349, April 2006.

[20] G. Dissanayake, H. Durrant-Whyte, and T. Bailey, "A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem," in *Robotics and Automation (ICRA), 2000. IEEE International Conference on*, vol. 2, 2000, pp. 1009–1014 vol.2.

[21] S. Zhang, L. Xie, and M. Adams, "Entropy based feature selection scheme for real time simultaneous localization and map building," in *Intelligent Robots and Systems (IROS). 2005 IEEE/RSJ International Conference on*, Aug 2005, pp. 1175–1180.

[22] S. Choudhary, V. Indelman, H. Christensen, and F. Dellaert, "Information-based reduced landmark SLAM," in *International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4620–4627.

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

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

[25] M. Mazuran, W. Burgard, and G. D. Tipaldi, "Nonlinear factor recovery for long-term SLAM," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 50–72, 2015.