

Scalable Distance-based Multi-Agent Relative State Estimation via Block Multiconvex Optimization

Tianyue Wu and Fei Gao
Institute of Cyber-Systems and Control,
Zhejiang University, Hangzhou 310027
Email: {tianyueh8erobot, fgaoaa}@zju.edu.cn

Abstract—This paper explores the distance-based relative state estimation problem in large-scale systems, which is hard to solve effectively due to its high-dimensionality and non-convexity. In this paper, we alleviate this inherent hardness to simultaneously achieve scalability and robustness of inference on this problem. Our idea is launched from a universal geometric formulation, called *generalized graph realization*, for the distance-based relative state estimation problem. Based on this formulation, we introduce two collaborative optimization models, one of which is convex and thus globally solvable, and the other enables fast searching on non-convex landscapes to refine the solution offered by the convex one. Crucially, both models enjoy *multiconvex* and *decomposable* structures, allowing efficient and safe solutions using *block coordinate descent* that enjoys scalability and a distributed nature. The proposed algorithms collaborate to demonstrate superior or comparable solution accuracy to the current centralized convex relaxation-based methods, which are known for their high optimality. Distinctly, the proposed methods perform scalability beyond the reach of previous convex relaxation-based methods. We also demonstrate that the combination of the two proposed algorithms achieves a more robust pipeline than deploying the local search method alone in a continuous-time scenario.

I. INTRODUCTION

Relative state estimation is a fundamental problem in multi-agent systems, where individual agents try to estimate the state of other agents relative to themselves. Although there are notable approaches [1–6] proposed for solving this problem with a limited number of agents, the exploration of such techniques that can scale to tens or hundreds of agents remains largely uncharted. At the same time, control approaches built specifically for large-scale systems have emerged in recent years [7–9], yet under the protection of external localization devices (e.g., motion capture systems) due to the lack of reliable relative state estimation schemes that provide information about inter-agent collision and collaboration. Therefore, we are motivated to study the relative estimation problem in the context of large-scale systems.

Modern distance sensors like ultra-wideband (UWB) offer a lightweight yet effective choice for inter-agent measurements in large-scale systems: they can measure distances over long ranges, spanning tens to hundreds of meters, and accommodate a large number of tags with unique IDs. Despite the unique physical benefits from distance sensors when deployed for state estimation at scale, the corresponding optimization problem is highly non-convex and high-dimensional. As a result, inference on this problem is sensitive to initial guesses and

time-consuming. To overcome the sensitivity to initialization, semidefinite relaxation (SDR) methods are recently applied to find solutions for relevant problems [1, 10, 11]. Unfortunately, the centralized nature and scale-sensitive complexity prevent such methods from being applied to high-dimensional problems. A challenging question, therefore, is how to exploit the robustness of such relaxation models, while allowing it to be solved in a *scalable* way. Moreover, relaxation-based methods are based on some loose convexification of the problems under noisy distance measurements, i.e., the solutions of the relaxation models are deviated from the global optimum of the original problems. A common approach is to further *refine* the solutions using local search methods based on numerical optimization [10, 11], which are initialized with the solutions of the relaxation models. However, the highly non-convex landscape and high dimension leads to very different estimation accuracies and efficiency obtained by different choices of local search methods [4, 11] (see Section VII-A). Therefore, another important problem is to design a local search method that is both accurate and efficient to refine upstream solutions.

In this paper, we present one suite of such methods to address the above problems in a unified theoretical framework, which contribute two key ideas: (a) a universal formulation, *generalized graph realization*, for distance-proprioception and distance-only relative state estimation; (b) *multiconvex* [12] and *structurally decomposable* optimization models with solution by *block coordinate descent* (BCD) [13], which enable both scalable solution of a robust convex relaxation and a following accurate refinement. The key idea (a), generalized graph realization, is the recipe to bridge the distance-based state estimation problem with the idea (b). Its essence lies in a geometric description of rigid body states, inspired by the well known idea that *the state of a rigid body in space can be determined using 3 non-colinear points on it* [14]. The key idea (b) includes a convex relaxation and a local search model, both enjoying decomposable and multiconvex structures, and exploits the scalability of BCD. The decomposable structures allow natural instantiation of the BCD framework. The block multiconvexity [12] design helps BCD to be instantiated with efficiency and safety, in the sense that each subproblem in the BCD allows fast implementation of exact solutions and convergence of iteration sequence can be formally guaranteed. In particular, the introduced convex model, unlike the SDR in [1, 10, 11], is an *edge-based* convex relaxation [15, 16]

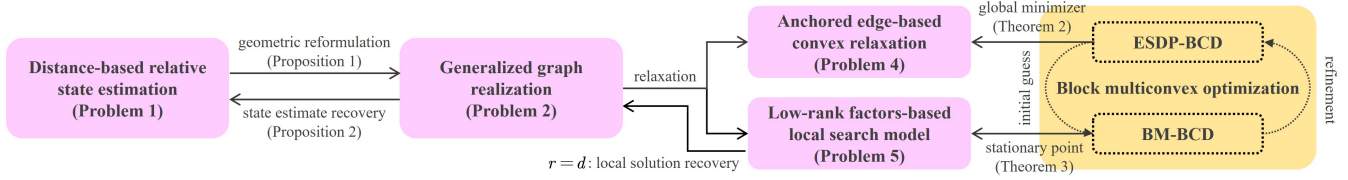


Fig. 1: **Relations between problems and algorithms in this work.** The Technical Roadmap section explains these problems and algorithms and corresponds them to the parts of the article.

featured by its decomposed constraints. The specialized local search model is used for accurately refining the crude solution by the convex relaxation model. The unique efficiency of this local search method also enables *online estimation* in continuous-time scenarios (see Section VII-C). Thanks to the BCD framework, our approach can be implemented in a distributed manner where each agent performs local computation and communication (see Section IV-B).

Our main contributions are as follows:

- 1) A generalized graph realization formulation for distance-based relative state estimation, supporting both distance-only and distance-proprioception setups.
- 2) Two multiconvex and decomposable optimization models, an edge-based relaxation model for initial guess-robust inference and a local search model for fast refinement; respective instantiations of the BCD framework on the two models, which result in two collaborative algorithms that are scalable and proved to converge in a unified theoretical framework.
- 3) Benchmark results of the proposed methods on large-scale problems with various sensor and system setups, which show that (i) the proposed two optimization methods collaborate to provide solutions with better or comparable precision against alternative methods, and (ii) the proposed methods exhibit scalability that other convex relaxation methods are naturally unavailable.
- 4) Demonstration of the convex relaxation-based method in a continuous-time scenario, where only high-frequency estimation by the local search method is not enough to provide time-consistent estimates.

Technical Roadmap. Fig. 1 illustrates the relations between problems and algorithms in this work. Section III mainly presents the transformation from the original problem (Problem 1) to the equivalent generalized graph realization problem (Problem 2 and Proposition 1), which is one of our main contributions. Section IV reviews the BCD algorithm, especially BCD on block multiconvex optimization that is used in this paper. Section V derives the optimization models derived from Problem 2 that BCD finally solves, which forms the first half of contribution 2). Section VI presents the implementation of BCD on the two optimization models, which leads to two collaborative algorithms, BM-BCD and ESDP-BCD, with convergence guarantees of their iteration sequences (Theorems 2 and 3) in a unified theoretical framework (Theorem 1). These results form the second half of contribution 2).

II. RELATED WORK

A. Relative State Estimation

A lot of previous works have studied the problem of relative state estimation within a few agents. Fusion of intra-agent (e.g., odometry) and inter-agent measurements is a popular paradigm for relative state estimation [1, 5, 6]. Nguyen and Xie [1] propose QCQP and SDP formulations for the relative transformation estimation problem in 4-DoF (assuming known roll and pitch angles measurements by IMUs) of VIO-UWB fusion. Ziegler et al. [17] are one of the few examples that employ distributed optimization techniques (in their work, ADMM) for scalably estimating relative states, where the setup is similar to [1]. However, the good performance of their method is based on the assumption that all agents are equipped with accurate odometry to provide near-perfect initial guesses.

Some works [2–4] propose to equip each agent with multiple distance sensors to achieve an odometry-free relative localization approach. These methods minimize the data throughput requirements and, with an elaborate calibration of the noise distribution, this setup can even achieve similar accuracy to those using odometry [2]. Although these works consider similar setups to ours, they have yet to explicitly address the computational challenges arising in the context of large-scale systems. The techniques introduced in this paper can be directly applied to the setups in those works and thus contribute in a complementary way with them.

B. Semidefinite Programming for State Estimation

1) *Pose Graph Optimization:* Recently, SDP has been successfully applied to the Pose Graph Optimization (PGO) problem. SE-Sync [18] and DC2-PGO [19] (a distributed variants of SE-Sync) employ the *Riemannian Staircase* (RS) approach [20] that allows to solve SDP efficiently using non-linear programming (NLP) algorithms. In particular, DC2PGO uses a BCD-type distributed solver [21] to solve the NLP generated by RS. Unfortunately, RS is not applicable to our problem since the semidefinite matrix in our formulation (see (13)) does not enjoy the distinctive geometry as those in [20], which is necessary for the adoption of RS. Instead, we employ a structurally decomposable relaxation model where the BCD framework can be applied naturally.

2) *Distance-based Robotic Localization:* Unlike in PGO problems, where vanilla rank relaxation methods can exhibit *guaranteed* tightness [22], this is often not the case in problems containing noisy distance measurements [23]. Halsted and Schwager [10] propose *Riemannian Elevator* to identify a rank-restricted space guaranteeing the existence of solutions for Shor relaxation. Papalia et al. [24] employ an improved

certification procedure of RS [25] to find a solution for the Shor relaxation of distance-aided SLAM. One crucial feature of these approaches is the introduction of edge directions [10] that leads to homogeneous QCQPs over nonlinear Riemannian manifolds. In parallel, Dümbsen et al. [26] derive the optimality certificate for the anchored, distance-only, and single-agent setup using a quadratic measurement model as in this paper. Interestingly, they empirically exhibit tightness of the relaxation at low noise levels. In a subsequent work [27], the authors develop a methodological approach to automatically find redundant constraints and ensure the tightness for SDR. In this paper, instead of focusing on the general tightness of SDR, we explore a further relaxation of the SDR [16] and provide some observation of practical significance of its tightness, as shown in Section V-B.

Another Gauss-Seidel-type method similar to BCD, named *sequential greedy algorithm* (SGA), is used in multi-agent applications [28] and similar distance-based localization problems to ours [29]. As the name of the algorithm suggests, SGA is sequential in nature. In this paper, we use a coloring scheme to parallelize the computation. More importantly, we exploit multiconvex structures embedded in the formulations to establish global convergence of the sequences, while such a crucial property is *not* formally established for SGA.

To conclude this section, we note that some recent work on inverse kinematics [30, 31] also explore a point-based description and geometric perspective to characterize the problem and employ a point-based reformulation, using centralized local search [30] or SDR [31] to achieve state-of-the-art performance.

III. PROBLEM STATEMENT AND REFORMULATION

In this section, we specify the setup of distance-based relative state estimation and transform it into the generalized graph realization problem, which can be compactly formulated as a rank-constrained SDP.

A. Problem Setup

We consider a practical multi-agent system where each individual is equipped with one or multiple distance sensors and accessible to certain proprioceptive measurements (e.g., 6-axis or 9-axis inertial measurement units (IMUs)¹). We follow a high-level description of proprioception measurements that an individual agent can obtain an accurate estimate of some components of its attitude, i.e., its roll, pitch and yaw angles, which is also used in related work [1, 3, 17].

We define a *common reference system*, in which the agents' states are expressed. Since we are considering the problem of relative state estimation, the common reference system can be set arbitrarily when no proprioception is employed. Its z-axis or x-axis should be aligned with the opposite direction of gravity or geographic east, respectively, if IMUs or magnetometers (typically embedded in 9-axis IMUs) are used. The goal of this paper is to recover the states of agents in the common reference system, which best explain the given distance-only or distance-propriocception measurements.

¹References to IMUs in the following text are 4-axis IMUs by default.

B. Notions and Preliminaries

We model the measurement topology underlying the relative state estimation problem using an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in which each node $i \in \mathcal{V}$ represents an agent in the system and the edges $(i, j) \in \mathcal{E}$ indicates the measurements between agents i, j . An index set $\mathcal{B}_i := \{0, 1, \dots\}$ is introduced to distinguish between the distance sensors on the agent i . Let $\mathbf{p} := [p_1^0, \dots, p_1^{|\mathcal{B}_1|}, \dots, p_n^0, \dots, p_n^{|\mathcal{B}_n|}] \in \mathbb{R}^{d \times \sum_i |\mathcal{B}_i|}$ denote a realization of points in $\mathbb{R}^d$, where $d = 3$ by default², and n the number of agents. Each component $p_i^u := [x_i^u, y_i^u, z_i^u]^T$ for $i = 1, 2, \dots, n$ and $u \in \mathcal{B}_i$, is the coordinate of sensor (i, u) . Letting $(\cdot)$ denote measurements, $(\cdot)$ denote the (latent) ground-truth, we have the following measurement model for the distance between sensors (i, u) and (j, v) :

$$\tilde{d}_{ij}^{uv} = d_{ij}^{uv} + \epsilon = \|\underline{p}_i^u - \underline{p}_j^v\| + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2), \quad (1)$$

$$\forall (i, j) \in \mathcal{E}, \quad u, v \in \mathcal{B}_i,$$

where ϵ is the measurement noise and σ is the noise level.

In this paper, we introduce a quadratic surrogate measurement [32] based on the physical measure model (1), which is ultimately used in optimization:

$$\tilde{d}_{ij}^{uv} := \left(\tilde{d}_{ij}^{uv} \right)^2 - \sigma^2 \simeq \left(d_{ij}^{uv} \right)^2 + \epsilon_{ij}^{uv}, \quad \epsilon_{ij}^{uv} \sim \mathcal{N}(0, (\sigma_{ij}^{uv})^2),$$

$$\sigma_{ij}^{uv} = \sqrt{(2\sigma \tilde{d}_{ij}^{uv})^2 + 2\sigma^4}, \quad \forall (i, j) \in \mathcal{E}, \quad u, v \in \mathcal{B}_i. \quad (2)$$

Letting $\underline{R}_i \in \text{SO}(d)$, $\underline{t}_i \in \mathbb{R}^d$ represent the rotation and translation components of the agent i 's state in the common reference system and $\bar{v}_i^u$ the (known) coordinate of sensor u on the agent i in its body reference system, we have

$$\underline{p}_i^u = \underline{R}_i \bar{v}_i^u + \underline{t}_i. \quad (3)$$

Given a set of noisy measurements $\tilde{d}_{ij}^{uv}$, it is straightforward to derive the maximum-likelihood estimation (MLE) model for distance-based relative state estimation:

Problem 1 (The MLE model for distance-based relative state estimation).

$$\min_{\{(R_i, t_i)\}_{i \in \mathcal{V}}} \sum_{\substack{(i, j) \in \mathcal{E} \\ u, v \in \mathcal{B}_i}} \frac{1}{(\sigma_{ij}^{uv})^2} \left(\|\underline{R}_i \bar{v}_i^u + \underline{t}_i - \underline{R}_j \bar{v}_j^v - \underline{t}_j\|^2 - \tilde{d}_{ij}^{uv} \right)^2, \quad (4)$$

where *some components of R_i is measured via proprioception*.

C. From Relative State Estimation to Graph Realization

In this subsection, we transform Problem 1 to a problem that finds the realization of distance sensors best explaining the measurements and rigid body constraints, bridging Problem 1 with the optimization models derived in Section V.

The core idea of this approach lies in the use of a set of points on a rigid body to represent its rotation [14], degrading the manifold constraints imposed by rotations to intuitive *calibration and proprioception constraints*, as follows:

Problem 2 (Generalized graph realization (with redundant constraints)).

$$\min_{\mathbf{p}} \sum_{\substack{(i, j) \in \mathcal{E} \\ u, v \in \mathcal{B}_i}} \frac{1}{(\sigma_{ij}^{uv})^2} \left(\|\underline{p}_i^u - \underline{p}_j^v\|^2 - \tilde{d}_{ij}^{uv} \right)^2, \quad (5)$$

²The following results can be easily adapted to the 2-dimensional case.

s.t.

- Distance-only setup where all the distance sensors are coplanar:

$$\|p_i^u - p_i^v\|^2 = \|\bar{v}_i^u - \bar{v}_i^v\|^2, \forall i \in \mathcal{V}, |\mathcal{B}_i| \geq 2, u, v \in \mathcal{B}_i, u < v; \quad (6)$$

- distance-4-axis-proprioception setup:

$$z_i^u - z_i^v = [-\sin \tilde{\phi}_i, \cos \tilde{\phi}_i \sin \tilde{\theta}_i, \cos \tilde{\phi}_i \cos \tilde{\theta}_i] (\bar{v}_i^u - \bar{v}_i^v), \quad \forall i \in \mathcal{V}, |\mathcal{B}_i| \geq 2, u, v \in \mathcal{B}_i, u < v, \quad (7)$$

$$\begin{aligned} (\bar{v}_{i|x}(y_i^u - y_i^v) - \bar{v}_{i|y}(x_i^u - x_i^v)) &= (\bar{v}_{i|x}(y_i^v - y_i^w) - \bar{v}_{i|y}(x_i^v - x_i^w)), \\ (\bar{v}_{i|x}(x_i^u - x_i^v) + \bar{v}_{i|y}(y_i^u - y_i^v)) &= (\bar{v}_{i|x}(x_i^v - x_i^w) + \bar{v}_{i|y}(y_i^v - y_i^w)), \\ \forall i \in \mathcal{V}, |\mathcal{B}_i| \geq 3, u, v, w \in \mathcal{B}_i, u < v < w, \end{aligned} \quad (8)$$

where $[\bar{v}_{i|x}; \bar{v}_{i|y}; \bar{v}_{i|z}] := \hat{R}_{i|y} \hat{R}_{i|x} (\bar{v}_i^u - \bar{v}_i^v)$, and $\hat{R}_{i|x}, \hat{R}_{i|y}$ are the estimates from proprioception sensors of x-axis and y-axis rotation components, respectively, in the agent i 's attitude;

- distance-6-axis-proprioception setup:

$$p_i^u - p_i^v = \hat{R}_i (\bar{v}_i^u - \bar{v}_i^v), \forall i \in \mathcal{V}, |\mathcal{B}_i| \geq 2, u, v \in \mathcal{B}_i, u < v, \quad (9)$$

where $\hat{R}_i$ is the estimates of agent i the full attitude from proprioception sensors.

Proposition 1 (Equivalence between Problem 1 and Problem 2). Problem 1 and Problem 2 are equivalent in the sense that (a) for any feasible $\{(R_i, t_i)\}$ of Problem 1, $\{p_i^u\}$ that is obtained from

$$p_i^u = R_i \bar{v}_i^u + t_i \quad (10)$$

is feasible in Problem 2 for all $i \in \mathcal{V}$ and $u \in \mathcal{B}_i$ and leads to the consistent objective function value with Problem 1; and (b) for all feasible $\{p_i^u\}$ of Problem 2, there exists $\{(R_i, t_i)\}$, where R_i agrees with the proprioception of agent i , such that (10) holds for all $i \in \mathcal{V}$ and $u \in \mathcal{B}_i$, and all $\{(R_i, t_i)\}$ satisfying (10) lead to an objective function value of Problem 1 consistent with that of Problem 2.

We present a high-level insight conveyed by Proposition 1 while deferring the proof of it to Appendix A: Proposition 1 suggests that by representing agents' states using point coordinates and preserving constraints related to multiple distance sensors on the same rigid body, i.e., the *calibration constraints* (6) and *proprioception constraints* (7) and (8), we can recover the agents' states by finding a set of points that agrees with these measurements and constraints, which we call a *realization* of the measurements and constraints.

In the following result, we present a method to recover the state $\{(R_i, t_i)\}$ from $\mathbf{p}$, which is held naturally by the proof of Proposition 1.

Proposition 2 (Recovering states from sensor coordinates).

- With $|\mathcal{B}_i| \geq 2$ and 4-axis proprioception, we have

$$\sin \hat{\psi}_i = (\bar{v}_{i|x}(\hat{y}_i^u - \hat{y}_i^v) - \bar{v}_{i|y}(\hat{x}_i^u - \hat{x}_i^v)) / (\bar{v}_{i|x}^2 + \bar{v}_{i|y}^2),$$

$$\cos \hat{\psi}_i = (\bar{v}_{i|x}(\hat{x}_i^u - \hat{x}_i^v) + \bar{v}_{i|y}(\hat{y}_i^u - \hat{y}_i^v)) / (\bar{v}_{i|x}^2 + \bar{v}_{i|y}^2),$$

where $\hat{\psi}_i$ is the estimate of yaw angle, thus the full attitude $\hat{R}_i$ is recovered. Then we have $\hat{t}_i = \hat{p}_i^u - \hat{R}_i \bar{v}_i^u$ that recovers the translation component;

- with $|\mathcal{B}_i| \geq 3$, we register $\hat{p}_i^u, \hat{p}_i^v, \hat{p}_i^w$, where $u < v < w$ and $\bar{p}_i^u, \bar{p}_i^v, \bar{p}_i^w$ are non-collinear, to $\bar{v}_i^u, \bar{v}_i^v, \bar{v}_i^w$. In this

way, the transformation in the common reference system, i.e., $(\hat{R}_i, \hat{t}_i)$, can be obtained in closed-form [14, Section 5].

Remark 1 (Redundant constraints of Problem 2). When $|\mathcal{B}_i| \geq 3$, part of the constraints in (6)-(9) may be redundant, i.e., removing some of them does not make the feasible set larger. We show a result that helps us to find an equivalent subset of these constraints in Appendix B. We use this result in our evaluation (Section VII) for better efficiency.

Remark 2 (Handedness ambiguities in distance-only setup). Note that we impose all the distance sensors on a plane in the distance-only setup. This is because the distance constraint in (6) is not sufficient to discriminate handedness realizations of the true realization in a noncoplanar case. We can eliminate this ambiguity by imposing constraints such as

$$\begin{aligned} ((p_i^u - p_i^v) \times (p_i^w - p_i^x)) \cdot (p_i^y - p_i^z) &= \\ ((\bar{v}_i^u - \bar{v}_i^v) \times (\bar{v}_i^w - \bar{v}_i^x)) \cdot (\bar{v}_i^y - \bar{v}_i^z), \end{aligned} \quad (11)$$

where $(\bar{v}_i^u - \bar{v}_i^v)$, $(\bar{v}_i^w - \bar{v}_i^x)$, and $(\bar{v}_i^y - \bar{v}_i^z)$ are not coplanar. These constraints are cubic, and extending the results in this work to such constraints seems untrivial, which we view as an important future direction (see Section VIII).

D. Problem 2 as Rank-constrained SDP

In this subsection, we present a equivalent rank-constrained SDP formulation of Problem 2 for more concise derivation of results in Section V. To reveal this, variable $X := \mathbf{p}^T \mathbf{p}$ is critically introduced, which is equivalent to

$$X \succeq \mathbf{p}^T \mathbf{p}, \text{rank}(X) \leq d, \quad (12)$$

and thus can be rewritten as [33]

$$Z := \begin{bmatrix} X & \mathbf{p}^T \\ \mathbf{p} & I_d \end{bmatrix} \succeq 0, \text{rank}(Z) \leq d. \quad (13)$$

In this way, we can represent the square of distances and the constraints by linear mappings on the components of decision variable Z . For the convenience of the derivations in Section V, we divide these linear mappings into two categories. In the first category, we use $\mathcal{A}^{q|o} \cdot X$ and $\mathcal{A}^{q|c} \cdot X$ to substitute the quadratic distance $\|p_i^u - p_j^v\|^2$ in the objective function (5) and the constraint (6), respectively. In the second category, we use $\mathcal{A}^l \cdot \mathbf{p}$ to uniformly capture the remaining linear constraints (7)-(9). $\mathcal{A}^{q|o}$, $\mathcal{A}^{q|c}$ and $\mathcal{A}^l$ are the corresponding linear operators. In this way, Problem 2 can be rewritten as follows:

Problem 3 (Rank-constrained SDP formulation of Problem 2).

$$\min_Z \left\| \mathcal{A}^{q|o} \cdot X - \bar{q} \right\|^2 \quad (14)$$

s.t.

$$\begin{aligned} Z &:= \begin{bmatrix} X & \mathbf{p}^T \\ \mathbf{p} & I_d \end{bmatrix} \succeq 0, \text{rank}(Z) \leq d, \\ \mathcal{A}^{q|c} \cdot X &= \bar{\mathbf{b}}^{q|c}, \\ \mathcal{A}^l \cdot \mathbf{p} &= \bar{\mathbf{b}}^l, \end{aligned} \quad (15)$$

where $\bar{q}$ is constituted by stacking measurements $\bar{q}_{ij}^{uv}$, and $\bar{\mathbf{b}}^{q|c}$ and $\bar{\mathbf{b}}^l$ are known constant vectors from constraints (6)-(9).

This problem is non-convex due to the rank constraint and intractable to solve directly for a Karush-Kuhn-Tucker point (KKT point) [34], not to mention the global minimizer.

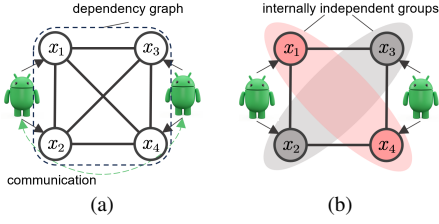


Fig. 2: **An example of dependency graphs and graph-coloring.** (a) A dependency graph of a non-parallelizable case, where each agent is responsible for two variables, and the solid lines indicate the dependency relations between variables. (b) A dependency graph of a parallelizable case and a feasible coloring scheme of the blocks.

IV. BLOCK COORDINATE DESCENT FOR MULTICONVEX OPTIMIZATION

In this section, we review the classic BCD algorithm employed in this paper, and highlight a class of problems that are convex in each block of variables although the objective function and the feasible set are non-convex in general.

A. Block Coordinate Descent

We first recap the BCD framework, which optimizes the following problem

$$\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}), \quad (17)$$

where $\mathbf{x} := [z_1, \dots, z_n]^T$ includes all variables z_i , and $\mathcal{X}$ is the feasible set and assumed closed. Letting *block division* $\{\Phi_q\}_{q=1, \dots, p} : \mathbf{x} = (x_1, \dots, x_p)$ divide $\mathbf{x}$ into p disjoint blocks, where each block of variables are denoted as x_q , the BCD algorithm of Gauss-Seidel type [13] optimizes the following *block update subproblem* over each block Φ_q cyclically

$$\min_{x_q \in \mathcal{X}_q} f_q(x_q) := f(\hat{x}_1, \dots, x_q, \dots, \hat{x}_p), \quad (18)$$

where $\hat{x}_i$ represents the variables excluded by block Φ_q and remains *fixed* at its last updated value in the BCD framework, and

$$\mathcal{X}_q := \{x_q : (\hat{x}_1, \dots, x_q, \dots, \hat{x}_p) \in \mathcal{X}\}. \quad (19)$$

In this paper, we also consider the *prox-linear update* [35, 36] for non-convex optimization, which exhibits better robustness to initialization while taking more time to converge (see Fig. 6), of which the k -th iteration is written as

$$\min_{x_q \in \mathcal{X}_q} \langle \nabla_q f(\hat{x}_q^{(k-1)}), x_q - \hat{x}_q^{(k-1)} \rangle + \frac{L_q^{(k-1)}}{2} \|x_q - \hat{x}_q^{(k-1)}\|^2, \quad (20)$$

where $\nabla_q f$ is the block-partial gradient of f , $L_q^{(k-1)} > 0$ and $\hat{x}_q^{(k-1)}$ is obtained by *extrapolation*

$$\hat{x}_q^{(k-1)} = x_q^{(k-1)} + w_q^{(k-1)}(x_q^{(k-1)} - x_q^{(k-2)}) \quad (21)$$

with $0 \leq w_q^{(k)} \leq \delta \sqrt{L_q^{(k-1)}/L_q^{(k)}}$, where $\delta < 1$ uniformly over all q and k . We consider the extrapolation (21) since it significantly accelerates the convergence of BCD in our application.

B. The Scalability and Distributed Nature of BCD

BCD is naturally suited to distributed architectures and is scalable, which we highlight in the following two ways.

Parallelization. The above updates (18) and (20) can be uniformly represented as a general *single-step Gauss-Seidel iteration*, called a *sweep* [13, Chapter 1]:

$$x_q^{(k+1)} = \pi_q(x_1^{(k+1)}, \dots, x_{q-1}^{(k+1)}, x_q^{(k)}, \dots, x_p^{(k)}), \quad (22)$$

where π_q is a mapping function for update w.r.t. block Φ_q . We introduce the notion of *dependency* as follows:

Definition 1 (Dependency between blocks) Given the arbitrary block division $\{\Phi_q\}$ and sweep π_q for $q = 1, \dots, p$, the block variable x_j is called *independent* on variable x_k if the result of corresponding sweep π_j is *invariant* under different values of x_k . Conversely, we say that x_j *depends* on x_k .

A graphic language describing the dependency relationships between blocks is called *dependency graph* [13, 19], of which we show examples in Fig. 2. A sweep (22) can be non-parallelizable if every block depends on all other blocks (see Fig. 2(a)). In contrast, in our problem formalized in Section III-B, an agent usually generates measurements only with those sensors on a subset of other agents, called the *neighbors* of the agent. We can tell from (4) or (5) that a variable p_i^u only depends on partial variables (e.g., see Fig. 2(b)) corresponding to sensors on the neighbors of agent i . In such a case, parallel computation of sweeps can happen.

A well-known method for identifying *internally independent* groups of blocks for parallelization, where blocks within the same group are independent of each other, is the *graph-coloring* method [13, 19]. Through this method, each variable block in the dependency graph is assigned a color, and the blocks with the *same* color can be merged as a new block to be updated in parallel.

Scale-invariant computation and local communication.

The computation demand for each sweep w.r.t. a block is related only to the degree of a block in the dependency graph. Similarly, constructing the sweep problem (22) only requires the responsible agent for a block to communicate locally with agents corresponding to the neighboring blocks (see Fig. 2(a)). Therefore, one can control the degree of the dependency graph to be constant as the system scales up, such that the computation and communication overhead produced by each sweep are scale-invariant. Furthermore, when parallelization is enabled with certain choice of graph-coloring algorithms [37], the number of colors can be easily controlled constant as the system's scale increases. In such a case, the sequential solution complexity per cycle of BCD, which is linearly related to the number of colors, can be bounded, and thus the computation time per cycle in BCD is approximately independent of the system's scale (see Fig. 8).

C. Convergence of BCD in Multiconvex Optimization

Although the BCD algorithm has many desired properties, it does not operate safely in general problems [38] where its convergence is often not guaranteed. Therefore, we take advantage of a specific structure, the *block multiconvex* structure [12], to wisely develop safe BCD-type algorithms.

The term of *block multiconvex* is formalized as follows.

Definition 2 (Block multiconvexity [12]). We call a set $\mathcal{X}$

block multiconvex if its projection to each block of variables, i.e., $\mathcal{X}_q$ (19), is convex. We call a function f block multiconvex if f is a convex function for each block of variables while other blocks are fixed.

When $\mathcal{X}$ and f are both block multiconvex, the subproblems in updates (18) and (20) are convex. We also need the term *Nash equilibrium* to describe the convergence property.

Definition 3 (Nash equilibrium). We call $\mathbf{x}^* = (x_1^*, \dots, x_p^*)$ a *Nash equilibrium* of (17) if

$$f(x_1^*, \dots, x_q^*, \dots, x_p^*) \leq f(x_1^*, \dots, x_q, \dots, x_p^*), \quad \forall x_q \in \mathcal{X}_q, \quad (23)$$

holds for all $q = 1, \dots, p$.

In the following Theorem 1, we present a crucial result applied in this paper, which claims the (asymptotical) progress of the sequence generated by BCD to a Nash equilibrium.

Assumption 1 [12, Assumption 2]. $f(\cdot)$ is continuous, bounded, and has a Nash equilibrium.

- The problem is strongly block multiconvex under the block division scheme, i.e., the subproblem is *strongly convex*, when the original update (18) is employed;
- $\nabla_q f$ is *Lipschitz continuous* and there exist parameters $L_q^{(k-1)}$ obey $f_q(x_q^{(k)}) - f_q(\hat{x}_q^{(k-1)}) \leq \langle \nabla_q f(\hat{x}_q^{(k-1)}), x_q - \hat{x}_q^{(k-1)} \rangle + \frac{L_q^{(k-1)}}{2} \|x_q - \hat{x}_q^{(k-1)}\|^2$, which is used in (20), when the prox-linear update (20) is employed.

Theorem 1 (Global convergence to a Nash equilibrium [12, Corollary 2.4]). Under Assumption 1, if the sequence $\{\mathbf{x}^{(k)}\}$ generated by update (18) or (20) is bounded, we have $\lim_{k \rightarrow \infty} \text{dist}(\mathbf{x}^{(k)}, \mathfrak{N}) = 0$, where $\mathfrak{N}$ is the set of Nash equilibrium.

We apply this result to establish the convergence guarantee to a *critical point* for the proposed methods in Section VI-C.

V. PRACTICAL OPTIMIZATION MODELS FOR GENERALIZED GRAPH REALIZATION

In this section, we derive a convex model and a local search model based on necessary low-level designs, which are highlighted in **bold** at the beginning of corresponding paragraphs. We begin by motivating the utilization of ad hoc optimization models instead of Problem 3 and vanilla SDR.

A. Dilemma of Estimating with Problem 3

A classic approach in related literature [11, 23, 34] to general rank-constrained SDP is to drop the rank constraint in (13) and obtain a (convex) SDP surrogate, solve the SDP, and rounding a solution of the original problem. However, the constraint (13) on Z leads to a polynomial increase in computational complexity with the systems' scale [10, 11], which cannot be operated scalably with general-purpose solvers [39].

Moreover, minimizing the objective function in Problem 3 does not impose any direct optimization on the first-order variable component $\mathbf{p}$ of Z , and thus $\mathbf{p}$ eventually converges to a *trivial* feasible solution. Thus, it is impossible to rounding an informed estimate directly from the (1,2) (first row, second column) or (2,1) (second row, first column) block of Z .

An alternative rounding method is to perform approximate rank- d decomposition, e.g., through eigenvalue decomposition, on X , the variable component of Z encoded into all the information regarding minimizing the objective function. This approach, although preserving the optimization behavior, cannot maintain the constraints imposed by constraints (15) and (16) on the (1,2) or (2,1) block in Z .

The solution we present in the following subsection (i) using a nonlinear programming (NLP) model formulation where rounding is straightforward and (ii) is manually arranging a set of anchors, i.e., sensors whose locations are known as a prior. We observe that, fortunately, a *small number* of anchors is sufficient to extract informed estimates from the (1,2) or (2,1) block of Z for the downstream refinement.

B. An Edge-based Semidefinite Relaxation Model

The decision variable of the vanilla SDR is densely constrained by the semidefiniteness of the whole cone matrix Z , thus not decomposable. Therefore, we are motivated to further relax the full semidefiniteness constraint to decompose Z according to the measurement topology $\mathcal{E}$.

Edge-based relaxation of SDP. We follow the idea of edge-based decomposition appearing in [16] to *relax* $Z \succeq 0$ such that the cone matrix Z is decomposed into its principal submatrices, as follows:

$$\begin{bmatrix} X_{ii}^{uu} & X_{ij}^{uv} & (p_i^u)^T \\ X_{ji}^{vu} & X_{jj}^{vv} & (p_j^v)^T \\ p_i^u & p_j^v & I_d \end{bmatrix} \succeq 0, \quad (24)$$

$$\forall (i, j) \in \mathcal{E} \vee (i = j \wedge (u, v) \in \mathcal{E}_i),$$

where X_{ij}^{uv} is an abbreviation for the auxiliary variable corresponding to $(p_i^u)^T p_j^v$ in (12), and $\mathcal{E}_i$ is defined as the set of pairs of sensors on agent i that are taken into account in the calibration constraint (6).

The constraints in (24) only appear between pairs of sensors where measurement edges exist in $\mathcal{G}$ (recall Section III-B) or multiple sensors on the same agent. Hence, the resulting problem is called *edge-based SDP* (ESDP). In this way, the high-dimensional constraint $Z \succeq 0$ (13) is decomposed into several low-dimensional constraints. We observe in evaluation (Section VII) that although ESDP is a further relaxation of SDP, solving it results in a similar accuracy with that of SDP.

Manual anchors. As mentioned in Section V-A, directly solving the rank-relaxation problem of Problem 3 leads to a dilemma in extracting estimates. Fortunately, Theorem 2 in [23], which is originally proposed for the distance-only setup, can be directly extended to the distance-proprioception measurements, which tells: *in the noiseless case, if there are anchors configured to uniquely localize all distance sensors, $\mathbf{p}$ can be estimated exactly by solving the rank relaxation of Problem 3 (with modification in formulation to include anchors information) and extracted from the (1,2) block of the solution Z .* i.e., the relaxation is tight. Conversely, if sensors cannot be uniquely localized, the relaxation is always loose.

The above result implies that the information provided by anchors is crucial. Inspired by this, we make the following

key observation: even with only a *small number* of imperfectly localized anchors, it is possible to recover informed estimates of $\mathbf{p}$ from the (1,2) block of Z via ESDP *if these anchor sensors generate measurements with a sufficient proportion of other sensors*. The following remark clarifies that anchoring some agents in practice is often possible.

Remark 2 (Feasibility of manual anchors). In practice, we can anchor a single agent with multiple distance sensors, considering that in the context of *relative* state estimation, one agent can be regarded anchored without any loss of generality, to provide anchor information using calibrated coordinates of the sensors on it (as done in Section VII-C). We can also use existing relative state estimation techniques (e.g., [1]) to provide (imperfect) relative positions of a small number of agents that serve as anchors for large-scale systems.

We formalize this anchored ESDP as follows:

Problem 4 (Anchored edged-based relaxation for Problem 3).

$$\begin{aligned} \min_Z \quad & \|\mathcal{A}^{q|o} \cdot X - \tilde{\mathbf{q}}\|^2 + \|\mathcal{A}^a \cdot Z - \tilde{\mathbf{q}}^a\|^2, \\ \text{s.t.} \quad & (15), (16) \text{ and } (24), \end{aligned} \quad (25)$$

where $\mathcal{A}^a \cdot Z$ is introduced to substitute the term $\|p_i^u - a_m\|^2$, the quadratic distance from any anchor a_m , and $\tilde{\mathbf{q}}^a$ represents the stacked anchor measurements, similar to Section III-D.

How many anchors and anchor measurements are sufficient? We define ‘sufficient’ in the sense that we can extract the exact or (at least) non-trivial estimates of the sensors from the (1,2) block of Z by solving Problem 4. Unfortunately, we cannot yet provide the exact conditions that make the above hold. Nevertheless, the following result implies a necessary number of anchors.

Proposition 3. Suppose the measurements are noise-free. If a sensor (i, u) is not uniquely localizable, then there must exist X_{ii}^{uu} and p_i^u in the solution set such that $X_{ii}^{uu} > (p_i^u)^T p_i^u$.

The above result, proved in Appendix C, suggests that if sensors’ positions cannot be uniquely determined, then the relaxation is always loose, and an informed estimate can not be safely obtained by solving Problem 4. For example, as is well known, in a distance-only setup, if the number of anchor sensors is less than $d + 1$, then we cannot get a uniquely determined sensor position at most times due to reflection invariance. Although including proprioception information allows the number of anchors required to be further reduced, we still consider it safe to use at least $d + 1$ anchor sensors.

We experiment to determine an empirical amount of anchor measurements required in the case of sparingly using only $d + 1$ anchor sensors, of which the results are shown in Fig. 3. Although it is difficult to guarantee absolute tightness with this very limited number of anchors (see Fig. 3(a)), we can obtain near-perfect estimates (see Fig. 3(b)) for most agents with a considerably cheap configuration of anchors and anchor measurements. These results validate that a small number of anchor agents can be used to support large-scale ESDP.

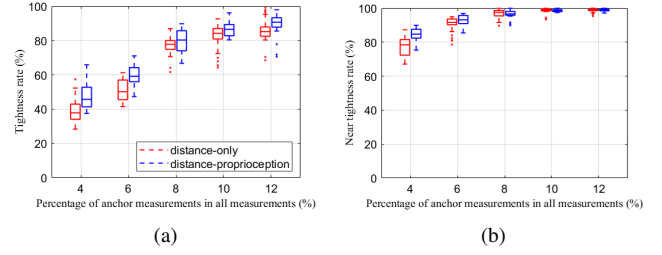


Fig. 3: **Tightness under varying anchor measurements rate.** The experiments are conducted in a constant topology 3D problem with 125 agents distributed in $[0, 12\text{m}]^3$. The agents are either equipped with two distance sensors (the distance-only setup) or also with an additional IMU (the distance-proprioception setup). Two neighboring agents (i.e., four sensors) are used as anchors. (a) The tightness rate which is defined as the proportion of sensors whose localization error is under 10^{-3}m . (b) The near tightness rate which is defined as the proportion of sensors whose localization error is under $5 \times 10^{-2}\text{m}$.

C. A Low-Rank Factors-based Local Search Model

In this subsection, we derive a non-convex model with bi-convexity for local search to refine the solution by convex relaxation or perform online estimation. We show in Section VI that this specialized model leads to favorable computational and convergence properties.

Bi-convex factorization. We draw inspiration from a classic application, *collaborative prediction* [40, 41], to introduce *bi-convexity* (the 2-blocks instance of block multiconvexity) to the *BM factorization* [42] for semidefinite constraint Z . By introducing the bi-convexity, we can establish a multiconvex optimization model (as suggested by Lemma 1).

Collaborative prediction [40, 41] fits an $m \times n$ target matrix X with a pair of rank- r factors $U \in \mathbb{R}^{m \times r}$ and $V \in \mathbb{R}^{n \times r}$, such that $X = UV^T$. The key idea to solve it is that if one of the factors, say U , is fixed, then fitting the target reduces as

$$\min_V \|\mathcal{A} \cdot \hat{U}V^T - \bar{b}\|^2, \quad (26)$$

which is *quadratic* on the block variable V and thus can be solved in a closed form. A 2-blocks BCD iterating between U and V can solve the problem.

On the other hand, the proposal of Burer and Monteiro [42], called BM factorization, factorizes a semidefinite positive matrix, say Z in (13), with some low-rank matrices $Y \in \mathbb{R}^{r \times \sum_i |\mathcal{B}_i| + d}$:

$$Z = Y^T Y = \begin{bmatrix} U^T \\ Q^T \end{bmatrix} \begin{bmatrix} U & Q \end{bmatrix}, \quad Q^T Q = I_d, \quad (27)$$

where $U \in \mathbb{R}^{r \times \sum_i |\mathcal{B}_i|}$ and $Q \in \mathbb{R}^{r \times d}$ with r satisfying $d \leq r \leq \sum_i |\mathcal{B}_i| + d$, being the rank upper bound one imposes on Z . This drastically reduces the search space and preserves the semidefiniteness constraint.

Following the above factorizations, we can introduce bi-convexity into the matrix Z in (13) by factorizing it as

$$Z = \begin{bmatrix} U^T \\ Q^T \end{bmatrix} \begin{bmatrix} V & Q \end{bmatrix}, \quad U - V = O, \quad Q^T Q = I_d, \quad (28)$$

while preserving the semidefiniteness of Z .

Constant projection operator. The constitutive relation from U (V) and Q to Z in (13) reveals $Q^T U \equiv \mathbf{p}$ ($Q^T V \equiv \mathbf{p}$). Consequently, we can view U as an r -dimension surrogate of

p , while Q acts as a (row) projection operator that maps U onto the realization space $\mathbb{R}^d$ of p .

Based on the above explanation regarding the physical interpretation of Q , we have two options: (i) simultaneously optimizing the projection operator Q and the r -dimension realization U , and (ii) fixing Q and optimizing U only, as fixing Q does *not* affect on the optimal values of the model and the corresponding optimal estimates. Since fixing Q is expected to result in a more efficient optimization procedure without the need to preserve the orthogonality constraint, we choose a specific Q and fix it throughout the optimization. Note that with Q fixed, different choices of Q do not intrinsically affect the model's behavior, so we can trivially determine Q as $[I_d \ O_{d \times (r-d)}]^T$, for example.

Surrogate U - V -coupled constraints. We replace the constraint (15) and $U - V = O$ with surrogate functions on the objective function by employing methods such as penalty methods and the augmented Lagrangian method [43, Chapter 17]. These constraints are featured by the fact that U and V are *coupled* in them. The account of necessity for surrogating U - V coupled constraints is postponed to Section VI-C. In our implementation, we employ the vanilla quadratic penalty method since it works considerably well in practice.

In comparison, we can rewrite (16) as $\mathcal{A}^l \cdot (Q^T U) = \bar{b}^l$ and $\mathcal{A}^l \cdot (Q^T V) = \bar{b}^l$ according to $p \equiv Q^T U$ and $p \equiv Q^T V$, respectively, i.e., U and V are *separate* with each other in constraint (16), and we strictly preserve it as in Problem 3.

We obtain the following constrained non-convex model:

Problem 5 (Low-rank factors-based model for Problem 3).

$$\min_{U, V} \|\mathcal{A}^{q|o} \cdot (U^T V) - \tilde{q}\|^2 + \quad (29)$$

$$\lambda \|\mathcal{A}^{q|c} \cdot (U^T V) - \bar{b}^{q|c}\|^2 + \gamma \|U - V\|_F^2$$

s.t.

$$\mathcal{A}^l \cdot (Q^T U) = \bar{b}^l, \quad (30)$$

$$\mathcal{A}^l \cdot (Q^T V) = \bar{b}^l, \quad (31)$$

where λ and γ are the penalty coefficients.

Algorithm 1 *The Meta Algorithm of Block Coordinate Descent for Problem 4 and Problem 5*

Input: Block division $\{\Phi_q\}_{q=1, \dots, p}$, an initialization $p^{(0)}$

Output: Estimates of agents' states $(\hat{R}, \hat{t})$

```

1:  $k \leftarrow 0$ 
2: while 1 do
3:    $k \leftarrow k + 1$ 
4:   for  $q = 1, \dots, p$  do
5:     Solve (18) or (20) w.r.t.  $\Phi_q$ 
6:   end for
7:   if  $\|p^{(k)} - p^{(k-1)}\|_F / \|p^{(k-1)}\|_F \leq \epsilon$  then
8:     break
9:   end if
10: end while
11: Recover  $(\hat{R}, \hat{t})$  according to Proposition 2
```

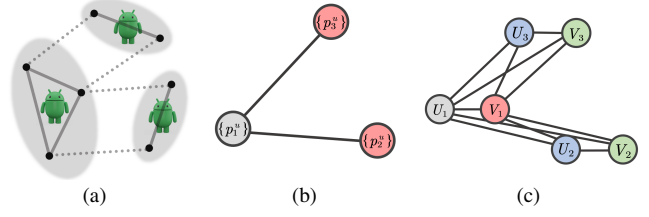


Fig. 4: **Examples of the specialized block division for Problem 4 and Problem 5, respectively.** (a) An example of a 3-agents system, where the gray ellipses, dashed lines, and solid lines denote the rigid bodies corresponding to agents, distance measures, and rigid body constraints, respectively. (b) The block division for ESDP-BCD in the example and a feasible coloring scheme for it. (c) The block division and a corresponding feasible coloring scheme for BM-BCD in the example, where U_i (V_i) represents variables corresponding to the sensors on agent i in U (V).

VI. SOLUTIONS VIA BLOCK COORDINATE DESCENT

In this section, we propose solution techniques for Problems 4 and 5, which results in two collaborative algorithms, ESDP-BCD and BM-BCD, with Algorithm 1 as a unified backbone. In Algorithm 1, we need to specify block division for BCD. We establish rules that should be followed by the block division used in the proposed approach, which are necessary for constructing the multiconvex structure that leads to desired computational (Proposition 4) and convergence (Theorems 2 and 3) results.

Rule 1. No equality constraints exist between variables in different blocks.

Rule 2. In Problem 5, variables in U and V cannot be in the same block at the same time.

Lemma 1. Under Rule 1 and Rule 2, the feasible sets and objective functions of Problem 4 and Problem 5 are block multiconvex and Assumption 1 holds.

Lemma 1 responds to the requirements of Theorem 1. We apply a simple cyclical block division in the evaluation (Section VII), as illustrated by Fig. 4: for Problem 4, all variables corresponding to distance sensors on the same agent constitute a block; for Problem 5, variables in U and V corresponding to distance sensors on the same agent, denoted as U_i and V_i in Fig. 4(c), constitute two separate blocks, respectively, which satisfies Rule 2. Since in Problem 2, the equality constraints only come from pairs of sensors on the same agent, Rule 1 holds for the above block division schemes. Then, we can apply graph-coloring to enable parallel execution, as illustrated in Fig. 4(b)) and Fig. 4(c)).

A. ESDP-BCD for Convex Block Updating on Problem 4

We propose *Edge-based SDP with Block Coordinate Descent* (ESDP-BCD) to globally solve Problem 4. We begin by presenting the block update subproblem of Problem 4 that allows exact and efficient block update and global convergence to a global minimizer (Theorem 2).

Subproblems of ESDP. We present the formulation of the subproblem using update (18) corresponding to agent i in the following, and postpone its derivation to Appendix C.

$$\min_{\{p_i^u\}, \{X_{ii}^{uu}\}} \|\mathcal{A}^{q|o} \cdot X - \tilde{q}\|^2 + \|\mathcal{A}^a \cdot Z - \tilde{q}_a\|^2, \quad \text{s.t.}$$

$$X_{ii}^{uu} \geq \|p_i^u\|^2, \forall u \in \mathcal{B}_i, \quad (32)$$

$$X_{ii}^{uu} \geq \|p_i^u\|^2 + (X_{ij}^{uv} - (p_i^u)^T \hat{p}_j^v)^2 / (\hat{X}_{jj}^{vv} - \|\hat{p}_j^v\|^2), \\ \forall j \in \mathcal{N}_i, u \in \mathcal{B}_i, v \in \mathcal{B}_j, \quad (33)$$

$$\begin{bmatrix} X_{ii}^{uu} & X_{ii}^{uv} & (p_i^u)^T \\ X_{ii}^{vu} & X_{ii}^{vv} & (p_i^v)^T \\ p_i^u & p_i^v & I_d \end{bmatrix} \succeq 0, \forall (u, v) \in \mathcal{E}_i, \quad (34)$$

constraints corresponding to agent i in (15), (16),

where $\hat{p}_j^v$ and $\hat{X}_{jj}^{vv}$ are the fixed variables corresponding to the sensor (j, v) .

The above problem is a (convex) quadratic conic programming that can be solved efficiently using off-the-shelf solvers [39]. The complexity of solving the problem only depends on the agent i 's number of neighbors. Similar results can be established on update (20), except we have no reason to employ it on such a convex problem.

The following result, proved in Section VI-C, shows that lines 2-10 of Algorithm 1 converge a global minimizer of Problem 4.

Assumption 2. Every agent is connected in $\mathcal{G}$, directly or indirectly, to at least one agent with a sensor p_i^u such that the sequence $\{(p_i^u)^{(k)}\}$ is bounded.

Assumption 3. The constraints (33) are replaced with surrogate functions (e.g., the barrier function) in the objective function when solving the subproblems.

Theorem 2. Let $Z^{(k)}$ be the k -th iteration generated by lines 2-10 in Algorithm 1 for solving Problem 4, and $\mathfrak{M}$ the set of global minimizers of Problem 4. Under Rules 1 to 2 and Assumptions 2 to 3, we have $\lim_{k \rightarrow \infty} \text{dist}(Z^{(k)}, \mathfrak{M}) = 0$.

Assumption 3 is proposed for a similar reason to surrogating U - V -coupled constraints in Section V-C, which is to construct the *Cartesian product* structure under Rules 1 and 2, as explained in detail in the next subsection. Our implementation uses the interior point method that employs the barrier method [39] to fulfill this assumption.

Refinement and extracting estimates. As claimed in Section V-B, we can directly extract an informed realization from the (1,2) block of Z . Despite not relying on initial guesses, ESDP provides only coarse estimates (especially for the rotation components) under noisy measurements. To further refine the estimates, we employ BM-BCD introduced in the following subsection.

B. BM-BCD for Fast Local Searching on Problem 5

We propose *Burer-Monteiro factorization with Block Coordinate Descent* (BM-BCD) to efficiently solve Problem 5. BM-BCD enables closed-form block update (Proposition 4) and global convergence (Theorem 3).

Linear subproblems. The formulation of Problem 5 allows extremely fast iterations due to the linearity of its subproblems under Rule 2, which is formalized as follows:

Proposition 4 (Linear subproblems and closed-form updates). Under Rule 2, the updates (18) and (20) for Problem 5 are

both the (feasible) *linearly constrained linear least square* problems, which allow closed-form solutions.

Under the applied block division described previously, the closed-form update only requires solving a low-dimensional linear system, which can be performed extremely fast.

The following result, proved in Section VI-C, shows that Algorithm 1 converge to KKT points of Problem 5.

Theorem 3. Fix λ and γ arbitrarily. Let $(U^{(k)}, V^{(k)})$ be the k -th iteration generated by lines 2-10 of Algorithm 1 for solving Problem 5 and $\mathfrak{C}$ the set of KKT points of Problem 5. Under Rules 1 to 2 and Assumption 2, we have $\lim_{k \rightarrow \infty} \text{dist}((U^{(k)}, V^{(k)}), \mathfrak{C}) = 0$.

Assumption 2 appears to be unguaranteed at first glance in Problem 5, but, in practice, we can artificially anchor one distance sensor without any loss of generality, such that the sequence of variables corresponding to this sensor is constant, and Assumption 2 naturally holds as long as $\mathcal{G}$ is connected.

Refinement and extracting estimates. Solving Problem 5 with lines 2-10 of Algorithm 1 results in a high-dimensional realization of p , i.e., U (or V), in $\mathbb{R}^r$. We propose a refinement step by setting $r = d$ in Problem 5, as line 11 in Algorithm 1, which iterates from the previous solution $Q^T U$ (or $Q^T V$). It is justified by the fact that when r is chosen to be exactly the dimension d , $Q^T U$ (or $Q^T V$) itself can be considered a realization in $\mathbb{R}^d$ and thus an (approximate) stationary point of Problem 2 according to Theorem 3. Then, we can recover the state estimates according to Proposition 2.

C. Convergence Guarantees

In this subsection, we show how applying Theorem 1 can lead to the claimed convergence properties Theorems 2 and 3.

Boundness of the sequences. Theorem 1 assumes the boundness of sequences. We start by declaring that this holds under Assumption 2.

Lemma 2 (Boundness of level sets). Under Assumption 2, the level sets $\mathcal{H}(\bar{x}) = \{x \in \mathcal{X} : f(x) \leq f(\bar{x}), \bar{x} \in \mathcal{X}\}$, for a bounded $\bar{x}$, in Problem 4 and Problem 5 are bounded.

The proof of Lemma 2 is provided in Appendix D. Remark 2.2 in [12] indicates that the boundness of the level set sufficiently guarantees the boundness of sequences generated by BCD. So, the boundness assumption is satisfied.

Equivalence between Nash equilibriums and KKT points. Theorem 1 points out the global convergence to a Nash point. We improve this result to critical points for our proposed methods. In particular, the following Lemma reveals the equivalence between Nash equilibriums and KKT points under Assumption 2, and the proof is given in Appendix D.

Lemma 3 (Equivalence between Nash equilibriums and KKT points over a Cartesian product set). Consider an optimization problem (17) given a block division $\{\Phi_q\}_{q=1,\dots,p}$ such that

$$\mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_p. \quad (35)$$

Let $f(\cdot)$ be continuously differentiable. The Nash equilibrium of the problem (17) also satisfies the KKT conditions if the constraints are continuously differentiable and under some qualifications (e.g., the LICQ [43, Chapter 12.2]).

We note that $\mathcal{X}$ is not a Cartesian product (35) with constraint (33) or the U - V -coupled constraints under Rule 2 that is necessary for Proposition 4. Fortunately, as long as these constraints are replaced with some surrogate function $s(\mathbf{x})$ on the objective function, e.g., as fulfilled by Assumption 3 for ESDP-BCD and the penalty terms in (29) for BM-BCD, we can still obtain a Cartesian product structure under Rule 1.

Application of Theorem 1 under specialized block division. According to Lemma 1, under Rules 1 and 2, the corresponding objective functions of the subproblems of Problem 4 and Problem 5 are strongly convex. That is, Assumption 1 that is required by Theorem 1 is satisfied. Combined with the boundness of sequences (Lemma 2) and equivalence between Nash equilibriums and KKT points (Lemma 3), a direct reference to Theorem 1 proves that BCD converges to critical points of our proposed optimization models. For convex problems, such as Problem 4, the critical point is also one of the global minimizers. Therefore, Theorem 2 and Theorem 3 hold.

VII. EXPERIMENTS

In this section, we assess the performance of our proposed methods compared to alternative methods under various system setups. We perform all experiments in MATLAB running on a Linux laptop with the Intel i5-11400H.

Metrics. We use the $(\Delta + 1)$ -coloring algorithm [37], where Δ is the maximum degree of the dependency graph, for parallelization by default, so that the number of colors can be controlled constant as systems scaling up with a constant Δ . We evaluate the total time to solve subproblems in parallel, abbreviated as PT (Parallel Time). We also show the total time to solve each subproblem serially, abbreviated as ST (Serial Time). To measure the accuracy of the relative state estimation, we use the average of the root-mean-squared-error (RMSE) for relative translations in body reference systems for all evaluated methods.

Choice of parameters in the algorithms. The parameter r in BM-BCD affects the results to a great extent. A numerical experiment demonstrates the performance of the algorithm under various r in Fig. 11, Appendix E. From the results, we can see that larger r results in better robustness to the initializations at the cost of taking more iterations. In the following experiments we take $r = d + 1$ and $r = d$ (i.e., refinement-only) for comparison. This choice is adequate to highlight the algorithm's features and ensure its efficiency.

The termination condition ϵ in Algorithm 1 for BM-BCD is uniformly taken as 5×10^{-4} . For ESDP-BCD and SOCP-BCD, ϵ is taken as 1×10^{-3} in Section VII-B and 5×10^{-3} in Section VII-A and Section VII-C. We sometimes choose a loose termination condition for the latter since they do not provide highly accurate estimates and, therefore, do not need to be 'fully convergent' before refinement.

A. Distance-Proprioception-based Systems

In this subsection, we intend to validate the efficacy and efficiency of the methods on problems filled with spurious minima. We consider 3D systems that consist of agents with two distance sensors and an IMU on each.

Simulation problems. We test on three simulation problems where agents respectively 'form' a **cube** (125 agents; Fig. 5(b)), a **pyramid** (120 agents; Fig. 5(d)) and are randomly distributed in $[0, 12\text{m}]^3$ (125 agents), with sparse-connected measurement topologies $\mathcal{G}$. The (maximum degree, minimum degree) of $\mathcal{G}$ in the three problems are (26, 9), (24, 6) and (18, 18), respectively. All distance measurements of an agent occur with the specific number of nearest neighboring agents. The ground-truth distance between adjacent agents in the first two problems are 3m and 4m, respectively. The default distance noise level is $\sigma = 0.1\text{m}$, and the errors of each attitude angle measurement are uniformly distributed in $[-1.5\text{ deg}, 1.5\text{ deg}]$, aligned with those observed in hardware experiments [3]. The positions of distance sensors are uniformly set as $\bar{\mathbf{v}}_i^0 = [0; 0.35\text{m}; 0]$ and $\bar{\mathbf{v}}_i^1 = [0; -0.35\text{m}; 0]$. We randomly generate the initial guess of each agent's location on a ball with radius ρ , and the ground-truth as its center. For convex-relaxation methods, we set 8 and 6 *imperfect* anchor agents in the **cube** and **pyramid** problems, respectively, each of which measures 15 other agents. See Fig. 5(c) and Fig. 5(e).

1) *Comparison of local search methods:* **Benchmark.** When using local search methods standalone, we compare the proposed method BM-BCD with (i) *Riemannian Gradient Descent* (RGD), a first-order algorithm for solving Problem 1 on smooth Riemannian manifolds based on a centralized backtracking line-search, previously used to solve a similar problem in [4], (ii) *Riemannian Trust-Region* (RTR), a numerical robust centralized second-order algorithm for solving Problem 1 on smooth Riemannian manifolds, (iii) *Quasi-Newton* (QN), a standard centralized quadratic first-order algorithm for solving Problem 5 with a fixed $r = d$, (iv) *Block Coordinate Gradient Descent* (BCGD), a BCD-type algorithm which replaces all the block updating steps in BM-BCD with backtracking line-search-based gradient descent with $r = d + 1$. The Riemannian optimization methods are implemented in the Manopt Toolbox [44], where Hessian is approximated using finite differences. QN is implemented in the MATLAB Optimization Toolbox.

We evaluate three variants of BM-BCD. BM-BCD(d) and BM-BCD($d+1$) use update (18), where the parameter r in Problem 5 is set as d and $d+1$, respectively, while BM-BCD($d+1$)[prox-linear] uses update (20).

Results. Fig. 6 shows the performance of each method in the distance-proprioception setup. Fig. 5(b) and Fig. 5(d) visualize the results of BM-BCD($d+1$) with $\rho=6\text{m}$ and $\rho=8\text{m}$, respectively. From the results, we can see that among the local search methods, the BM-BCD family and QN manifest desired precision under mild initialization conditions, whereas the methods based on Riemannian optimization prone to fall into worse local minima than BM-BCD and QN. This may be an inherent disadvantage of Riemannian optimization algorithms, as its non-convex search space may lead to ill-conditioned local geometry and introduce additional shallow local minima. Compared to QN, the computational efficiency provided by the BM-BCD method is irreplaceable, with magnitudes of advantage. The method that also iterates fast is BCGD. However,

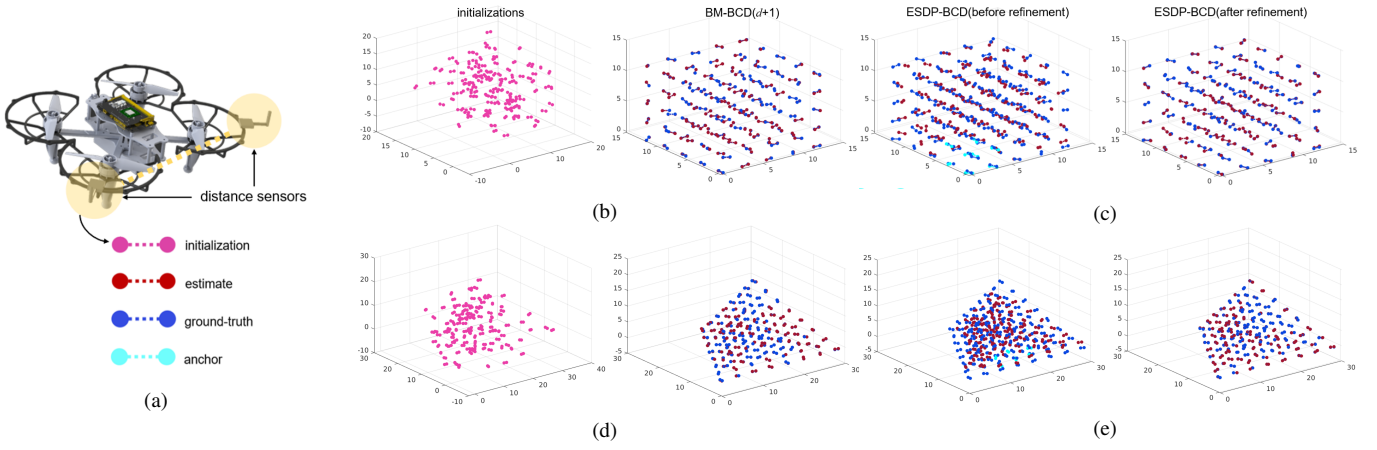


Fig. 5: **Visualization of the problems and results of the proposed methods.** (a) Illustration of a physical agent corresponding to the sensor configuration in Section VII-A: an agent can be graphically abstracted as two connected dots together with a dashed line connecting them, where the solid dots represent the two distance sensors on the agent. (b) Initialization and results of BM-BCD($d+1$) in **cube**. (c) Results before (left) and after (right) refinement of ESDP-BCD in **cube**. (d) Initialization and results of BM-BCD($d+1$) in **pyramid**. (e) Results before (left) and after (right) refinement of ESDP-BCD in **pyramid**.

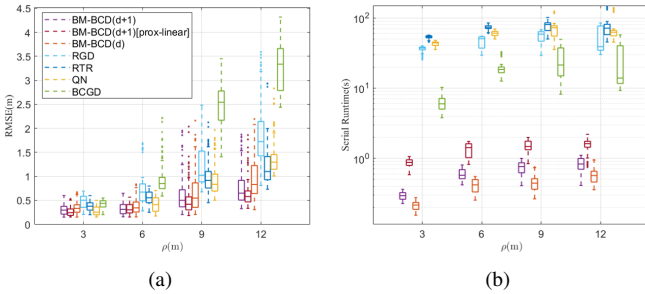


Fig. 6: **Estimation precision and serial runtime of the NLP methods.** The evaluation is performed in the 125 randomly-distributed-agents scenario. Statistics are computed under 100 times of data generation.

due to the lack of careful design, BCGD *fails* to converge to desired termination conditions sometimes.

One of the promising results shown in Fig. 6(a) is that the BM-BCD family exhibits robustness to initial conditions, which is unmatched by all other methods. Among the BM-BCD family, BM-BCD($d+1$)[prox-linear] exhibits the most stable performance, probably because the local linearization in the update (20) helps the method to escape certain local minimas [12, 36], at the cost of taking more iterations.

2) *Comparison of convex relaxation methods: Benchmark.* The convex relaxation-based ESDP-BCD method is compared with (i) *Semidefinite Programming* (SDP), a direct rank-relaxed version of Problem 3, (ii) *Second-Order Cone Programming with Block Coordinate Descent* (SOCP-BCD), a second-order cone relaxation [45, Eq. (6)] of Problem 3 with anchors distributed at the boundary of the system³, which is *structural decomposable* and thus can be solved via a BCD-style algorithm as ESDP-BCD. For a fair comparison, SDP, RE and ESDP-BCD share the same anchor information. SDP and SOCP-BCD also use BM-BCD as refinement in Table I, while the shown errors in the experiment of Fig. 7 are all

³SOCP-BCD demands strictly on the distribution of anchors in our experiments, so we use an (impractical) approach that lets the anchors be scattered across the boundary to allow SOCP to produce valid estimates.

calculated *before refinements* and in the common reference system instead of body reference systems. We solve convex programming in the above methods using the interior point method implemented in the MOSEK solver [46].

Results. Convex relaxation-based methods result in relatively consistent solutions on a given setup due to its robustness to initial guesses. As can be seen in Table I, SDP results in the minimum RMSE before refinement compared to other methods and ESDP-BCD is the second, and the results of the two methods are *close*. After the refinement stage, the two methods behave *similarly* in terms of precision, although ESDP is a further relaxation of SDP. This is due to the robustness to initialization provided by BM-BCD. SOCP-BCD results in a highly suboptimal solution due to too little anchor information. In addition, due to the ease with which ESDP-BCD and SOCP-BCD can take advantage of initial guesses and their decomposable structures, these methods exhibit higher computational efficiency than SDP in such large-size problems. We postpone a fairer comparison of scalability to the distance-only setup in the following subsection where ESDP-BCD is randomly initialized.

We also show the behavior of the convex relaxation methods under different anchor and anchor measurement configurations in Fig. 7. It can be seen that increasing the number of anchors and anchor measurements can both help the convex relaxation-based methods perform better in accuracy, where SOCP-BCD is most sensitive to changes of such conditions.

B. Distance-only Systems

In this subsection, we consider the distance-only setup in 2D and 3D problems, where each agent is equipped with three distance sensors for 3D problems and two for 2D problems.

Simulation problems. We test on three simulation problems where agents respectively 'form' a 3D **cube** and a 2D **hexagon** (217 agents; Fig. 12 in Appendix E). The (maximum degree, minimum degree) of $\mathcal{G}$ in the **hexagon** problem is (12, 5), respectively. The positions of sensors on each agent are uniformly set as $\bar{v}_i^0 = [0; 0.35\text{m}]$ and $\bar{v}_i^1 = [0; -0.35\text{m}]$ for

Problem	Method	PT(s)	ST(s)	RMSE(m)
cube	ESDP-BCD	7.12	33.7	0.23 (1.28)
	SDP	—	>100	0.23 (0.86)
	SOC-BCD	8.84	38.9	0.36 (2.08)
pyramid	ESDP-BCD	6.23	35.8	0.40 (1.54)
	SDP	—	93.9	0.38 (1.32)
	SOC-BCD	7.64	44.2	0.50 (2.53)

TABLE I: Numerical results of the proposed and alternative methods. The methods labeled using **magenta** are proposed in this paper. The accuracy metric RMSE is presented as 'RMSE in body reference systems after refinement (RMSE in the common reference system before refinement)'. Statistics are computed as the average results under 100 times of data generation for each problem and each method except SDP. SDP is tested 30 times for data generation on each problem.

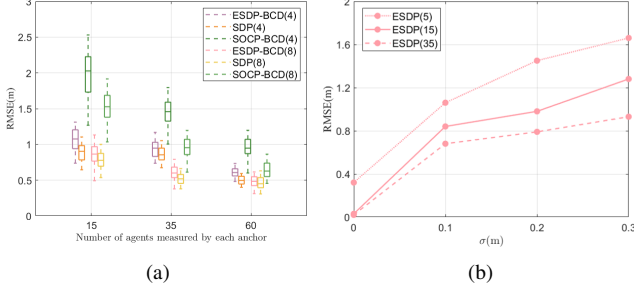


Fig. 7: **Convex relaxation-based methods under various numbers of anchor measurements.** The evaluation is performed in the 125 randomly-distributed-agents scenario. Statistics are computed under 100 times of data generation for ESDP-BCD and SOC-BCD and 30 times for SDP. (a) The benchmark results under various numbers of anchor measurements, where the number marked in the legend represents the number of anchors. (b) The results of different numbers of measurements per anchor with 6 anchors under various noise levels, where the number marked in legend represents the number of anchor measurements.

$d = 2$. We set 4 anchor agents in the **hexagon** problem, each of which measures other 15 agents. The initial guess of each agent's state is randomly distributed in a range for a relatively fair comparison of scalability. The rest of the settings are consistent with the previous subsection.

Benchmark. In the distance-only setup, we evaluate the convex relaxation-then-refinement pipeline and introduce *Riemannian Elevator* (RE) [10], a state-of-the-art method for centralized distance-only localization in the term of solution optimality, as an additional benchmark. RE uses RTR for refinement in our experiment, consistent with [10]. We also use RTR to ablate the refinement algorithm BM-BCD in ESDP-BCD and SDP, as remarked in Table II. The remaining setup of benchmark methods is consistent with that of VII-A2.

Results. It can be seen from Table II that all ESDP-BCD, RE, and SDP can provide informed initializations for the refinement stage. The relaxation used in RE results in the best accuracy among the methods, consistent with the previous study's results [10]. This is because the algorithmic design of RE creates a restriction on the rank of the solution. In contrast, SDP and ESDP-BCD use the standard interior-point method solver, leading to the max-rank solution in their solution sets [39]. However, with the help of BM-BCD, the final estimates of ESDP-BCD and SDP can still be better or quite precise than RE, which cannot be achieved with RTR.

Of particular importance, as shown in Fig. 8, ESDP-BCD

Problem	Method	RMSE(m)
cube	ESDP-BCD	0.48 $\pm$ 0.11 (0.57 $\pm$ 0.13)
	ESDP-BCD (refined with RTR)	0.52 $\pm$ 0.13 (0.57 $\pm$ 0.13)
	RE	0.45 $\pm$ 0.08 (0.43 $\pm$ 0.11)
	SDP	0.45 $\pm$ 0.09 (0.53 $\pm$ 0.15)
	SDP (refined with RTR)	0.49 $\pm$ 0.10 (0.53 $\pm$ 0.15)
	SOC-BCD	0.79 $\pm$ 0.31 (1.24 $\pm$ 0.29)
hexagon	ESDP-BCD	0.32 $\pm$ 0.09 (1.08 $\pm$ 0.23)
	ESDP-BCD (refined with RTR)	0.40 $\pm$ 0.11 (1.08 $\pm$ 0.23)
	RE	0.35 $\pm$ 0.07 (0.75 $\pm$ 0.15)
	SDP	0.31 $\pm$ 0.08 (0.86 $\pm$ 0.17)
	SDP (refined with RTR)	0.37 $\pm$ 0.10 (0.86 $\pm$ 0.17)
	SOC-BCD	0.82 $\pm$ 0.36 (1.82 $\pm$ 0.43)

TABLE II: Numerical results of the proposed and alternatives methods. The methods labeled using **magenta** are proposed in this paper. The accuracy metric RMSE is presented as 'RMSE in body reference systems after refinement (RMSE in the common reference system before refinement)'. The results of RE are obtained from 20 times of data generation, and other experiment methods and setups are consistent with those of Fig. 7.

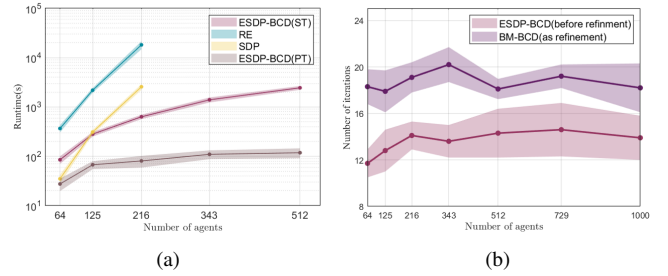


Fig. 8: **Scalability of the methods under various number of agents.** The evaluation is performed in a system with l^3 agents, where l is a positive integer, and the other parameters are consistent with the **cube** problem. The initial guess of each agent's state is randomly distributed in the cube for a relatively fair comparison. Statistics are computed under 20 times of data generation for ESDP-BCD and SDP, 20 times for RE at less than 216 agents, and 4 times at 216 agents. When the number of agents exceeds 216, the SDP and RE methods cause the computer to be out-of-memory.

exhibits scalability that SDP and RE *cannot* match due to its decomposable structure and the adoption of the BCD algorithmic framework. In particular, the number of iterations in both stages of ESDP-BCD remains approximately *constant* as the system is scaled up, as highlighted in Fig. 8(b).

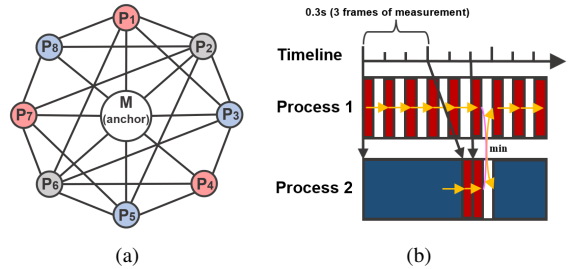


Fig. 9: **Illustration of the mothership-parasite system.** (a) The measurement topology and its coloring scheme, where 'M' indicates the mothership and P_i indicates the i -th parasite. (b) The pipeline of the combination of BM-BCD and ESDP-BCD. The figure shows how each of the two processes works and coordinates with each other, where the red and blue rectangles indicate the execution of BM-BCD and ESDP-BCD, respectively, the yellow arrows indicate the estimation results of one frame offering initialization for the subsequent optimization, and the pink curves indicate the fusion of the estimation results from the two processes (by taking the one with the lower objective function value).

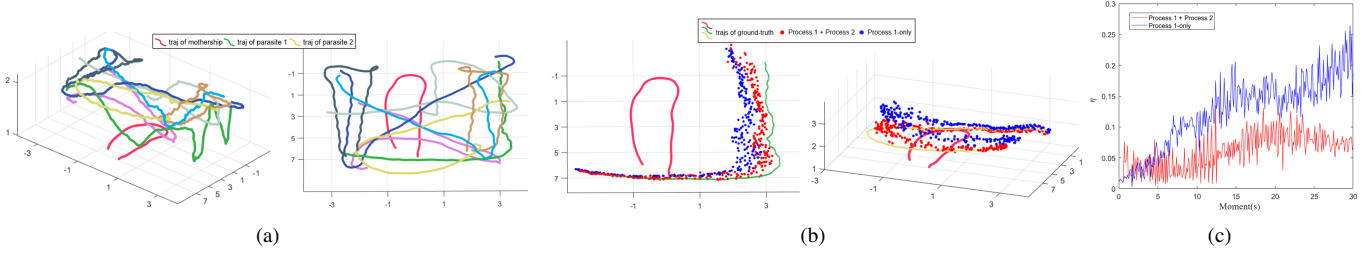


Fig. 10: **Ground-truth and estimates from the two methods on the mothership-parasite system.** (a) The 3D view (left) and XY plane view (right) of the ground-truth trajectories of each agent. (b) Estimation results of parasite 1 (left) and parasite 2 (right) from 'Process 1 + Process 2' (red) and 'Process 1-only' (blue) in the body frame of mothership at each moment. (c) The relative estimation error of the parasite 2 in the body frame of mothership at each moment.

C. A Heterogeneous System in a Continuous-Time Scenario

One of the classic applications of relative state estimation is the *mothership-parasite system* [47, 48]. In such a system, the mothership is designed to allow for large sensor baselines and an assembly of more sensors that *serve as natural anchors*. In contrast, the parasite supports only relatively small sensor baselines. We show that combining BM-BCD and ESDP-BCD, i.e., an algorithmic redundancy design, leads to more consistent estimation over time.

Real-world platform setup. We briefly describe the experimental platform and postpone the details of in Appendix F, which explains how we collect data and calibrate the bias of the UWB distance sensors [2, 49]. The distance measurements after smoothing are sampled at 10 Hz. The system consists of one mothership and 8 parasites. There are 4 distance sensors on the mothership with the furthest interval distance around 2.3m. Parasites are equipped with two distance sensors on each with baselines around 0.7m. Each agent is equipped with an IMU. The ground-truth trajectories of different agents are illustrated in Fig. 10(a), each of which is represented by a separate color. The measurement topology between agents and the coloring scheme are shown in Fig. 9(a).

Methods. BM-BCD is combined with ESDP-BCD in a flow illustrated in Fig. 9(b). In the system of this section, the BM-BCD($d+1$) can operate up to 50 Hz, while the ESDP-BCD can easily operate up to 2 Hz, taking sensors on the mothership as anchors. The method of combination is described below: (a) On each frame of measurements, Process 1 immediately performs online inference using BM-BCD($d+1$) initialized with the estimate from the last measurement; (b) Process 2 reads the latest distance measurement and infers using ESDP-BCD with the latest estimate as the initialization. Since ESDP-BCD cannot infer at the similar high frequency as BM-BCD, the results of ESDP-BCD may exhibit a latency. To handle this latency, Process 2 samples the measurements generated during the computation of ESDP-BCD in a 0.3s interval and perform BM-BCD($d+1$) with the results of ESDP-BCD as the initialization until the latest frame of measurements have been used. The solution from the two processes with a lower objective function value serves as the final estimate for that frame. Process 1 also runs standalone for comparison.

Results. We show the results of employing both pipelines, 'Process 1-only' or 'Process 1 + Process 2', in Fig. 10(a). Since the absolute precision of estimates is related to the

relative position between agents, we use a feature number $\eta = \text{RMSE}/\underline{d}_{ij}$, where $\underline{d}_{ij}$ is the ground-truth distance between agent i and j , to define a relative precision of the estimation, as used in Fig. 10(c). It can be seen that η can increase over time when only executing Process 1, while executing 'Process 1 + Process 2' results in a more consistent η . We suppose that the estimation 'drift' of the local search method occurs because the solution of BM-BCD($d+1$) is affected by initializations, i.e., the last estimate in our methods: because of the near-continuous variation of the measurements over time, the landscape of the optimization also varies continuously, such that similar initializations lead to similar local minima in the continuous-time scenario. Therefore, once BM-BCD($d+1$) falls into a shallow local minima, it may be stuck in similar local minima in the subsequent few frames. In contrast, executing ESDP-BCD periodically to 'shuffle' the estimates alleviates such a problem due to the robustness to arbitrary initializations of ESDP-BCD.

VIII. CONCLUSION AND FUTURE WORK

This paper presents a systematic algorithmic scheme for the problem of distance-based multi-agent relative state estimation, demonstrating that combining scalability and robustness is possible. Both the scalable global estimation and robust local search stages significantly contribute to the results. BCD on the multiconvex structures is the key to efficient and safe algorithmic implementation and scalability.

The optimization problem in this paper belongs to the general scope of polynomial optimization problems (POPs) [50], which is widely emerging in the field of robotics [30, 51, 52]. The current results are limited to formulations including only certain terms in the canonical basis of polynomial, while it can be untrivial to generalize a similar idea of constructing multiconvex and decomposable models to general POPs. Nevertheless, we wish to bring the scalable/parallelizable solution procedure achieved in this paper to more POP-based robotics problems in the future, especially those that are time-critical such as trajectory optimization [51].

Moreover, a recent work [53] demonstrates that the solution of SOCP with *low-dimensional but massive* constraints, such as (32)-(34), can be accelerated by magnitudes with a specialized solution algorithm. We hypothesize that ESDP-BCD is potentially largely accelerated by exploiting such a configuration of constraints with a dedicated solver.

REFERENCES

- [1] Thien Hoang Nguyen and Lihua Xie. Relative transformation estimation based on fusion of odometry and uwb ranging data. *IEEE Transactions on Robotics*, 2023.
- [2] Andrew Fishberg and Jonathan P How. Multi-agent relative pose estimation with uwb and constrained communications. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 778–785. IEEE, 2022.
- [3] Mohammed Shalaby, Charles Champagne Cossette, James Richard Forbes, and Jerome Le Ny. Relative position estimation in multi-agent systems using attitude-coupled range measurements. *IEEE Robotics and Automation Letters*, 6(3):4955–4961, 2021.
- [4] Charles Champagne Cossette, Mohammed Ayman Shalaby, David Saussié, Jérôme Le Ny, and James Richard Forbes. Optimal multi-robot formations for relative pose estimation using range measurements. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2431–2437. IEEE, 2022.
- [5] Hao Xu, Luqi Wang, Yichen Zhang, Kejie Qiu, and Shaojie Shen. Decentralized visual-inertial-uwb fusion for relative state estimation of aerial swarm. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 8776–8782. IEEE, 2020.
- [6] Yulun Tian, Yun Chang, Fernando Herrera Arias, Carlos Nieto-Granda, Jonathan P How, and Luca Carlone. Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems. *IEEE Transactions on Robotics*, 38(4), 2022.
- [7] Saptarshi Bandyopadhyay, Soon-Jo Chung, and Fred Y Hadaegh. Probabilistic and distributed control of a large-scale swarm of autonomous agents. *IEEE Transactions on Robotics*, 33(5):1103–1123, 2017.
- [8] Augustinos D Saravanos, Yuichiro Aoyama, Hongchang Zhu, and Evangelos A Theodorou. Distributed differential dynamic programming architectures for large-scale multiagent control. *IEEE Transactions on Robotics*, 2023.
- [9] Augustinos D Saravanos, Yihui Li, and Evangelos A Theodorou. Distributed hierarchical distribution control for very-large-scale clustered multi-agent systems. *arXiv preprint arXiv:2305.18718*, 2023.
- [10] Trevor Halsted and Mac Schwager. The riemannian elevator for certifiable distance-based localization. *Preprint*, 2022.
- [11] Leo Liberti, Carlile Lavor, Nelson Maculan, and Antonio Mucherino. Euclidean distance geometry and applications. *SIAM review*, 56(1):3–69, 2014.
- [12] Yangyang Xu and Wotao Yin. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on imaging sciences*, 6(3): 1758–1789, 2013.
- [13] Dimitri Bertsekas and John Tsitsiklis. *Parallel and distributed computation: numerical methods*. Athena Scientific, 2015.
- [14] Berthold KP Horn. Closed-form solution of absolute orientation using unit quaternions. *Josa a*, 4(4):629–642, 1987.
- [15] Hayato Waki, Sunyoung Kim, Masakazu Kojima, and Masakazu Muramatsu. Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity. *SIAM Journal on Optimization*, 17(1):218–242, 2006.
- [16] Zizhuo Wang, Song Zheng, Yinyu Ye, and Stephen Boyd. Further relaxations of the semidefinite programming approach to sensor network localization. *SIAM Journal on Optimization*, 19(2):655–673, 2008.
- [17] Thomas Ziegler, Marco Karrer, Patrik Schmuck, and Margarita Chli. Distributed formation estimation via pairwise distance measurements. *IEEE Robotics and Automation Letters*, 6(2):3017–3024, 2021.
- [18] David M Rosen, Luca Carlone, Afonso S Bandeira, and John J Leonard. Se-sync: A certifiably correct algorithm for synchronization over the special euclidean group. *The International Journal of Robotics Research*, 38(2-3):95–125, 2019.
- [19] Yulun Tian, Kasra Khosoussi, David M Rosen, and Jonathan P How. Distributed certifiably correct pose-graph optimization. *IEEE Transactions on Robotics*, 37(6):2137–2156, 2021.
- [20] Nicolas Boumal. A riemannian low-rank method for optimization over semidefinite matrices with block-diagonal constraints. *arXiv preprint arXiv:1506.00575*, 2015.
- [21] Yulun Tian, Kasra Khosoussi, and Jonathan P How. Block-coordinate minimization for large sdps with block-diagonal constraints. *arXiv preprint arXiv:1903.00597*, 2019.
- [22] Afonso S Bandeira, Nicolas Boumal, and Amit Singer. Tightness of the maximum likelihood semidefinite relaxation for angular synchronization. *Mathematical Programming*, 163:145–167, 2017.
- [23] Anthony Man-Cho So and Yinyu Ye. Theory of semidefinite programming for sensor network localization. *Mathematical Programming*, 109(2-3):367–384, 2007.
- [24] Alan Papalia, Andrew Fishberg, Brendan W O’Neill, Jonathan P How, David M Rosen, and John J Leonard. Certifiably correct range-aided slam. *arXiv preprint arXiv:2302.11614*, 2023.
- [25] David M Rosen. Scalable low-rank semidefinite programming for certifiably correct machine perception. In *Algorithmic Foundations of Robotics XIV: Proceedings of the Fourteenth Workshop on the Algorithmic Foundations of Robotics 14*, pages 551–566. Springer, 2021.
- [26] Frederike Dümbgen, Connor Holmes, and Timothy D Barfoot. Safe and smooth: Certified continuous-time range-only localization. *IEEE Robotics and Automation Letters*, 8(2):1117–1124, 2022.
- [27] Frederike Dümbgen, Connor Holmes, Ben Agro, and Timothy D Barfoot. Toward globally optimal state

- estimation using automatically tightened semidefinite relaxations. *arXiv preprint arXiv:2308.05783*, 2023.
- [28] Micah Corah and Nathan Michael. Efficient online multi-robot exploration via distributed sequential greedy assignment. In *Robotics: Science and Systems*, volume 13. Cambridge, MA, 2017.
- [29] Qingjiang Shi, Chen He, Hongyang Chen, and Lingge Jiang. Distributed wireless sensor network localization via sequential greedy optimization algorithm. *IEEE Transactions on Signal Processing*, 58(6):3328–3340, 2010.
- [30] Filip Marić, Matthew Giamou, Adam W Hall, Soroush Khoubyarian, Ivan Petrović, and Jonathan Kelly. Riemannian optimization for distance-geometric inverse kinematics. *IEEE Transactions on Robotics*, 38(3):1703–1722, 2021.
- [31] Matthew Giamou, Filip Marić, David M Rosen, Valentin Peretroukhin, Nicholas Roy, Ivan Petrović, and Jonathan Kelly. Convex iteration for distance-geometric inverse kinematics. *IEEE Robotics and Automation Letters*, 7(2):1952–1959, 2022.
- [32] Nikolas Trawny and Stergios I Roumeliotis. On the global optimum of planar, range-based robot-to-robot relative pose estimation. In *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3200–3206. IEEE, 2010.
- [33] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear matrix inequalities in system and control theory*. SIAM, 1994.
- [34] Chuangchuang Sun and Ran Dai. Rank-constrained optimization and its applications. *Automatica*, 82:128–136, 2017.
- [35] Yangyang Xu and Wotao Yin. A globally convergent algorithm for nonconvex optimization based on block coordinate update. *Journal of Scientific Computing*, 72(2):700–734, 2017.
- [36] Jinshan Zeng, Tim Tsz-Kit Lau, Shaobo Lin, and Yuan Yao. Global convergence of block coordinate descent in deep learning. In *International conference on machine learning*, pages 7313–7323. PMLR, 2019.
- [37] Leonid Barenboim and Michael Elkin. Distributed graph coloring: Fundamentals and recent developments. *Synthesis Lectures on Distributed Computing Theory*, 4(1):1–171, 2013.
- [38] Stephen J Wright. Coordinate descent algorithms. *Mathematical programming*, 151(1):3–34, 2015.
- [39] Sanjay Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on optimization*, 2(4):575–601, 1992.
- [40] Jasson DM Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*, pages 713–719, 2005.
- [41] Nathan Srebro, Jason Rennie, and Tommi Jaakkola. Maximum-margin matrix factorization. *Advances in neural information processing systems*, 17, 2004.
- [42] Samuel Burer and Renato DC Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical programming*, 95(2):329–357, 2003.
- [43] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.
- [44] Nicolas Boumal, Bamdev Mishra, P-A Absil, and Rodolphe Sepulchre. Manopt, a matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research*, 15(1):1455–1459, 2014.
- [45] Paul Tseng. Second-order cone programming relaxation of sensor network localization. *SIAM Journal on Optimization*, 18(1):156–185, 2007.
- [46] Mosek ApS. Mosek optimization toolbox for matlab. *User’s Guide and Reference Manual, Version*, 4:1, 2019.
- [47] John J Leonard and Alexander Bahr. Autonomous underwater vehicle navigation. *Springer handbook of ocean engineering*, pages 341–358, 2016.
- [48] Alex Alcocer, Paulo Oliveira, and Antonio Pascoal. Underwater acoustic positioning systems based on buoys with gps. In *Proceedings of the Eighth European Conference on Underwater Acoustics*, volume 8, pages 1–8, 2006.
- [49] Andrew Fishberg, Brian Quiter, and Jonathan P How. Murp: Multi-agent ultra-wideband relative pose estimation with constrained communications in 3d environments. *arXiv preprint arXiv:2312.17731*, 2023.
- [50] Jean Bernard Lasserre. *Moments, positive polynomials and their applications*, volume 1. World Scientific, 2009.
- [51] Sangli Teng, Ashkan Jasour, Ram Vasudevan, and Maani Ghaffari. Convex geometric motion planning on lie groups via moment relaxation. *arXiv preprint arXiv:2305.13565*, 2023.
- [52] Heng Yang and Luca Carlone. Certifiably optimal outlier-robust geometric perception: Semidefinite relaxations and scalable global optimization. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):2816–2834, 2022.
- [53] Qianhao Wang, Zhepei Wang, Mingyang Wang, Jialin Ji, Zhichao Han, Tianyue Wu, Rui Jin, Yuman Gao, Chao Xu, and Fei Gao. Fast iterative region inflation for computing large 2-d/3-d convex regions of obstacle-free space. *arXiv preprint arXiv:2403.02977*, 2024.

APPENDIX A PROOF OF PROPOSITION 1

Proof: The distance-only setup. In the distance-only setup, the aspect (a) is straightforward due to the rigid body invariance rendered by rotation. For the aspect (b),

- when $|\mathcal{B}_i| = 2$ or all the p_i^u are colinear, it is clear that there must exist infinite R_i that satisfies (10), i.e., the aspect (b) holds.
- When $|\mathcal{B}_i| = 3$ and (6) holds, we can apply the result in [14, Section 5] to register any $\hat{p}_i^u, \hat{p}_i^v, \hat{p}_i^w$, where $u < v < w$ and $\hat{p}_i^u, \hat{p}_i^v, \hat{p}_i^w$ are non-collinear, to $\bar{p}_i^u, \bar{p}_i^v, \bar{p}_i^w$, with a unique solution [14, Section 2]. That is, there must exist

a unique (R_i, t_i) that satisfies (10) so that the aspect (b) holds.

- When $|\mathcal{B}_i| > 3$ but all the points are coplanar, we can choose any 3 non-collinear points to compute a unique solution (R_i, t_i) as explained above. Then the other points on the plane can be uniquely determined by these 3 points with known distances by (6) from these points to the chosen 3 points. That is, if (6) holds and all the points are coplanar, there must exist a unique (R_i, t_i) that satisfies (10).

We note that, if $|\mathcal{B}_i| > 3$ and the points are non-coplanar, we can choose 3 non-collinear points to compute a unique (R_i, t_i) , and it can be validated that there exist two sets of other points satisfying (6) with *handedness symmetry* about the plane consisting of the 3 chosen points. Due to the uniqueness of (R_i, t_i) , it can be clearly shown that one of the sets must not satisfy condition (10) so that the aspect (b) does not hold. That is, the statement 'coplanar' is necessary.

The 4-axis proprioception setup. Firstly, we can show that

$$\begin{aligned} p_i^u &= R_i \bar{v}_i^u + t_i, \forall i \in \mathcal{V}, u \in \mathcal{B}_i \Leftrightarrow \\ p_i^u - p_i^v &= R_i (\bar{v}_i^u - \bar{v}_i^v), \forall i \in \mathcal{V}, u, v \in \mathcal{B}_i. \end{aligned} \quad (36)$$

When 4-axis proprioception is available, the right-hand side of (36) can be specially written as

$$\begin{aligned} p_i^u - p_i^v &= R_i (\bar{v}_i^u - \bar{v}_i^v) \\ &= \begin{bmatrix} \cos \psi_i & -\sin \psi_i & 0 \\ \sin \psi_i & \cos \psi_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{R}_{i|y} \hat{R}_{i|x} (\bar{v}_i^u - \bar{v}_i^v), \end{aligned} \quad (37)$$

where $\hat{R}_{i|y}$ and $\hat{R}_{i|x}$ are the measured rotation components.

Introducing $[\bar{v}_{i|x}; \bar{v}_{i|y}; \bar{v}_{i|z}] := \hat{R}_{i|y} \hat{R}_{i|x} (\bar{v}_i^u - \bar{v}_i^v)$ for the sake of concise writing, a straightforward computation shows the following relations from the first two rows of (37):

$$\sin \hat{\psi}_i = (\bar{v}_{i|x} (\hat{y}_i^u - \hat{y}_i^v) - \bar{v}_{i|y} (\hat{x}_i^u - \hat{x}_i^v)) / (\bar{v}_{i|x}^2 + \bar{v}_{i|y}^2), \quad (38)$$

$$\cos \hat{\psi}_i = (\bar{v}_{i|x} (\hat{x}_i^u - \hat{x}_i^v) + \bar{v}_{i|y} (\hat{y}_i^u - \hat{y}_i^v)) / (\bar{v}_{i|x}^2 + \bar{v}_{i|y}^2), \quad (39)$$

both of which hold for any $i \in \mathcal{V}$ and $u, v \in \mathcal{B}_i$. So far, a one-to-one correspondence between $\{(p_i^u, p_i^v)\}$ and $\{(R_i, t_i)\}$ is found, which is used in Proposition 2 to recover states from the sensor coordinates.

Analogously, the last row of (37) gives that

$$\begin{aligned} z_i^u - z_i^v &= \bar{v}_{i|z} \\ &= [-\sin \tilde{\phi}_i, \cos \tilde{\phi}_i \sin \tilde{\theta}_i, \cos \tilde{\phi}_i \cos \tilde{\theta}_i] (\bar{v}_i^u - \bar{v}_i^v), \end{aligned} \quad (40)$$

which appears in (7).

The remaining step now is to eliminate the $\hat{\psi}_i$ in (38) and (39) and meanwhile preserve the constraints implied therein, thus obtaining a set of relations equivalent to those in (37). First, we have from (38)

$$\begin{aligned} \sin \hat{\psi}_i &= (\bar{v}_{i|x} (\hat{y}_i^u - \hat{y}_i^v) - \bar{v}_{i|y} (\hat{x}_i^u - \hat{x}_i^v)) / (\bar{v}_{i|x}^2 + \bar{v}_{i|y}^2) \\ &= (\bar{v}_{i|x} (\hat{y}_i^u - \hat{y}_i^w) - \bar{v}_{i|y} (\hat{x}_i^u - \hat{x}_i^w)) / (\bar{v}_{i|x}^2 + \bar{v}_{i|y}^2), \\ &\quad \forall u, v, w \in \mathcal{B}, u < v < w. \end{aligned} \quad (41)$$

A similar relationship can be obtained from (39). These two equalities appear in (8).

Moreover, (38) and (39) impose that $\sin^2 \hat{\psi}_i + \cos^2 \hat{\psi}_i = 1$, i.e.,

$$(\hat{x}_i^u - \hat{x}_i^v)^2 + (\hat{y}_i^u - \hat{y}_i^v)^2 = \bar{v}_{i|x}^2 + \bar{v}_{i|y}^2,$$

which can be combined with (40) to obtain a relation in a more compact form

$$(\hat{x}_i^u - \hat{x}_i^v)^2 + (\hat{y}_i^u - \hat{y}_i^v)^2 + (\hat{z}_i^u - \hat{z}_i^v)^2 = \bar{v}_{i|x}^2 + \bar{v}_{i|y}^2 + \bar{v}_{i|z}^2,$$

which is exactly the calibration constraint (6).

From the above derivation, we can state that given any sensor coordinates satisfying (6) (7) and (8), we can recover an $\{R_i\}$ satisfying (37), i.e., satisfying (10) according to (36), so that the aspect (b) of Proposition 1 holds. The aspect (a) is also naturally guaranteed by the above derivation.

The 6-axis proprioception setup. Eq. (9) is an instance of (36) with a fully observed R_i . The aspects (a) and (b) hold with no elaboration. $\square$

APPENDIX B

FINDING A EQUIVALENT SUBSET OF CONSTRAINTS IN PROBLEM 2

In this appendix, we follow an intuitive geometric idea to find an equivalent subset of the constraints of Problem 2.

The distance-only setup. It is well known that in a plane, a point can be uniquely determined by known distances to any 3 non-collinear known points. According to this, when $|\mathcal{B}_i| \geq 4$, supposing there exist 3 non-collinear distance sensors, we first choose 3 non-collinear sensors, denoted as p_i^0, p_i^1 and p_i^2 , and an equivalent subset of (6) for agent i can be written as

$$\|p_i^u - p_i^v\|^2 = \|\bar{v}_i^u - \bar{v}_i^v\|^2, \forall u \in \{0, 1, 2\}, v \in \mathcal{B}_i, u < v. \quad (42)$$

The above set actually includes the minimum number of constraints to ensure equivalence with (6).

When all the sensors are colinear, we only need to choose 2 sensors first and only maintain the calibration constraints between the other sensors and these 2 sensors.

The 4-axis proprioception setup. When $|\mathcal{B}_i| \geq 5$ and 4 non-coplanar sensors exist, we follow a similar idea to the distance-only setup: considering that a point in space can be uniquely determined by known distances to any 4 non-coplanar points, we first specify 4 'anchor' non-coplanar distance sensors that satisfy both aspects of Proposition 1. In this way, only the calibration constraints (6) between other sensors and these anchor sensors are needed to preserved to guarantee the other sensors satisfy Proposition 1, while we can discard the proprioception constraints (7)-(8) between non-anchor sensors and the anchors.

Therefore, we only need to preserve necessary constraints between the 4 anchors and the calibration constraints between non-anchor sensors and the anchors. Specially, we choose to only maintain all the linearly independent proprioception constraints among these 4 sensors⁴ and the constraints of (6)

⁴Although in some cases it is possible further to reduce the constraint set of these 4 anchor sensors, it requires an exhaustive case-by-case discussion

between them and the other sensors. Let these 4 anchor sensors be p_i^0, p_i^1, p_i^2 and p_i^3 , we can write this equivalent subset of the constraints as

$$\begin{aligned} \|p_i^u - p_i^v\|^2 &= \|\bar{p}_i^u - \bar{p}_i^v\|^2, \forall u \in \{0, 1, 2, 3\}, v \in \mathcal{B}_i, u < v \\ z_i^u - z_i^v &= [-\sin \tilde{\phi}_i, \cos \tilde{\phi}_i \sin \tilde{\theta}_i, \cos \tilde{\phi}_i \cos \tilde{\theta}_i] (\bar{p}_i^u - \bar{p}_i^v), \\ u, v, w &\in \{0, 1, 2, 3\}, u < v < w, \end{aligned} \quad (43)$$

$$\begin{aligned} (\bar{p}_{i|x}(y_i^u - y_i^v) - \bar{p}_{i|y}(x_i^u - x_i^v)) &= (\bar{p}_{i|x}(y_i^v - y_i^w) - \bar{p}_{i|y}(x_i^v - x_i^w)), \\ (\bar{p}_{i|x}(x_i^u - x_i^v) + \bar{p}_{i|y}(y_i^u - y_i^v)) &= (\bar{p}_{i|x}(x_i^v - x_i^w) + \bar{p}_{i|y}(y_i^v - y_i^w)), \\ u, v, w &\in \{0, 1, 2, 3\}, u < v < w, \end{aligned} \quad (44)$$

where we do not explicitly remove the linearly dependent constraints, as it is trivial to identify them.

When all the sensors are coplanar, we just need first to characterize 3 anchor sensors and redo the above derivation to obtain a set of constraints analogous to (42) and (45).

The 6-axis proprioception setup. This setup is very similar to a distance-only setup, so we directly give the result when $|\mathcal{B}_i| \geq 5$ as follows:

$$p_i^u - p_i^v = \hat{R}_i (\bar{p}_i^u - \bar{p}_i^v), \forall u \in \{0, 1, 2, 3\}, v \in \mathcal{B}_i, u < v. \quad (46)$$

APPENDIX C

DERIVATION OF RESULTS ABOUT ESDP

A. Proof of Proposition 3

Proof: If a certain sensor (i, u) is not uniquely localizable, then there are at least two different solutions of p_i^u in the solution set. On the other hand, Theorem 4.2 in [16] says that if $X_i^u = (p_i^u)^T p_i^u$, then p_i^u is invariant in the solution set. So we must have $X_i^u > (p_i^u)^T p_i^u$ considering constraint (24) (which implies that the relaxation is not tight). $\square$

B. Derivation of Eq. (32) and Eq. (33)

Constraint (24) can be rewritten in a similar form as (12):

$$\begin{bmatrix} X_{ii}^{uu} & X_{ij}^{uv} \\ X_{ij}^{uv} & X_{jj}^{vv} \end{bmatrix} \succeq [p_i^u \quad p_j^v]^T [p_i^u \quad p_j^v],$$

i.e.,

$$\begin{bmatrix} X_{ii}^{uu} - (p_i^u)^T p_i^u & X_{ij}^{uv} - (p_i^u)^T p_j^v \\ X_{ij}^{uv} - (p_j^v)^T p_i^u & X_{jj}^{vv} - (p_j^v)^T p_j^v \end{bmatrix} \succeq 0,$$

of which all of the leading principal minors in the matrix are nonnegative, i.e.,

$$X_{ii}^{uu} - (p_i^u)^T p_i^u \geq 0,$$

$$\begin{aligned} &\begin{vmatrix} X_{ii}^{uu} - (p_i^u)^T p_i^u & X_{ij}^{uv} - (p_i^u)^T p_j^v \\ X_{ij}^{uv} - (p_j^v)^T p_i^u & X_{jj}^{vv} - (p_j^v)^T p_j^v \end{vmatrix} = \\ &(X_{ii}^{uu} - (p_i^u)^T p_i^u) (X_{jj}^{vv} - (p_j^v)^T p_j^v) - \\ &\quad (X_{ij}^{uv} - (p_i^u)^T p_j^v)^2 \geq 0. \end{aligned} \quad (47)$$

Fixing the components corresponding to the sensor on agent j , i.e., p_j^v and X_{jj}^{vv} , in (47), we can obtain the constraint (33) by organizing.

APPENDIX D

PROOF OF LEMMAS IN SECTION VI-C

A. Proof of Lemma 3

Proof: **Boundness of a single element in Problem 5.**

Assume (i, u) is a sensor that is directly connected to a prior-known sequence-bounded sensor (a, v) guaranteed to exist by Assumption 2. Then we have

$$\begin{aligned} \kappa((U_i^u - U_a^v)^T (V_i^u - V_a^v) - \tilde{q}_{ia}^{uv})^2 + \gamma \|U_i^u - V_i^u\|_F^2 \\ \leq F(\bar{U}, \bar{V}), \end{aligned} \quad (48)$$

where (U_i^u, V_i^u) and (U_a^v, V_a^v) are the corresponding surrogates/column of sensor (i, u) and (a, v) in variables U and V , which are in the level set $\mathcal{H}_F((\bar{U}, \bar{V}))$, where $F(\cdot)$ is the objective function in Problem 5. κ in (48) is $\frac{1}{(\sigma_{ia}^{uv})^2}$ in (5) if the sensor (i, u) and the sensor (a, v) are on the different agents, or is λ in (29) if they are on the same agent. According to Assumption 2, $\{(p_a^v)^{(k)}\}$, i.e., $\{((U_a^v)^{(k)}, (V_a^v)^{(k)})\}$, is bounded. Therefore, $\{((U_i^u)^{(k)}, (V_i^u)^{(k)})\}$ must be bounded, otherwise (48) does not hold, so that variables corresponding to any sensor connected (on the same agent as or have a measurement with) the sensor (i, u) can be proved to be bounded in the same way.

Boundness of a single element in Problem 4. We first have (32) so that

$$\begin{aligned} &\sqrt{\kappa} \left((p_i^v)^T p_i^u - 2(p_a^v)^T p_i^u + X_{aa}^{vv} - \tilde{q}_{ia}^{uv} \right) \\ &\leq \sqrt{\kappa} \left(X_{ii}^{uu} - 2(p_a^v)^T p_i^u + X_{aa}^{vv} - \tilde{q}_{ia}^{uv} \right) \leq \sqrt{G(\bar{Z})}, \end{aligned} \quad (49)$$

where the variables are in the level set $\mathcal{H}_G(\bar{Z})$ as previous and $G(\bar{Z})$ is the objective function in Problem 4. According to Assumption 2, $\{(p_a^v)^{(k)}\}$ is bounded, so that p_i^u must be bounded indicated by (49). Then, we immediately have $X_{ii}^{uu} \leq \sqrt{G(\bar{Z})/\kappa} + 2(p_a^v)^T p_i^u - X_{aa}^{vv} + \tilde{q}_{ia}^{uv}$ from (49), which indicates that X_{ii}^{uu} is also bounded.

Let (j, w) be a sensor directly connected with sensor (i, u) . We have

$$X_{ii}^{uu} X_{jj}^{ww} \geq (X_{ij}^{uw})^2 \quad (50)$$

so that

$$\begin{aligned} &\sqrt{\kappa} \left(X_{ii}^{uu} - 2\sqrt{X_{ii}^{uu} X_{jj}^{ww}} + X_{jj}^{ww} - \tilde{q}_{ij}^{uw} \right) \\ &\leq \sqrt{\kappa} (X_{ii}^{uu} - 2X_{ij}^{uw} + X_{jj}^{ww} - \tilde{q}_{ij}^{uw}) \leq \sqrt{G(\bar{Z})}. \end{aligned} \quad (51)$$

Since X_{ii}^{uu} is bounded, X_{jj}^{ww} is bounded from (51). According to (32) and (50), p_j^w and X_{ij}^{uw} are also bounded. The above proof can be applied in the same way between different sensors on the same agent.

Boundness of all elements. Since Assumption 2 says that any sensor is directly or indirectly connected to a prior-known bounded-sequence sensor, then we can traverse any path in $\mathcal{G}$ from the bounded-sequence sensor to that sensor and apply the above proof in succession between neighboring sensors. In this way, the boundedness of the variable corresponding to any sensor can be obtained. $\square$

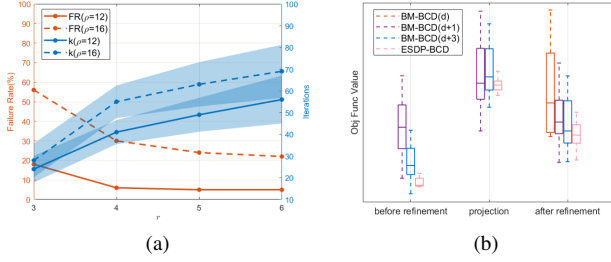


Fig. 11: **BM-BCD with various r and under harsh initializations.** Statistics are computed under 100 times of data generation. (a) The failure rate and iterations needed for each r . (b) The resulting objective function value of different stages in the solution process, where ‘before refinement’, ‘projection’, and ‘after refinement’ respectively indicate the objective function values of Problem 4 or Problem 5, the objective function values of Problem 2 at estimates extracted from Problem 5 (i.e., $Q^T U$) or Problem 4 (the (1,2) block of Z), and the objective function values of Problem 2 after refinement.

B. Proof of Lemma 4

Proof: We follow a similar way to Proposition 3.9 in [13] to prove this lemma. Let $\mathbf{x}^* = (x_1^*, \dots, x_p^*)$ be a Nash equilibrium of (17) such that

$$f(x_1^*, \dots, x_q^*, \dots, x_p^*) \leq f(x_1^*, \dots, x_q, \dots, x_p^*), \quad \forall x_q \in \mathcal{X}_q$$

holds for all $q = 1, \dots, p$ according to Definition 3, i.e., x_q^* is a global minimizer of $f(x_1^*, \dots, x_q, \dots, x_p^*)$. So we have

$$\nabla_q f(\mathbf{x}^*)^T d = \nabla f_q^*(x_q^*)^T d \geq 0, \quad \forall d \in \mathcal{T}_{\Omega_q}^*(x_q^*) \quad (52)$$

where $f_q^*(x_q) := f(x_q; \hat{\mathbf{x}}_{[p]/\{q\}}^*)$ is the block objective function of x_q with remaining blocks fixed as $\mathbf{x}^*$ and $\mathcal{T}_{\Omega_q}^*(x_q^*)$ is the tangent cone [43, Definition 12.2] of $f_q^*(\cdot)$ at x_q^* .

Since the block division makes $\mathcal{X}$ a Cartesian product over each block, it follows that (52) holds for d in all tangent cones of arbitrarily fixed remaining blocks at x_q^* . So by adding (52) over all q , we have

$$\nabla f(\mathbf{x}^*)^T d \geq 0, \quad \forall d \in \mathcal{T}_{\Omega}(\mathbf{x}^*), \quad (53)$$

where $\mathcal{T}_{\Omega}(\mathbf{x}^*)$ is the tangent cone of the full object function $f(\cdot)$. Therefore, $\mathbf{x}^*$ is a local solution of (17) according to Theorem 12.3 in [43] and under some constraint qualification it can be rewritten in the form of the KKT condition. $\square$

APPENDIX E VARIOUS r IN BM-BCD

In this appendix, we study how the parameter r in BM-BCD affects its performance. We conduct experiments in the 3D **pyramid** problem using the same setup as Section VII. We use poor initializations to trigger failure and record the failure rate (FR), defined as the rate RMSE exceeds 60 cm.

Results. From Fig. 11(a), we can see that the failure rate hopefully decreases as the rank r becomes larger. The story behind this observation, which can be revealed in Fig. 11(b), is that a larger r , such as that in BM-BCD($d+3$), can lead to a more consistent solutions under poor initializations, which can be captured by the low variance of the corresponding objective function values. This implies that shallow local minima of the optimization model are seemingly reduced as r

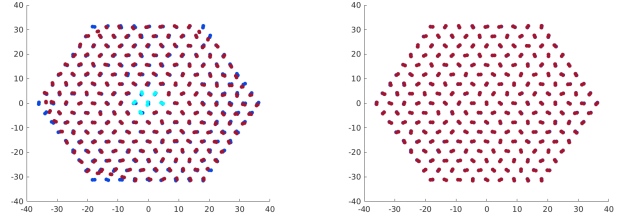


Fig. 12: **Visualization of the problem hexagon and results.** Results before (left) and after (right) refinement of ESDP-BCD in hexagon, where the graphic elements have the same meaning as in Fig. 5.



Fig. 13: **Illustration of the used hardware platforms.** The mothership (left) and parasite (right) used in our experiment. The dark red circles highlights the equipped UWB sensors.

increases. Thus, a higher r provides more consistent (although not always better) initializations for the refinement stage after projecting the high-rank solution onto the realization space than those of lower r . However, we can only empirically claim the robustness from higher-rank spaces so far, although there are remarkable results about the strong relationship between local minima of NLP problems from BM factorization and the global minimizer of a class of SDP [20].

APPENDIX F DETAILED SETUP AND METHOD OF THE REAL-WORLD EXPERIMENT IN SECTION VII-C

A. Hardware Platform

We use a DJI FlyCart 30 quadcopter to serve as our mothership, as shown in Fig. 13(a). One of the platforms serving as the parasite is shown in Fig. 13(b). Nooploop LTP-AS2 UWB module is used to measure distances. We use a motion capture (mocap) system to record the ground-truth trajectories, and the origin of the body coordinate system of each agent is determined by the placement of the mocap markers on it. The calibrated distance sensors on the mothership is $\bar{p}_0^0 = [0.725; -0.897; -0.06]\text{m}$, $\bar{p}_0^1 = [-0.691; -0.897; -0.06]\text{m}$, $\bar{p}_0^2 = [-0.691; 0.837; 0.04]\text{m}$, and $\bar{p}_0^3 = [0.725; 0.837; 0.04]\text{m}$. The calibrated coordinates of the distance sensors on each

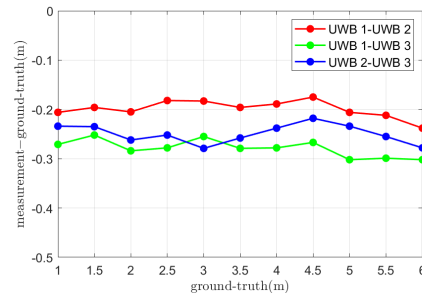


Fig. 14: **Mean biases between different pairs among 3 UWBs.** The y-axis is the mean bias, which is calculated as the average of (measurement-ground-truth) of the collected data.

parasite vary slightly because the layout of mocap markers is different on each parasite, but the length of the baselines are all around 0.7m.

B. Data Processing

As observed in [2, 49], the Nooploop UWB modules generate data in distributions with non-zero means (called *bias*). To narrow the gap from simulation to the reality, we post-process the reading of UWBs to approximately satisfy the zero-mean assumption used in our formulation (see (1)).

Before collecting the dataset on which we run the experiment, we first collect a calibration dataset to decide a strategy for processing measurements. We find that the biases of the measurements are relatively stable when the operating conditions do not vary much, while varying among different pairs of sensors, as shown in Fig. 14. However, these biases can be significantly different (e.g., with a difference greater than 10 cm) when there is occlusion or when a large attitude difference between the two UWB sensors, as previously observed in [49]. When collecting our dataset, we artificially control the relative altitude between UWBs so as not to be too large and avoid occlusions between them. We believe that with further advancement of distance sensors and a more elaborate system design, such manual operation can be avoided.

Once the above pattern is found, we employ a simple calibration scheme: we use the calibration dataset to compute an average bias for each pair of UWBs where measurements exist, and then subtract this bias from measurements generated by the corresponding UWB pair when running the experiment.

C. Notes on the Results

We note that we use a near perfect relative estimate (from the mocap system) to initialize the two methods at moment $t = 0$ so that the local search method does not fall into a shallow local minima at the beginning, making it easier to exhibit the gradually increasing relative error η . When giving a good initialization at beginning, we find that if a higher frequency (e.g., 50Hz) of the measurement is available (we simulate it using high-frequency mocap data), the estimation drift of the local search method is mitigated if we run it at correspondingly high frequencies. This is because when the frequency is high, the initialization, i.e., the estimation of last frame, of the optimization corresponding to each frame of measurement can be viewed as being better-maintained than that of lower frequency of measurements. However, high-frequency online estimation greatly enhances the requirements for onboard computation as well as communication. We therefore believe that the convex relaxation approach has a unique interest in maintaining time-consistent estimates even when the local search frequency is not that high.