Exploiting Chordal Sparsity for Globally Optimal Estimation with Factor Graphs

Avinash Subramanian[1], Connor Holmes[2], Timothy D. Barfoot[2], Frank Dellaert[1], and Frederike Dümbgen[3]

***Abstract*—Robust and efficient state estimation is crucial for perception, navigation, and control in robotics. State estimation problems are conveniently modeled using the factor-graph framework as enabled by modern software packages such as *GTSAM* or *g2o*. However, the standard solvers included in such frameworks are local and may converge to poor local minima, posing significant safety concerns. Conversely, techniques based on convex relaxations have been shown to provide a means of globally solving or certifying many state estimation problems. However, these relaxations 1) often require substantial effort to formulate, and 2) may incur significantly higher cost compared to efficient local solvers, as they require solving a large semidefinite program (SDP). In this work, we address both shortcomings by 1) creating a new procedure within the *GTSAM* framework for automatically constructing convex SDP relaxations for any factor graphs with common factor and variable types, and by 2) exploiting the Bayes tree constructions native to *GTSAM* to decompose the SDP problem, leading to significant speedup in solver time for chordally sparse problems. We demonstrate the favorable scaling of this structure-exploiting *global* estimator compared to standard local solvers for two case studies: A 3D pose-graph SLAM problem with a ring factor graph and a 2D localization problem with a chain factor graph. The software framework is available at https://github.com/borglab/gtsam.**

I. Introduction

Fast and accurate state estimation is crucial for autonomous navigation and control of robots [1]. Factor graphs have emerged as a powerful framework, popular amongst robotics practitioners, for modeling complex estimation and sensor fusion problems: They are defined as a bipartite graph with edges between variable nodes (denoting states to be estimated) and factor nodes (denoting various measurements or priors) [2]. The factor graph modeling framework has been democratized by state-of-the-art software implementations such as *GTSAM* [3] and *g2o* [4]: These provide a comprehensive set of pose parameterizations, measurement and prior factor types, as well as efficient local optimization solvers such as Gauss-Newton and Levenberg-Marquardt for solving the estimation problem. While these local solvers are computationally efficient, they typically require high-quality initialization and do not provide guarantees of global optimality, which render them risky for safety-critical applications.

Recently, certifiably optimal state estimation techniques [5] have been proposed that leverage the fact that polynomial optimization problems often admit a tight convex relaxation in the form of an semidefinite program (SDP) [6], [7]. These methods provide *a posteriori* optimality guarantees and scale favorably compared to *always-optimal*, but worst-case exponential, branch-and-bound methods. However, the naive use of SDP solvers for the resulting relaxations may incur prohibitive costs. To address this issue, a number of related works have proposed techniques to exploit problem structure in the quest for a scalable certifiably global state estimator. Rosen et al. [8] introduced SE-Sync, which exploits the low-rank structure of SDP solutions through the Burer-Monteiro factorization [9] by applying the Riemannian staircase method. Tian et al. [10] extended this work to use a distributed algorithm via Riemannian block coordinate descent.

Other works have focused on exploiting the chain-like sparsity structure exhibited by a subset of estimation problems. Building on the seminal works from the SDP literature [11], [12], Duembgen et al. [13] introduced a distributed global estimator that can exploit chordal sparsity, and laid some foundations in using the factor-graph language to simplify problem formulation. However, the empirical validation and provided software were limited to specific problems and sparsity patterns, *i.e.*, localization with chain-like sparsity. Going beyond the (first-order) Shor's relaxation, Kang et al. [14] showcased a GPU-accelerated solver for trajectory optimization exploiting chain-like sparsity also for higher-order moment-hierarchy relaxations. Graesdal et al. [15] proposed a method for contact-rich motion planning where the same chain-like sparsity structure is exploited in the setting of graphs-of-convex-sets relaxations.

While some software packages that exploit chordal sparsity exist [16], [17], they come with software interfaces not necessarily suited for roboticists and do not always allow for control of the (non-unique) decomposition technique used. The aforementioned use cases in robotics instead manually decompose the chain-like problems, but this approach does not scale well to more complex sparsity patterns. Instead, we propose integrating chordal decomposition techniques into *GTSAM*, which is commonly used to formulate robotics problems, and is already equipped with variable elimination and chordal decomposition tools. In concurrent work, the fast SDP solver techniques based on the Riemannian staircases are also being integrated in *GTSAM* [18], which will significantly reduce the effort required for practitioners to investigate the efficacy of

[1]Avinash Subramanian and Frank Dellaert are with the College of Computing, Georgia Institute of Technology, Atlanta, GA, USA, asubramanian97@gatech.edu, frank.dellaert@cc.gatech.edu.

[2] Connor Holmes and Timothy Barfoot are with the Robotics Institute, University of Toronto, Toronto, ON, Canada, tim.barfoot@utoronto.ca, connor.holmes@utoronto.ca.

[3]Frederike Dümbgen is with the Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA, fduembgen@cmu.edu.

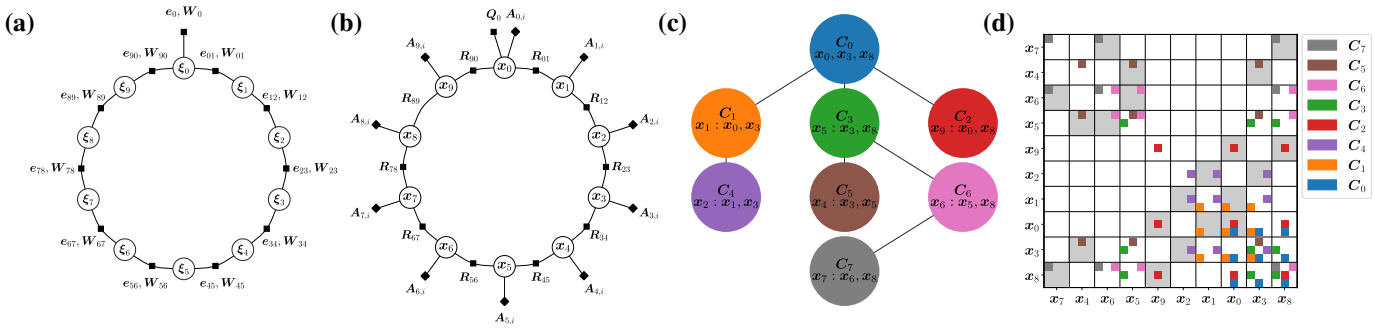

Fig. 1. Visualization of the proposed framework. (a) The user creates the factor graph representation of the problem (here, a 3D pose-graph SLAM problem with 10 variables is shown), (b) we automatically lift the factor graph to a quadratically-constrained quadratic program (QCQP) in lifted variables $\boldsymbol{x}_i$, (c) we construct the Bayes tree through variable elimination, which is then used to directly create a clique-based semidefinite program (SDP) problem (d): instead of solving for one large positive semidefinite (PSD) variable, we solve for smaller PSD variables (one per clique), shown by the corresponding colors.

alternative structure-exploiting solvers for their use cases. In summary, we make the following contributions:

1) We develop an easy-to-use *GTSAM*-based software framework that democratizes access to SDP-based certifiable estimators.
2) We show how to use variable elimination on a factor graph, and the resulting Bayes tree, to create a chordally decomposed estimator for arbitrary sparsity patterns beyond commonly treated chains.
3) We demonstrate the favorable scaling of this estimator on two problems of interest: A 3D pose-graph SLAM problem modeled as a ring factor graph with poses in $\mathrm{SE}(3)$, and a 2D localization problem modeled using a chain factor graph with poses in $\mathrm{SE}(2)$.

II. Method

In this section, we show how to obtain a computationally efficient convex relaxation for many canonical state estimation problems by leveraging both the graphical modeling capabilities and efficient factorization procedures provided by *GTSAM*. The workflow is visualized in Figure 1 for the 3D pose-graph SLAM problem with a ring factor-graph structure. In particular, we first construct the original estimation problems using a standard factor graph (Section II), then lift the problem to an equivalent Quadratically Constrained Quadratic Program (QCQP) (Section II-A), and finally perform variable elimination to construct a Bayes tree (Section II-B). The cliques of the Bayes tree correspond to the chordally decomposed SDP relaxation. Importantly, the entire process following the factor graph setup can be automated, resulting in an easy-to-use yet certifiably optimal estimation pipeline. Most common state estimation problems can be formulated as inference on a factor graph. State estimation involves determining the maximum a-posteriori estimate of the robot state from noisy measurements, equivalent to solving a nonlinear least squares optimization problem assuming zero-mean Gaussian noise and conditional independence:

$$\begin{aligned}\{\hat{\boldsymbol{\xi}}_i\}_{i=1}^N = \underset{\boldsymbol{\xi}_i\in\mathcal{X}, i\in[N]}{\arg\min} & \sum_{k\in\mathcal{Q}} \boldsymbol{e}_k(\boldsymbol{\xi}_k)^\top \boldsymbol{W}_k \boldsymbol{e}_k(\boldsymbol{\xi}_k) \\ & + \sum_{(i,j)\in\mathcal{R}} \boldsymbol{e}_{ij}(\boldsymbol{\xi}_i,\boldsymbol{\xi}_j)^\top \boldsymbol{W}_{ij} \boldsymbol{e}_{ij}(\boldsymbol{\xi}_i,\boldsymbol{\xi}_j),\end{aligned} \tag{1}$$

where $\boldsymbol{\xi}_i$, $i \in [N] := \{1,\dots,N\}$ represent the state variables to be estimated, with $\mathcal{X}$ often chosen to be $\mathrm{SO}(d)$ or $\mathrm{SE}(d)$. We denote by $\mathcal{Q}$ ($\mathcal{R}$) the set of indices (pairs) at which we have absolute (relative) measurements or prior terms. The cost terms of the first and second sum correspond to unary (absolute) factors and binary (odometry) factors, respectively.[1] These are defined by their respective error terms $\boldsymbol{e}_k(\cdot)$, $\boldsymbol{e}_{ij}(\cdot)$ and corresponding inverse covariance matrices $\boldsymbol{W}_k$, $\boldsymbol{W}_{ij}$. Problem (1) is a nonconvex optimization problem for nonlinear $\boldsymbol{e}_k$, $\boldsymbol{e}_{ij}$ or for a nonconvex set $\mathcal{X}$. Problem (1) can be represented by a factor graph $\mathcal{G}_f = (\mathcal{V},\mathcal{M},\mathcal{E})$ [2], where $\mathcal{V}$ contains all states, $\mathcal{M}$ contains the measurements or priors, and $\mathcal{E}$ defines the (bipartite) connectivity between $\mathcal{V}$ and $\mathcal{M}$.

Example — 3D SLAM with ring factor graph: Figure 1(a) presents the ring factor graph for an example 3D pose-graph SLAM problem. The variable nodes of the factor graph are the poses $\boldsymbol{\xi}_i : \boldsymbol{T}_{i0} \in \mathrm{SE}(3)$ to be estimated. The odometry (binary) factors are noisy relative pose measurements $\boldsymbol{e}_{ij} = \|\boldsymbol{T}_{i0} - \widetilde{\boldsymbol{T}}_{ij}\boldsymbol{T}_{j0}\|^2_{\sigma_{ij}}$ where $\widetilde{\boldsymbol{T}}_{ij}$ is a measurement of the relative transform from pose i to pose j and $\widetilde{\boldsymbol{T}}_{i0}$ a measurement of pose i. We are also given an absolute measurement on the initial pose, which forms a unary factor $\boldsymbol{e}_k = \|\boldsymbol{T}_{i0} - \widetilde{\boldsymbol{T}}_{i0}\|^2_{\sigma_i}$). We assume isotropic noise for all measurements thus $\boldsymbol{W}_{ij} = \sigma_{ij}^{-2}\boldsymbol{I}$ and $\boldsymbol{W}_k = \sigma_i^{-2}\boldsymbol{I}$.

A. Lifting to a QCQP

The variables and factors of Problem (1) are lifted to a higher-dimensional space to yield a QCQP for which standard SDP convex relaxations exist [19], [20]. Importantly, this operation leaves the graph's original connectivity unchanged. The intuition behind variable lifting is to introduce enough higher-order terms such that the cost and the constraints stemming from $\mathcal{X}$ in Problem (1) become quadratic in the lifted vectors. The state $\boldsymbol{\xi}_i$ yields the lifted variable

$$\boldsymbol{x}_i \in \mathbb{R}^D, \boldsymbol{x}_i := \begin{bmatrix}1 & \boldsymbol{\xi}_i^\top & \boldsymbol{\ell}_i(\boldsymbol{\xi}_i)^\top\end{bmatrix}^\top, \tag{2}$$

where we have introduced the lifting function $\boldsymbol{\ell}_i$ and the homogenization constant 1, which allows for constant and linear terms to be represented using quadratic terms as well.

[1]Note that we focus on unary and binary factors, which are the most common and the only types required for the presented examples. The framework does allow for any factor types.

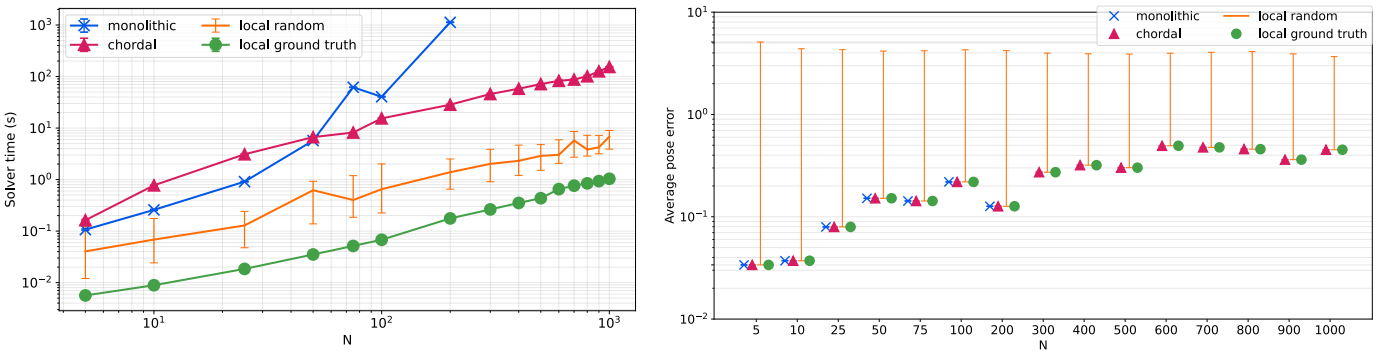

Fig. 2. Solver times (left) and Average pose error (right) for the 3D pose-graph SLAM problem with a ring factor graph as a function of the number of poses N. The presented *chordal* estimator has a complexity comparable with local solvers and with little variance in solution time and accuracy.

The QCQP reformulation of (1) takes the form:

$$\begin{aligned}\{\hat{\boldsymbol{x}}_k\}_{k\in\mathcal{V}} = &\underset{\boldsymbol{x}_k, k\in\mathcal{V}}{\arg\min} \sum_{k\in\mathcal{Q}} \boldsymbol{x}_k^\top \boldsymbol{Q}_k \boldsymbol{x}_k + \sum_{(i,j)\in\mathcal{R}} \boldsymbol{x}_{ij}^\top \boldsymbol{R}_{ij} \boldsymbol{x}_{ij} \\ &\text{s.t. } \boldsymbol{x}_k^\top \boldsymbol{A}_{j_k}^s \boldsymbol{x}_k = b_{j_k}, \quad k\in\mathcal{V}, j_k \in [N_k^s] \\ &\boldsymbol{x}_k^\top \boldsymbol{A}_{j_k}^p \boldsymbol{x}_k = 0, \quad k\in\mathcal{V}, j_k \in [N_k^p]\end{aligned} \quad , \quad (3)$$

where $\boldsymbol{x}_{ij}$ is the concatenation of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$.The constraint matrices $\boldsymbol{A}_{j_k}^s$ and scalars b_{j_k}, $j_k \in [N_k^s]$ enforce the substitution constraints from (2) for lifted variable k, and $\boldsymbol{A}_{j_k}^p$, $j_k \in [N_k^p]$ enforce the primary constraints from $\mathcal{X}$. The cost matrices $\boldsymbol{Q}_k$ and $\boldsymbol{R}_{ij}$ are derived in closed form for each of the measurement and prior types. This QCQP can be represented as a *lifted* factor graph as shown in Figure 1(b).

B. Bayes Tree Construction

By collecting all variables in one large vector $\boldsymbol{x}$, the QCQP could be relaxed to an SDP with variable $\boldsymbol{X} \succeq 0$ of size ND, obtained by forming the rank-one variable $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top$ and relaxing its rank constraint. For the detailed reformulation, we refer the reader to [13]. However, the cost to solve such an SDP would in general be approximately cubic in its dimension. We will refer to this formulation as the *monolithic* SDP.

Instead, we formulate an equivalent SDP problem with multiple, often significantly smaller, PSD variables. Starting from the QCQP factor graph, a variable elimination (factorization) procedure is executed using a specific ordering (*e.g.*, COLAMD or METIS) as explained in detail by Dellaert and Kaess [21, Ch. 4]. This yields a Bayes net, which is known to always be chordal [21]. The maximal cliques of the Bayes net can be listed in an undirected tree called a clique tree (also known as a junction tree) or a directed tree that preserves the order of eliminations called a Bayes tree [21]. The Bayes tree is defined as $\boldsymbol{T}_c := (\mathcal{V}_c, \mathcal{E}_c)$, where the vertices index the maximal cliques, and edges between two cliques indicate shared variables. Figure 1(c) illustrates the Bayes tree attained by performing variable elimination on the shown QCQP factor graph with a METIS ordering.

By associating T smaller PSD matrices $\boldsymbol{C}_t$, $t \in \mathcal{V}_c$ with the maximal cliques obtained from the Bayes tree, we arrive at a chordally decomposed SDP formulation:

$$\begin{aligned}\{\hat{\boldsymbol{C}}_i\}_{t\in\mathcal{V}_c} = &\underset{\boldsymbol{C}_1,\ldots,\boldsymbol{C}_T}{\arg\min} \sum_{t\in\mathcal{V}_c} \operatorname{tr}(\boldsymbol{Q}_t \boldsymbol{C}_t) \\ &\text{s.t. } \mathcal{A}_t(\boldsymbol{C}_t) = \boldsymbol{b}_t, \quad t\in\mathcal{V}_c \\ &\mathcal{S}_t^{t'}(\boldsymbol{C}_t) = \mathcal{S}_{t'}^{t}(\boldsymbol{C}_{t'}), \quad t,t' \in \mathcal{E}_c \\ &\boldsymbol{C}_t \succeq 0, \quad t\in\mathcal{V}_c\end{aligned} \quad , \quad (4)$$

where $\mathcal{A}_t$ is the linear operator stacking all primary, substitution, and possibly redundant constraints[2] of clique t , and $\mathcal{S}_t^{t'}$ is the operator selecting the shared variables with t' from clique t. $\boldsymbol{Q}_t$ is the clique-wise cost matrix, obtained from splitting the cost variables equally across overlapping nodes. The decomposition of the PSD variable into its cliques is shown in Figure 1(d).

We note that the fact that the chordally decomposed SDP Problem 4 is equivalent to the *monolithic* SDP is a consequence of the fundamental result by Fukuda et al. [23]: If the aggregate sparsity pattern of cost and constraint matrices corresponds to a chordal graph, the large PSD constraint $\boldsymbol{X} \succeq 0$ can be decomposed into a set of smaller PSD constraints on the maximal cliques of the graph, coupled by equality constraints on their intersections [24]. The key insight leveraged in our formulation is that the cliques of the Bayes tree correspond to the maximal cliques of the chordal completion of the sparsity pattern of the *monolithic* SDP.

C. Implementation

The lifting procedure and the formulation of the SDP problem are implemented in C++ using the MOSEK fusion API for C++ [25]. Similar to GTSAM's implementation, a template metaprogramming approach is used to generate specialized code at compile time for the different matrix Lie group types (currently `Rot2`, `Rot3`, `Pose2`, `Pose3` supported) and factor types (currently `FrobeniusBetween`, `FrobeniusPrior`, `PriorFactor` supported) of *GTSAM*.[3]

[2]Redundant constraints are irrelevant to the original problem but yield a tighter convex relaxation (closer to the global optimum) without changing the original feasible set [22]. We implicitly assume that redundant constraints do not change the problem sparsity. If redundant constraints add novel connections in the original factor graph, they must be added *before* the Bayes tree construction.

[3]Note that any factor and variable types that can be encoded with polynomial cost and constraints are suitable for the lifting procedure.

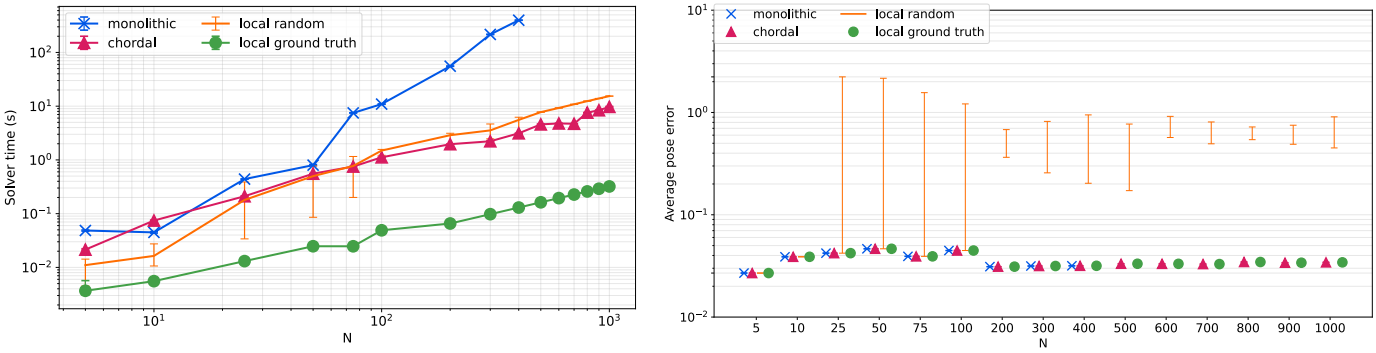

Fig. 3. Solver times (left) and Average pose error (right) for the 2D pose localization problem with a chain factor graph representation as a function of the number of poses N. The presented *chordal* estimator has complexity comparable to that of local solvers, with little variance in solution time and accuracy.

III. Results and Discussion

The proposed method is applied to a 3D pose-based SLAM problem represented by a ring factor graph and a 2D localization problem represented by a chain factor graph. For comparison of solver times and accuracy of estimated poses, three other estimation algorithms are implemented: The *monolithic* SDP estimator that does not exploit any sparsity structure but results in a certifiably global estimate, *local random* and *local ground truth*, which use *GTSAM*'s Levenberg-Marquardt nonlinear least-squares solver with starting values chosen as random poses or ground truth poses, respectively. All experiments are run on an Apple M1 MacBook Pro.

A. 3D pose-graph SLAM example

We first treat the 3D pose-graph SLAM problem with ring structure described in Section II. We study the scalability of the proposed chordally sparse SDP formulation as the number of poses N increases. All problems have the same fixed scale; thus, increasing N increases the sampling density of measurements. For each value of N, the sensor noise is randomly sampled and composed with the nominal measurement. Each of the 4 estimators is run five times with the same measurement, such that the only stochasticity arises from randomness in the estimation algorithm (in starting value or convergence). The empirical results in the left plot of Figure 2 show that the solution time of the *chordal* estimator grows (roughly) linearly in the number of poses, and is similar to the two local solvers, albeit with larger constant. The *local random* estimator is much slower than the *local ground truth* estimator and has a larger variance, indicating the dependence on the quality of the initialization. Lastly, we note that the *chordal* estimator has relatively low variance in solution time, which may be desirable for deployment on hardware for applications with strict time budgets.

The right pane of Figure 2 presents a box plot of the average pose error between the estimate and the ground truth, computed using the log-map norm ($\left\|\log\left(T_{\mathrm{gt},i}^{-1}T_i\right)\right\|_2$). The plot shows that the *chordal* estimator has similar accuracy to the *local ground truth* estimator, with the *local random* estimator often resulting in an order-of-magnitude higher average pose error. In addition, *local random* has a large variance in errors, indicating the peril of converging to spurious local minima after poor initialization. Conversely, the *chordal* estimator performs deterministic global optimization and thus has zero variance in average pose error.

B. 2D pose localization example

The 2D pose localization example we study has poses in $\mathrm{SE}(2)$, relative measurements between consecutive poses, and absolute measurements for each pose; the problem can thus be represented by a chain factor graph.

Figure 3 shows that, similar to the 3D pose-graph SLAM problem, the proposed *chordal* estimator has a comparable runtime complexity to that of local solvers, but it is even faster thanks to increased sparsity. For certain large estimation problems, the *chordal* estimator is even faster than the *local random* estimator, which is notable since it comes with global optimality guarantees since the convex relaxation is known to be tight. Lastly, we observe that the *chordal* estimator has low variance in solver time and always finds the global minimum, while the *local random* estimator often converges to poor local minima.

IV. Conclusions and Future Work

This work demonstrates a scalable certifiable estimator applicable to state estimation problems on arbitrary factor graphs (including those with a ring topology) composed of a subset of commonly used factor and variable types. We present an implementation in *GTSAM* to facilitate use by practitioners. Furthermore, we show that existing *GTSAM* functionality for constructing a Bayes tree automatically yields the chordal decomposition crucial for the estimator. We have shown, for two representative problems, that this chordal estimator has a runtime complexity that empirically is linear in the problem size, similar to local solvers, yet reliably finds certifiably global optima without requiring a high-quality initialization. Future work involves implementing the proposed lifting transformations for a broader range of variable and factor types that arise in different inference problems. A systematic comparison with the alternative structure-exploiting solver based on the Riemannian staircase [18], enabled by its *GTSAM* integration, is also planned. Finally, a promising extension is the implementation of a distributed version of this chordal estimator in *GTSAM*, leveraging splitting methods such as the Alternating Direction Method of Multipliers (ADMM) [26], [13].

ACKNOWLEDGMENT

The work was done in part while FD was with the WILLOW team, Inria Paris, ENS, PSL University and supported by the European Union, through the Horizon Europe research and innovation program under the Marie Skłodowska-Curie (GA no.101207106).

REFERENCES

[1] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2017.

[2] F. Dellaert, "Factor graphs : Exploiting structure in robotics," *Annual Review of Control, Robotics and Autonomous Systems*, vol. 4, pp. 141–166, 2021.

[3] ——, "Factor graphs and GTSAM: A hands-on introduction," *Georgia Institute of Technology, Tech. Rep*, vol. 2, no. 4, 2012.

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

[5] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert, "Initialization techniques for 3D SLAM: A survey on rotation estimation and its use in pose graph optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4597–4604.

[6] J. B. Lasserre, "Convergent SDP-Relaxations for Polynomial Optimization with Sparsity," in *Mathematical Software (ICMS)*. Berlin, Heidelberg: Springer, 2006, pp. 263–272.

[7] P. A. Parrilo, "Semidefinite programming relaxations for semialgebraic problems," *Mathematical Programming*, vol. 96, no. 2, pp. 293–320, 2003.

[8] 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," *International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 95–125, 2019.

[9] S. Burer and R. D. Monteiro, "Local Minima and Convergence in Low-Rank Semidefinite Programming," *Mathematical Programming*, vol. 103, no. 3, pp. 427–444, 2005, https://doi.org/10.1007/s10107-004-0564-1.

[10] Y. Tian, K. Khosoussi, D. M. Rosen, and J. P. How, "Distributed Certifiably Correct Pose-Graph Optimization," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 2137–2156, 2021.

[11] L. Vandenberghe, "Chordal Graphs and Semidefinite Optimization," *Foundations and Trends in Optimization*, vol. 1, no. 4, pp. 241–433, 2015.

[12] Y. Zheng, G. Fantuzzi, and A. Papachristodoulou, "Chordal and factor-width decompositions for scalable semidefinite and polynomial optimization," *Annual Reviews in Control*, vol. 52, pp. 243–279, 2021.

[13] F. Dümbgen, C. Holmes, and T. D. Barfoot, "Exploiting chordal sparsity for fast global optimality with application to localization," *Algorithmic Foundations of Robotics XVI (WAFR 2024)*, vol. 2, 2026, https://arxiv.org/abs/2406.02365.

[14] S. Kang, X. Xu, J. Sarva, L. Liang, and H. Yang, "Fast and Certifiable Trajectory Optimization," *Algorithmic Foundations of Robotics XVI (WAFR 2024)*, vol. 1, 2026.

[15] B. Graesdal, S. Chia, T. Marcucci, S. Morozov, A. Amice, P. Parrilo, and R. Tedrake, "Towards tight convex relaxations for contact-rich manipulation," *Robotics: Science and Systems XX*, 2024.

[16] M. Garstka, M. Cannon, and P. Goulart, "COSMO: A conic operator splitting method for convex conic problems," *Journal of Optimization Theory and Applications*, vol. 190, no. 3, pp. 779–810, 2021.

[17] J. Wang, V. Magron, and J.-B. Lasserre, "TSSOS: A Moment-SOS Hierarchy That Exploits Term Sparsity," *SIAM Journal on Optimization*, vol. 31, no. 1, pp. 30–58, 2021.

[18] Z. Xu, N. R. Sanderson, H. J. Zhang, and D. M. Rosen, "Certifiable estimation with factor graphs," *arXiv preprint arXiv:2603.01267*, 2026.

[19] J. B. Lasserre, *An Introduction to Polynomial and Semi-algebraic Optimization*. Cambridge University Press, 2015, vol. 52.

[20] J. Nie, *Moment and polynomial optimization*. SIAM, 2023.

[21] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Foundations and Trends in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.

[22] F. Dümbgen, C. Holmes, B. Agro, and T. D. Barfoot, "Toward Globally Optimal State Estimation Using Automatically Tightened Semidefinite Relaxations," *IEEE Transactions on Robotics*, vol. 40, pp. 4338–4358, 2024, http://arxiv.org/abs/2308.05783.

[23] M. Fukuda, M. Kojima, K. Murota, and K. Nakata, "Exploiting Sparsity in Semidefinite Programming via Matrix Completion I: General Framework," *SIAM Journal on Optimization*, vol. 11, no. 3, pp. 647–674, 2001.

[24] M. Garstka, M. Cannon, and P. Goulart, "A clique graph based merging strategy for decomposable SDPs," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 7355–7361, 2020.

[25] MOSEK Aps, "MOSEK Fusion API for C++," *Version*, vol. 9, p. 13, 2019.

[26] S. Boyd, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.