

Understanding Hidden Knowledge in Generic Graphs

Haodi Ping[✉], Yongcai Wang[✉], Member, IEEE, Yu Zhang[✉], Deying Li[✉], Member, IEEE,
and Lihua Xie[✉], Fellow, IEEE

Abstract—When the edge between two nodes is not measured, is there any hint to know the edge property, and will the inferred edge property be useful? To answer these questions, this paper uniformly defines the properties of unmeasurable edges in generic graphs. For an unmeasurable edge (i, j) , it is called *rangeable* if its length is unique in any realization of the graph, *rigid* if the number of its possible lengths is finite, and *flexible* if it has infinite possible lengths. The rangeable edge can provide deterministic hidden knowledge as if the edge is measured. A condition for an unmeasured edge being rangeable in 2D space is firstly proposed, based on which a centralized identification algorithm (DRE) is designed. However, the centralized rangeable edge identification has the overhead of global information collection. Therefore distributed condition and algorithm to identify rangeable edges are further investigated. We prove that an unmeasurable edge (i, j) is rangeable if there are at least two Disjoint Minimally Rigid Branches (DMRBs) between i and j . The unmeasurable edge (i, j) is rigid and flexible when the number of DMRB is one and zero, respectively. A distributed *Branching and Blacklisting* (BB) algorithm is proposed to find DMRBs, so that rangeable edges are identified distributively. Then, the applications of rangeable, rigid, and flexible edges are discussed. Experimental evaluations show that the centralized and distributed algorithms can identify a rich set of unmeasurable but rangeable edges in distance graphs, even more than the number of directly measured edges. Moreover, BB has a similar identification performance as the centralized DRE algorithm and outperforms existing distributed unmeasurable edge inference algorithms significantly.

Index Terms—Generic graph, unmeasurable edge, rangeable, rigid, node localizability.

I. INTRODUCTION

IN THE new era of Internet of Things (IoT), the inter-node distances can be measured through various ranging techniques, such as Wi-Fi Round Trip Time (RTT) [1], Ultra-wide

Manuscript received 25 February 2023; revised 23 October 2023; accepted 4 February 2024; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor D. Malone. The work of Yongcai Wang was supported in part by the National Natural Science Foundation of China under Grant 61972404; and in part by the Public Computing Cloud, Renmin University of China. The work of Deying Li was supported in part by the National Natural Science Foundation of China under Grant 12071478. (Corresponding author: Yongcai Wang.)

Haodi Ping is with the Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China (e-mail: haodi.ping@bjut.edu.cn).

Yongcai Wang, Yu Zhang, and Deying Li are with the Department of Computer Sciences, Renmin University of China, Beijing 100872, China (e-mail: ycw@ruc.edu.cn; 2020104230@ruc.edu.cn; deyingli@ruc.edu.cn).

Lihua Xie is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: elhxie@ntu.edu.sg).

Digital Object Identifier 10.1109/TNET.2024.3364177

Band (UWB) [2], LoRa RTT [3], and Received Signal Strength [4]. Using network entities as nodes and inter-node distance measurements as edges, a distance graph can be formulated. Distance graphs have facilitated many applications. For example, UWB ranging technique has enabled social distancing during COVID-19 [5]. The latest smartphones, smartwatches, and IoT devices are also equipped with UWB modules [6], offering the promising prospects of the ranging technique. More application fields include swarm of drones [7], formation of agents [8], multi robots [9], and sensor networks [10]. Location is one of the most fundamental information in these applications. Using distances as constraints, ranging-based localization algorithms [11], [12], [13] can calculate locations without introducing infrastructures. A set of node coordinates satisfying all distance constraints is called the *realization* of a graph [14]. A graph is *generic* if all node coordinates are algebraically independent over the rationals [15].

Along with the ranging-based localization, a question arises: How well can the locations be determined from the distance graph? The edge counting-based rigidity and localizability studies provide guidance, describing the calculated location is unique or ambiguous. The seminal work is Laman condition [16], which defines minimally rigid graphs by counting edges. A graph is *rigid* if it has a spanning Laman graph [14], and *redundantly rigid* if remains rigid after removing any edge. Based on Laman condition, the Pebble Game [17] algorithm finds rigid or redundantly rigid components (subgraphs) in a polynomial time. A graph is *globally rigid* if it is 3-connected and redundantly rigid in 2D [15], meaning that the network is *localizable* (i.e., theoretically has a unique realization). In actual scenarios, a network is generally non-localizable [18], [19], which motivates a series of node localizability conditions and algorithms to identify localizable nodes (i.e., a subset of nodes whose locations can be uniquely determined from the distance graph). Since the necessary and sufficient condition for node localizability is still absent, currently localizable nodes are found through sufficient conditions, including trilateration [20], wheel extension [21], [22], RR3P [23], [24], [25], triangle extension [26], and iterative maximum flow [27].

In the aforementioned rigidity and localizability analysis, a common agreement is that only when the connectivity (i.e., count of edges) of the graph reaches specific conditions can the network or a node be localizable. A notable problem is that only the measured edges (i.e., explicit knowledge) are utilized.

The measured edges in the distance graph act as constraints to reduce nodes' freedom degrees. Many nodes that are actually localizable may be wrongly detected as non-localizable when using sufficient conditions based on measured edges [20], [21], [22], [23], [24], [26], [27], [28], especially in sparse graphs.

For more accurate localizability detection, an immediate idea is to leverage the hidden information in unmeasurable edges. Only piecemeal studies have explored the value of unmeasurable edges, primarily focusing on two-dimensional space (\mathbb{R}^2). Jackson et al. [29], [30] defined that a pair of vertices is globally linked if the distance between them is the same in all equivalent generic realizations. But global linkedness is reported to be not a generic property, i.e., a certain edge may be globally linked in one generic realization, but not in another. Yang et al. [24] defined the edge with unique length in all graph realizations as an "implicit edge". To characterize implicit edge, a sufficient condition indicated that an unmeasured edge shared by two independent rigid components is implicit. However, this condition incurs a combinational number of graph partitions, which is computationally inefficient. It was further indicated that an edge is implicit if its endpoints form a binary vertex cut of a redundantly rigid graph. In this paper, we refer to such edges as *Redundantly Rigid graph Cut (RR-Cut)* Edges. The "RR-Cut edges" can be identified in polynomial time, but their number is shown to be very limited. Oliva et al. [31] proposed "shadow edge". They consider the free node connecting two localizable nodes. The two possible locations of such a node are calculated through the known locations of the localizable nodes. Then, it checks whether one of the two locations falls into the sensing radius of any other localizable nodes. Once the checking succeeds, only one location is feasible and a shadow edge is added. But shadow edges use the Unit Disk Graph (UDG) constraint, which assumes nodes have disk shape sensing region. Guo et al. [32] focused on the flipping ambiguity of UAV networks. The condition when the node locations may flip along an axis without violating the distance constraints is proposed. Using UDG constraint, if the flipping ambiguity is resolved, one edge can be added since the realization becomes unique as if the edge exists. In our previous work [12], the structure with four vertices and five edges (i.e., a K_4 graph with only one edge missing) is defined as the basic flipping graph (BFG). In BFG , the condition to infer the exact length of the missed edge (called "negative edge") is proposed based on the UDG constraint. Therefore, existing unmeasurable edge inference algorithms are piecemeal. Only some specific kind of rangeable edges are inferred by each method, lacking a thorough understanding and a systematic methodology to infer the hidden knowledge.

To address the aforementioned problem, this paper uniformly defines the properties of unmeasured edges. An unmeasured edge is *rangeable* if its length is unique in any realization of the graph, *rigid* if it has a finite number of lengths, and *flexible* if it has infinite possible lengths. The unmeasured edges and their applications will enrich the hidden knowledge provided by distance graphs. We present both centralized and distributed conditions and algorithms to identify rangeable edges. Research on edges of unique length is piecemeal, and

the research deficiency is more acute on rigid and flexible edges. Thus, the identification of rigid and flexible edges and their possible applications are further presented.

The main contributions of this paper are as follows:

(1) The rangeable edge identification is explored since it can provide deterministic knowledge as if the edge is measured. A centralized condition is firstly proposed to characterize when an unmeasured edge is rangeable. Based on the condition, a centralized *Detect Rangeable Edge (DRE)* algorithm is designed to identify rangeable edges.

(2) A distributed condition is proposed to characterize rangeable edges grounded in the concept of Disjoint Minimally Rigid Branches (DMRB). A *Branching and Blacklisting (BB)* algorithm is designed to find DMRBs, thereby enabling the identification of rangeable edges. BB is distributed and can be efficiently conducted through triangle extension. BB can also return rigid and flexible edges by counting DMRB.

(3) The significance of identifying rangeable, rigid, and flexible edges is demonstrated through various applications. (i) The connectivity information introduced by rangeable edges greatly enhances the localizable node detection accuracy, enabling nodes previously misclassified as non-localizable to be correctly identified as localizable; (ii) The rangeable edges are instrumental in mining cohesive sub-graphs that have predictable high localization accuracy, termed *Strongly Localizable Areas (SOLA)*; (iii) An approach to calculate the specific lengths of the inferred rangeable edges is designed; (iv) The rigid edges can relax the localizability condition when using the trilateration idea; The flexible edges can characterize weak regions of a network.

(4) Property analyses and extensive experiments validate the proposed methods. We show both centralized and distributed algorithms can detect a large amount of rangeable but unmeasured edges in distance graphs. The distributed BB algorithm achieves almost the same performance as the centralized DRE algorithm and outperforms existing edge inference algorithms [12], [24], [31], [32] significantly. The inferred hidden information shows great value in the aforementioned applications.

The remaining sections of this paper are organized as follows. Related work is given in Section II. The centralized condition and centralized algorithm for rangeable edge identification are presented in Section III. The distributed condition and distributed algorithm for rangeable, rigid, and flexible edge identification are given in Section IV. The applications of rangeable, rigid, and flexible edges are presented in Section V. Experimental evaluations are in Section VI. The paper is concluded with remarks in Section VII.

II. RELATED WORK

The localizability analysis studies and edge inference methods in distance graphs are introduced.

A. Distance Graph Analysis

In various distance graph applications, it is desired that a network structure can be uniquely determined from a distance graph \mathcal{G} . Whether the structure recovered from \mathcal{G} is

unique or ambiguous can be characterized by a series of edge counting-based graph analysis studies. The fundamental theory is Laman condition [16]. A graph \mathcal{G} with n vertices is a Laman graph in \mathbb{R}^2 if \mathcal{G} has exactly $2n - 3$ edges and any k -vertex subgraph of \mathcal{G} has at most $2k - 3$ edges. A Laman graph is also called a *minimally rigid graph*. A graph is *rigid* if it has a spanning Laman graph [14]. Then, *redundant rigidity* is a higher level of rigidity, meaning that \mathcal{G} remains rigid after removing any edge [15]. Beyond redundant rigidity, \mathcal{G} is *globally rigid* in \mathbb{R}^2 if it is 3-connected and redundantly rigid [15], [33], [34]. The 3-connectivity property means that the removal of any two vertices will not disconnect the graph. Different rigidity levels correspond to different levels of structural recovery potential. A network structure is uniquely determinable when the graph is globally rigid, finitely determinable when the graph is rigid, and has infinite ambiguities when the graph is not rigid.

Another closely related area is node localizability, exploring the locations of which nodes can be uniquely determined by the constraints of the distance graph when the entire network is not uniquely determinable. For a node, the key to determining its localizability is counting the disjoint paths from it to anchors [24], [27], [35]. Some preliminary knowledge about rigidity and localizability that will be later used in this article is listed in Appendix A to make this paper self-contained.

Overall, the rigidity and localizability studies are based on the count of edges. Richer edge information brings stronger rigidity and better localizability. Existing distance graph analysis mainly utilizes measured edges. When the edge between two nodes is not measured, is there any hint to know the edge property, and will the inferred edge property be useful? There exist piecemeal studies exploring the inference of unmeasured edges and discussing applications of these edges.

B. Information Inference in Distance Graph

In a distance graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, an edge $(i, j) \in \mathcal{E}$ is called *measurable* if $(i, j) \in \mathcal{E}$. Among the unmeasured edges $\notin \mathcal{E}$, there is a portion that has unique lengths because there are enough constraints to limit their lengths. This property has been noted in the literature. To characterize the unmeasured but length-unique edges, Jackson et al. [29], [30] defined *globally linked pairs of vertices*. But it is not a generic property, meaning that it will be affected by specific realization. Next, we review other edge inference studies in detail.

1) *Implicit Edge and RR-Cut Edge*: Yang et al. defined that an unmeasured edge (i, j) is *implicit* if the distances between i and j are the same in all realizations of a graph \mathcal{G} [24]. A sufficient condition has been proposed to characterize implicit edges. In a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with two subgraphs $\mathcal{G}_1 = \{\mathcal{V}_1, \mathcal{E}_1\}$ and $\mathcal{G}_2 = \{\mathcal{V}_2, \mathcal{E}_2\}$, where $(\mathcal{E}_1, \mathcal{E}_2)$ is a partition of \mathcal{E} , if any two nodes $i, j \in \mathcal{V}_1 \cap \mathcal{V}_2$ both belong a rigid component in \mathcal{G}_1 and a rigid component in \mathcal{G}_2 , (i, j) is implicit if $(i, j) \notin \mathcal{E}$. This condition incurs a combinational number of graph partitions, which is computationally inefficient. Thus it was further proposed that in a redundantly rigid but not 3-connected component, if i and j form a binary vertex cut, (i, j) is an implicit edge. We denote such an edge as “RR-Cut

edge”. But a redundantly rigid component usually has few vertex cuts since it is a well-connected structure. Thus, the number of “RR-Cut edges” is very few.

2) *Shadow Edge*: Another main idea to leverage the *Unit Disk Graph* (UDG) constraint, i.e., $d_{ij} \leq R$ if $(i, j) \in \mathcal{E}$ and $d_{ij} > R$ if $(i, j) \notin \mathcal{E}$. Oliva et al. [31] proposed that if a node i has only two neighbors with known locations in the trilateration process, it has two possible locations $\hat{\mathbf{p}}_i^1$ and $\hat{\mathbf{p}}_i^2$. Suppose another node k with known location $\hat{\mathbf{p}}_k$. If $\|\hat{\mathbf{p}}_i^1 - \hat{\mathbf{p}}_k\|_2$ (or $\|\hat{\mathbf{p}}_i^2 - \hat{\mathbf{p}}_k\|_2$) is less than R , $\hat{\mathbf{p}}_i^2$ (or $\hat{\mathbf{p}}_i^1$) is the only feasible location. Then, the distance between i and k is unique, and (i, k) is called a *shadow edge*.

To detect shadow edges, a Shadow Edge Localization Algorithm (SELA) was designed. SELA needs at least three initially localized nodes with known locations. Then, each free node i iteratively checks its neighbors. If i is connected to 3 localized nodes, it marks itself as localized. If i connects 2 localized nodes, it checks whether it has a shadow edge to any localized node. SELA needs a node to know the locations of all localized nodes to check the shadow edges.

3) *Avoiding Flip Ambiguities (AFA) Edge*: Instead of assuming a constant ranging scope R , Guo et al. [32] defined the maximum communication distance, within which i and j can certainly communicate with each other.

$$D_{ij} = \min(\max(\max(D_i^1) - \varepsilon), \max(\max(D_j^1) - \varepsilon)), \quad (1)$$

where D_i^1 is the set of one-hop distances. Then, for any three mutually-connected localized nodes $\{a, b, c\}$, an x-a-y coordinate system is defined. The two possible locations $\hat{\mathbf{p}}_i^1$ and $\hat{\mathbf{p}}_i^2$ of node i are calculated in this coordinate system using the distance measurements. If $\|\hat{\mathbf{p}}_i^1 - \hat{\mathbf{p}}_b\|_2$ (or $\|\hat{\mathbf{p}}_i^2 - \hat{\mathbf{p}}_b\|_2$) is less than D_{ib} , $\hat{\mathbf{p}}_i^2$ (or $\hat{\mathbf{p}}_i^1$) is the only feasible location. Then, $\hat{\mathbf{p}}_i$ is uniquely determined. The edge (i, b) is an edge *avoiding flip ambiguities (AFA)*. An Avoiding Flip Ambiguities Localization Algorithm (AFALA) was further designed to simultaneously avoid flip ambiguities and calculate node locations.

Both SELA and AFALA use UDG constraints, and they are incremental algorithms. A shadow edge or an AFA edge can be determined only when the localization process finishes. They cannot judge the unique length property of a desired edge $(i, j) \notin \mathcal{E}$ directly. But this is actually needed in practice.

4) *Negative Edge*: In our previous work [12], we inferred the length of the unmeasured edge in a special structure generally seen in a node’s neighborhood. The graph with four vertices and five edges, i.e., a K_4 graph with only one edge missing, is defined as a *basic flipping graph (BFG)*. The missing edge has been proved to have and only have two possible lengths d_{ib}^+ and d_{ib}^- . Then, if d_{ib}^- is less than R , d_{ib}^+ is inferred to be the only feasible length, which is called a *negative edge*. To find negative edges, a node only needs to detect BFG in its neighborhood and check the two possible lengths. Finding negative edges does not require node locations, and decisions can be made using only local information. The major defect is that only two hop unmeasurable edges can be inferred.

But we should note that these existing inference methods are still piecemeal and non-uniform. In particular, the number of inferable implicit edges is very limited. The shadow edges

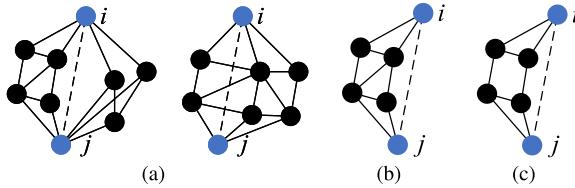


Fig. 1. The dashed edge in (a) is rangeable; in (b) is rigid; in (c) is flexible.

and AFA edges are inferred only in the trilateration process. The negative edges are inferred only in \mathcal{BFG} . In this paper, we will present the general properties and general methods for inferring the hidden knowledge from the unmeasured edges.

III. PROBLEM FORMULATION AND CENTRALIZED RANGEABLE EDGE DETECTION

This section formulates the problem and uniformly defines the unmeasured edge properties. The centralized condition and algorithm to identify rangeable edges are presented.

A. Problem Formulation

Let the terminology “node” denote various application-dependent network entities. A distance graph can be represented by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{d}\}$, where \mathcal{V} is the set of nodes, \mathcal{E} is the set of edges, and \mathbf{d} is the set of inter-node distances measured by ranging techniques. For an arbitrary node $v_i \in \mathcal{V}$, if it can measure the distance to another node $v_j \in \mathcal{V}$ by a ranging technique, there is a measurement $d_{ij} \in \mathbf{d}$, and hence an edge $(i, j) \in \mathcal{E}$. Let $\dim \in \{2, 3\}$ be the dimension. A realization of \mathcal{G} is a map $\mathcal{V} \rightarrow \mathbb{R}^{\dim}$ that the realized node coordinates satisfy all the measured inter-node distance constraints. The following content is based on $\dim = 2$, whose main idea can be similarly extended to $\dim = 3$.

Given $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{d}\}$, an edge (i, j) is measurable if $(i, j) \in \mathcal{E}$. In this paper, the hidden knowledge in unmeasured edges $\notin \mathcal{E}$ is mined. From aforementioned different terminologies of unique length unmeasured edges, there still lacks a unified definition of the properties of unmeasured edges. Thus, regarding the possibilities of edge lengths, this paper classifies the unmeasured edges into three categories.

Definition 1 (Properties of Unmeasured Edges): In a distance graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, let d_{ij} be the length of edge (i, j) . An unmeasured edge $(i, j) \notin \mathcal{E}$ is:

- **Rangeable**, if d_{ij} is unique in any realization of \mathcal{G} .
- **Rigid**, if d_{ij} has a finite number of possible lengths in different realizations of \mathcal{G} .
- **Flexible**, if d_{ij} has infinite number of possible lengths.

Fig. 1 gives typical examples of different kinds of unmeasured edges in \mathbb{R}^2 , where the edges plotted as solid lines are measured edges. Fig. 1(a) shows two examples of rangeable edges. The first one is because the left component and the right component are both rigid. Each of them has a discrete possible realizations. Because these two components are generically rigid, we will prove it has zero measure that the two rigid graphs in another realization reach a consensus length of (i, j) which is different from the ground truth. The second one is because (i, j) is in a globally rigid component.

In Fig. 1(b), the length of (i, j) has limited possibilities, since i, j are in a rigid graph with a limited number of realizations; In Fig. 1(c), the length of (i, j) can change continuously as the non-rigid graph deforms continuously. Intuitively, the most important unmeasured edges are the rangeable edges. Since their lengths are unique, knowing these constraint information will improve the accuracy of localizability analysis. For example, in the first subgraph of Fig. 1(a), let the three black nodes in the right part be the anchor nodes. The trilateration method [20] detects v_j as localizable since it is connected to three anchors and v_i as non-localizable since it is connected to only two anchors. But once the rangeable edge (i, j) is added, node v_i becomes localizable so that the localizability detection result is improved. Thus, we investigate identifying rangeable unmeasured edges. The centralized exploration is firstly conducted.

B. Centralized Rangeable Edge Detection

Note that the edge properties defined in Definition 1 are analogous to graph rigidity properties. From the definition, if an unmeasured edge (i, j) is inside a rigid component, its possible length is finite, so it is at least a rigid edge. Similarly, if an unmeasured edge (i, j) is inside a globally rigid component, its length is unique and (i, j) is rangeable. But this condition is not necessary. For example, the left graph in Fig. 1(a) is not globally rigid, but (i, j) is rangeable.

Theorem 1 (Centralized Condition for Edge Rangeability): For a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, if an unmeasured edge $(i, j) \notin \mathcal{E}$ is inside a redundantly rigid component \mathcal{C} of \mathcal{G} , and there are at least three vertex-disjoint paths from i to j inside \mathcal{C} , (i, j) is rangeable.

Proof: If \mathcal{C} is 3-connected, it yields that \mathcal{C} is globally rigid and then it is trivial that (i, j) is rangeable; If \mathcal{C} is not 3-connected, there exist two vertices $\{u, v\}$ whose removal will disconnect \mathcal{C} to several components so that $\mathcal{C} = \bigcup_i \mathcal{C}_i$ and any two components $\{\mathcal{C}_i, \mathcal{C}_j\}$ share two common vertices $\{u, v\}$.

Without loss of generality, let \mathcal{P}_1 , \mathcal{P}_2 , and \mathcal{P}_3 denote the three vertex-disjoint paths from i to j inside \mathcal{C} . Denote the last vertex connecting j on each path as b_1 , b_2 and b_3 , respectively. It can be concluded that:

- 1) i has three vertex-disjoint paths to $B = \{b_1, b_2, b_3\}$.
- 2) The vertices $b_1 \neq b_2 \neq b_3$ since \mathcal{P}_1 , \mathcal{P}_2 , and \mathcal{P}_3 are vertex-disjoint.

It has been revealed that the following two are equivalent [23]:

- a) A vertex i belongs to the redundantly rigid component of B in which i has three vertex-disjoint paths to three distinct vertices of B .
- b) The vertex i belongs to a globally rigid subgraph of \mathcal{G}^I that contains at least three vertices in B , where \mathcal{G}^I is the extended graph obtained by edge replacement [23].

Based on 1) and 2), a) can be satisfied, which is equivalent to b). Without loss of generality, let $\bar{\mathcal{G}}$ be the globally rigid subgraph in b). Considering that j has three distinct edges connecting B , it is trivial that $\bar{\mathcal{G}} \cup j$ is also globally rigid. Since $\bar{\mathcal{G}}$ contains i , (i, j) is rangeable. \square

A centralized algorithm to *Detect Rangeable Edges (DRE)* is designed based on Theorem 1. Algorithm 1 shows the DRE routine, which firstly decomposes the graph into redundant rigid components with Pebble Game [17]. Then 3-connected components in the redundantly rigid components are detected by SPQR-tree [36], [37]. The components that satisfy both 3-connection and redundantly rigidity are globally rigid. In globally rigid components, all unmeasured edges are rangeable. For an unmeasured edge (i, j) in a redundantly rigid component but not in a globally rigid component, we check whether i has three vertex-disjoint paths to j in the redundant component by a maximum flow algorithm [27]. If YES, (i, j) is rangeable.

Algorithm 1 Detect Rangeable Edge (DRE)

```

Input:  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 
Output:  $E^R = \{(i, j) | (i, j) \notin \mathcal{E} \wedge (i, j) \text{ is rangeable}\}$ 
Initialize:  $E^R \leftarrow \emptyset$ ;
1  $\{\mathcal{C}_1, \dots, \mathcal{C}_k\} \leftarrow$  decompose  $\mathcal{G}$  into redundantly rigid
   components by Pebble Game [17];
2 foreach  $\mathcal{C} \in \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$  do
3   find 3-connected components  $\{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_m\}$  by
      SPQR-Tree [36], [37];
4   add all unmeasured edges in  $\mathcal{G}_1$  to  $\mathcal{G}_m$  into  $E^R$ ;
5   foreach  $\{i, j\} \in \mathcal{C}$ ,  $(i, j) \notin \mathcal{E}$ , and
        $\{i, j\} \notin$  any 3-connected component do
6     check the number of disjoint paths between  $i$ 
       and  $j$  by maximum flow [27];
7     if  $i$  and  $j$  have 3 disjoint paths then
8       add  $(i, j)$  to  $E^R$ ;
9 return  $E^R$ ;

```

C. Time Complexity of DRE

In DRE, the graph is firstly decomposed into redundantly rigid components by the Pebble Game algorithm, which has a worst-case performance of $O(n^2)$ for a network of n nodes [17]. Then, for finding 3-connected structures, the SPQR-Tree algorithm requires $O(\bar{n} + \bar{m})$ time for each component, where \bar{n} and \bar{m} are the average number of nodes and edges in a component [37]. The maximum flow algorithm to count disjoint paths requires a complexity of $O(\bar{n}^2(\Delta^3 + \bar{m}))$, where Δ is the upper bound of neighbor numbers, i.e., every node has no more than Δ neighbors [27].

Later we will show the effectiveness of DRE algorithm. But detecting rangeable edges distributively has more application significance, because many localization algorithms [13], [38] and localizability detection algorithms such as TP [20], WE [22], TE [26] are all distributed. In practical applications, a node can hardly know the global information of the network. So it is necessary to detect the rangeable edges distributively.

IV. DISTRIBUTED RANGEABLE EDGE DETECTION

Considering an arbitrary node i , we further propose a distributed condition and a distributed algorithm to infer rangeable edges from i to all other unmeasured nodes in \mathcal{G} .

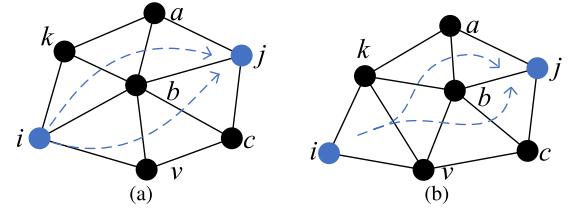


Fig. 2. (a) There are two disjoint rigid branches from i to j . (b) The two rigid branches from i to v are not disjoint.

A. Distributed Rangeability Theory

We first give some concepts that will enable us to derive the condition for finding rangeable edges in a distributed sense. The first is a structure called minimally rigid branch (MRB). The second is a special kind of MRB combination: Disjoint MRB (DMRB). The distributed rangeability condition will then be proposed based on DMRB.

1) MRB and DMRB Concepts:

Definition 2 (Minimally Rigid Branch (MRB) Between i and j): Suppose $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a minimally rigid graph containing i and j , $(i, j) \notin \mathcal{E}$. If the removal of any vertex v (except i, j) along with its connected edges from \mathcal{G} makes the remaining graph non-rigid, \mathcal{G} is an MRB between i and j , denoted by $\mathbf{B}(i, j)$.

An example of MRB can be referred to Fig. 1(b) without considering the dashed line. The graph will become non-rigid if any vertex is removed. We analyze the property of MRB that will be later used in the DMRB lookup.

Lemma 1 (Node Degree Property of MRB): In \mathbb{R}^2 , if $\mathbf{B}(i, j)$ is an MRB, then for any vertex $v \neq i, j$ in $\mathbf{B}(i, j)$, node degree $d(v) \geq 3$, and $d(i) \geq 2, d(j) \geq 2$.

Lemma 2 (MRB $+ (i, j)$ is an M-Circuit): If $\mathbf{B}(i, j)$ is an MRB between i and j , then the graph composed by $\mathbf{B}(i, j) + (i, j)$ is an M-Circuit.

An M-Circuit characterizes the minimal redundant rigid graph (formally defined in Appendix A). See Appendix B and Appendix C for the proofs of Lemma 1 and Lemma 2, respectively. Then, the concept of DMRB is defined based on *binary vertex cut set*. A binary vertex cut set in a graph is a pair of vertices whose removal will disconnect the graph.

Definition 3: (Disjoint Minimally Rigid Branches (DMRB) between i and j): For two MRBs $\mathbf{B}_1(i, j)$ and $\mathbf{B}_2(i, j)$, if $\mathbf{B}_1(i, j) \cup \mathbf{B}_2(i, j)$ don't have any binary vertex cut set, they are called disjoint minimally rigid branches between i and j .

In Fig. 2(a), there are two MRBs from i to j . The MRB $\mathbf{B}_1(i, j)$ is composed by $\{i, k, a, b, j\}$ and the MRB $\mathbf{B}_2(i, j)$ is composed by $\{i, v, b, c, j\}$. Since there exist no binary vertex cut set whose removal can disconnect $\mathbf{B}_1(i, j) \cup \mathbf{B}_2(i, j)$, the two branches are disjoint. In Fig. 2(b), the two MRBs are $\{i, k, v, a, b, j\}$ and $\{i, k, v, b, c, j\}$, respectively. Their union has a binary cut set $\{k, v\}$, whose removal will disconnect $\mathbf{B}_1(i, j) \cup \mathbf{B}_2(i, j)$, so the two branches are not disjoint.

2) Distributed Rangeable Edge Theorem: The theorem to characterize rangeability edges in a distributed sense is proposed based on DMRB.

Theorem 2 (Distributed Condition for Edge Rangeability): For an unmeasurable edge (i, j) , if i and j are constrained in at least two DMRBs, (i, j) is rangeable.

Before proving the theorem, we show the properties of (i, j) when there are two DMRBs between i and j .

Lemma 3 (Three Vertex Disjoint Paths): If there are two DMRBs $\mathbf{B}_1(i, j)$ and $\mathbf{B}_2(i, j)$ between i and j , then there are at least three vertex disjoint paths between i and j .

The proof of Lemma 3 is in Appendix D. Consider there are two DMRBs between i and j , denoted by $\mathbf{B}_1(i, j)$ and $\mathbf{B}_2(i, j)$. We consider the union graph of $\mathbf{B}_1(i, j)$ and $\mathbf{B}_2(i, j)$, denoted by \mathcal{M} . From Lemma 2, $\mathbf{B}_1(i, j) + (i, j)$ and $\mathbf{B}_2(i, j) + (i, j)$ are both M -circuits. The merging of $\mathcal{M} = \mathbf{B}_1(i, j) \cup \mathbf{B}_2(i, j)$ can be considered as merging two M -circuits and deleting the common edge (i, j) .

Lemma 4: If $\mathbf{B}_1(i, j)$ and $\mathbf{B}_2(i, j)$ are DMRBs between i and j , $\mathbf{B}_1(i, j) \cup \mathbf{B}_2(i, j)$ is a redundantly rigid graph.

The proof of Lemma 4 is in Appendix E. The intuition to prove Theorem 2 is as follows. Since $\{i, j\}$ are constrained in a redundantly rigid component, there is no flex-type ambiguity [12]. The length of (i, j) has multiple realizations only when flipping ambiguities happen. Because $\mathbf{B}_1(i, j)$ and $\mathbf{B}_2(i, j)$ are disjoint branches without common flipping axis, in generic graphs, two disjoint branches have zero measure to generate another consensus length for (i, j) instead of the ground truth.

Proof of Theorem 2: Let \mathcal{R} be the set of all realizations of $\mathcal{M} = \mathbf{B}_1(i, j) \cup \mathbf{B}_2(i, j)$. Let $d_r(i, j)$ denote the Euclidean distance between the two vertices i and j in a specific realization $r \in \mathcal{R}$. Let $D_{\mathcal{M}}(i, j) = \cup_{r \in \mathcal{R}} d_r(i, j)$ be the set of possible lengths of $d_r(i, j)$ in all possible realizations of \mathcal{M} . Since \mathcal{M} is rigid, $D_{\mathcal{M}}$ is finite. Let d be the ground truth distance between i and j , thus $d \in D_{\mathcal{M}}$. We show that d is the only element of the set $D_{\mathcal{M}}$. Because \mathbf{B}_1 and \mathbf{B}_2 are both rigid, $D_{\mathbf{B}_1}$ and $D_{\mathbf{B}_2}$ are both finite. In MRB, the ambiguities of $d(i, j)$ are caused by cut sets. Since \mathbf{B}_1 and \mathbf{B}_2 are disjoint and share no common cut set, the values in $D_{\mathbf{B}_1}$ and $D_{\mathbf{B}_2}$ are independent and $D_{\mathbf{B}_1}$ and $D_{\mathbf{B}_2}$ have measure zero. Hence, every element in $D_{\mathbf{B}_1} \setminus d$ has zero probability to appear in $D_{\mathbf{B}_2} \setminus d$. Thus it has zero probability that there is another consensus length for $d_r(i, j)$ instead of d . \square

B. Distributed DMRB Construction

Theorem 2 indicates that the key to identifying rangeable edges is to find DMRBs between a pair of nodes. We revisit a Triangle Extension (TE) operation, which allows easy and distributed minimally rigid structure construction. Then, we design the construction of DMRB based on TE.

1) Triangle Extension:

Definition 4 (Triangle Extension: TE [26]): Given $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, TE starts from a root node i and one of its neighbor v which initializes a minimally rigid K_2 graph, denoted by T . TE is a series of extensions from T . In each extension, a node u is added into $\mathcal{V}(T)$ and two edges $(g_1, u), (g_2, u)$ are added into $\mathcal{E}(T)$ if $g_1, g_2 \in \mathcal{V}(T)$ and $(g_1, u), (g_2, u) \in \mathcal{E}$.

If a node u is extended from two nodes g_1 and g_2 in T , g_1 and g_2 are said to be u 's parents; u is a child of g_1 and

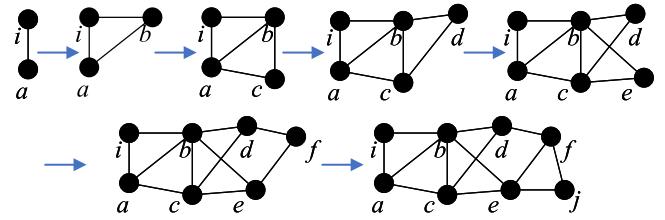


Fig. 3. An example of triangle extension from (i, a) to j .

g_2 ; The ancestors of u are recursively defined as u 's parents, i.e., $\{g_1, g_2\}$ and ancestors of $\{g_1, g_2\}$.

Definition 5 (TE-branch): A TE-branch is constructed by TE from a node i to a node j , which is denoted by $\mathbf{B}(i, j)$. It includes j , the ancestors of j , and the edges between them.

Fig. 3 shows an example of TE from a K_2 graph until to a node j . Note that there may exist multiple TE-branches between i and j , since the root nodes may be $\{i$, an arbitrary neighbor of $i\}$, and each step of extension may proceed in a different direction.

2) How to Ensure MRBs are Disjoint?: From Definitions 3 and 5, the TE-branches without common binary vertex cut set are exactly DMRBs. Thus, if any binary cut is not repeatedly used when constructing TE-branches from i to j , the extended branches can constitute DMRBs between i and j . To avoid repeated binary cuts, the binary vertex cut sets in an arbitrary branch $\mathbf{B}(i, v)$ are desired, which can be found through the properties of TE-branch.

In TE, already-extended nodes will not be extended again, so the extension is one-way, and every node is only added once in a TE-branch. This gives an important inner structure of a TE-branch:

Proposition 1 (Inner Structure of a TE-branch): If a branch $\mathbf{B}(i, j)$ is obtained by TE, $\mathbf{B}(i, j)$ has following properties:

- 1) Every node (except the end node) has a child.
- 2) Every node (except the two root nodes) has two parents.
- 3) A TE-branch is a 2-connected graph.

With the above structure, binary vertex cut sets in a TE-branch can be found by the following lemma.

Lemma 5 (Binary Vertex Cut Sets in a TE-branch): In a branch $\mathbf{B}(i, v)$, each parent pair of an extended node forms a binary vertex cut set of the branch (except the two roots), and only a pair of parents can form a binary vertex cut set.

Proof: In a TE-branch, since the extension is one-way, every node connects with its ancestors only through its parents. After removing the two parents, the node has no path to the ancestors, so the graph is disconnected. Therefore, each pair of parents of an extended node form a binary vertex cut set.

Then, we prove only a pair of parents can form a binary vertex cut set. Suppose two nodes $\{u, v\}$ are removed, which are not a pair of parents of any node. From Proposition 1, each node except the end node has a child, so either u or v has a child. The child and the child's children still have a path to its early ancestors through the child's another parent instead of u and v . Other later extended nodes whose ancestors are not u, v are not impacted by the removal of u, v . So the removal

of $\{u, v\}$ cannot disconnect the TE-branch if they are not a pair of parents. \square

Thus, binary vertex cut sets in a branch can be detected by parent pairs of all extended nodes.

C. Distributed Rangeable Edge Detection Algorithm

Based on Theorem 2 and Lemma 5, we are ready to design a distributed rangeable edge detection algorithm. If any pair of nodes is not repeatedly used for extension, MRBs can be constructed to be disjoint. Avoiding duplicate cut sets is implemented by logging the used cut sets. The BB algorithm starts from an arbitrary root node i . It finds disjoint rigid branches from i to all other nodes in the network. Each other node j maintains a **list of binary vertex cut sets (called Blacklist)** for the branches from i to j , denoted by $L(i, j)$. It also maintains $c(i, j)$, i.e., the number of disjoint branches that can reach j from i , initialized as $c(i, j) = 0$.

TE will be rooted from i and one of its neighbors (say k) and tries to extend other nodes by TE. If j can be extended by an MRB B_1 rooted from i and k , j updates $L(i, j)$ to the binary vertex cut set in $B_1(i, j)$ and set $c(i, j) = 1$. The branch then continuously extends other nodes, until no node can be extended. The extended graph of this round is denoted by $G_{i,k}$, in which $\{i, k\}$ are the roots of the extension. Binary vertex cut sets in $G_{i,k}$ will be detected, denoted by $L_{G_{i,k}}$ and each extended node j updates its blacklist to $L(i, j) \cap L_{G_{i,k}}$, i.e., keeping only binary vertex cut sets in the extended graph. This is because the later rigid extension will help to eliminate some binary vertex cut sets in the branch from i to j .

The root then starts the next round of branching. Suppose in the new round an MRB (say B_2) rooted from i is trying to extend j . Node j will check whether the new branch contains a binary vertex cut set in its blacklist $L(i, j)$. We denote this operation as *CheckBL*.

- 1) If YES, node j will refuse to be extended in this new branch.
- 2) If NO, node j will be extended by this new branch. j increases $c(i, j) = c(i, j) + 1$, and sets its blacklist to the union of its blacklists and $B_2(i, j)$ blacklist.

The blacklists of extended nodes only keep the binary vertex cut sets in the extended graph after each round of extension. The TE-branching from i ends when no branching can be started from the root i , i.e., all neighbors refuse to be extended according to their blacklists.

The detailed implementation of BB is given in Algorithm 2. Each node v (exclude the root i) is initialized from *O-state* (*Open*), $L(i, v) = \emptyset$, $c(i, v) = 0$. The root selects a *O-state* neighbor to start TE. Every TE-extended node v updates its blacklist $L(i, v)$ and $c(i, v)$. When this TE cannot extend anymore, the root starts another TE using the same neighbor. When the root and the neighbor cannot extend any new child, the neighbor is marked in *C-state* (*Close*).

Then each node except the root i updates its blacklist according to the binary cut set in the extended graph, i.e., $L(i, v) = L(i, v) \cap L_{G_{i,k}}$. The root selects another *O-state*

Algorithm 2 Branching and Blacklisting (BB)

```

Input: root node  $i$ ; neighbors of  $i$ :  $N[i]$ ;
Output: set of nodes with rangeable edges to  $i$ ;
Initialize:  $S(v) \leftarrow O\text{-state}$ ,  $\forall v$ ;  $MRB(i, v) \leftarrow \emptyset$ ,  $c(i, v) \leftarrow 0$ ,  $L(i, v) = \emptyset$ ,  $GL \leftarrow \emptyset$ ,  $\forall (i, v) \notin \mathcal{E}$ ;
while ( $\exists k \in N[i]$  is in O-state) do
     $i$  selects  $k$  to start a TE-branch  $B$ ;
    if  $B$  reaches  $v$  then
        if CheckBL( $L(i, v)$ ,  $B$ ) returns NO then
             $c(i, v) \leftarrow c(i, v) + 1$ ;
             $MRB(i, v) \leftarrow MRB$  from  $i$  to  $v$  in  $B$ ;
             $L(i, v) \leftarrow$  binary vertex cut sets of  $MRB(i, v)$ ;
            continues  $B$  extension;
        else
             $B$  tries to extend another node ;
            if no other node can be extended by  $B$  then
                 $G_{i,k} \leftarrow$  the extended graph from  $i$ ,  $k$ ;
                 $GL \leftarrow$  binary vertex cut sets of  $G_{i,k}$ ;
                every node  $v \in B$  except  $i, k$  updates
                 $L(i, v) \leftarrow L(i, v) \cap GL$ ;
                mark  $k$  in C-state;
    return  $\{v | c(i, v) \geq 2\}$ ;

```

neighbor to start TE. When all the neighbors of i change to *C-state*, the BB algorithm rooted at i ends. All the nodes return their predictable index $c(i, v)$. The unmeasured edge (i, v) is *rangeable* if $c(i, v) \geq 2$, *rigid* if $c(i, v) = 1$, and *flexible* if $c(i, v) = 0$.

D. Time Complexity Analysis of BB

The first round of TE will cover all the n nodes at most and generate $n - 2$ binary vertex cut sets. Its time complexity is $O(n)$ because it doesn't need to do the blacklisting operation in extending each node. But in the later rounds of TE extension, each node checks its blacklist before being extended. In the k th round of TE extension, in the worst case, the blacklisting operation needs to compare $n - 2$ binary vertex cut sets (in the branch) with another $k(n - 2)$ binary vertex cut sets (stored at the node). Each cut set is a pair of nodes. We encode each pair (u, v) ($u \leq v$ without loss of generality) by a number $u \times \log_{10}^n + v$ so that finding a common pair can be checked by number equivalence.

There are two ways to further speed up the *CheckBL*($L(i, v)$, B_k) operation. At first, a hashtable can be used to store the blacklist of binary vertex sets at each node. Since the time cost of searching in a hash table is $O(1)$, finding whether the binary vertex cut set in the current branch is in a node's blacklist needs $O(n)$ time cost. But the hashtable needs much more storage cost. The second method is to store the "numbers" at each node in a sorted array. Then binary search is conducted which takes $O(n \log n)$ time cost in the blacklisting check process in the worst case. So the worst case complexity of BB is $O(n^2 \log n)$.

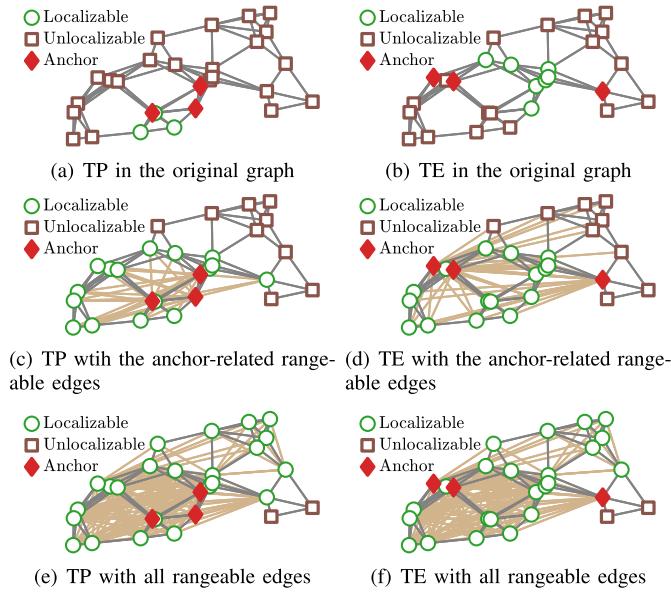


Fig. 4. Visualizing the localizable nodes detected by TP and TE.

V. APPLICATIONS OF HIDDEN KNOWLEDGE

The utility of rangeable, rigid, and flexible edges is explored through their applications. Specifically, inferred rangeable edges offer three valuable applications. (1) They enhance the detection of localizable nodes, surpassing existing methodologies in accuracy and reliability. (2) They facilitate the identification of StrONGLy Localizable Areas (SOLAs) within sparse networks, pinpointing regions with high localization potential. (3) They augment the network with richer distance information, improving spatial awareness. Moreover, rigid edges serve to relax the trilateration-based localizability condition. Flexible edges shed light on the network's structural nuances, revealing insights into the distribution and connectivity patterns.

A. Enhancing Node Localizability With Rangeable Edges

Node localizability is an important fundamental problem [13], [23], [24]. Existing distributed node localizability detection algorithms, e.g., Trilateration Protocol (TP) [20] and distributed Triangle Extension (TE) [26], only utilize the measured edges, so many localizable nodes cannot be discovered. Each inferred rangeable edge can work as an edge length constraint to restrict the freedom of the node locations. We add the inferred rangeable edges into the original graph to improve the node localizability detection accuracy.

1) *Visualizing Localizable Node Improvement:* Without loss of generality, we visualize the results with the example graph in Fig. 4. The black lines and brown lines represent the original edges and inferred rangeable edges, respectively. The green circles represent localizable nodes detected by a certain algorithm and brown squares show the non-localizable nodes.

In Fig. 4(a), TP [20] only detects 3 localizable nodes in the original graph. In Fig. 4(c), the anchors run BB and add 31 anchor-related rangeable edges. TP detects 16 localizable

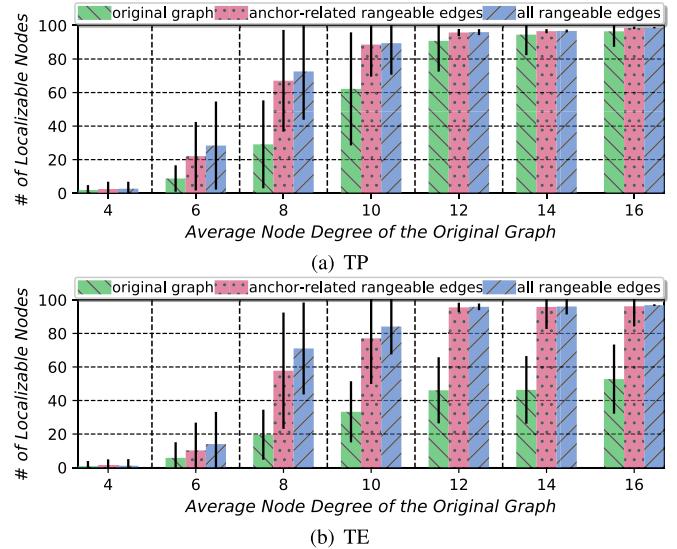


Fig. 5. The number of localizable nodes in the original graph and after adding the anchor-related rangeable edges or all rangeable edges.

nodes. In Fig. 4(e), every node runs BB and 127 rangeable edges are inferred. TP detects 23 localizable nodes, marking a notable enhancement compared to the original graph.

Similar results can be obtained when TE [26] is adopted. In Fig. 4(b), TE detects 7 localizable nodes. In Fig. 4(d), anchors conduct BB and infer 43 rangeable edges. Then TE can find 16 localizable nodes. In Fig. 4(f), each node conducts BB and 127 rangeable edges are found. TE finds 23 localizable nodes. Note that all nodes except the two at the right-bottom form a global rigid component, so they are actually localizable. The results show the inferred rangeable edges help both TP and TE remarkably improve their localizable node detection capability.

2) *Statistical Results:* The statistical results on the improvement of node localizability detection accuracy are investigated in Fig. 5. The network scale is set to 100 nodes with 3 anchors. The network density is assessed by average node degree (*AvgDeg*), which can be controlled by varying the ranging scope *R*. Under each *AvgDeg*, 1,000 random networks are generated. In sparse networks where *AvgDeg* is around 6 to 10, much more localizable nodes can be found when each node conducts BB and uses the rangeable edge information. When the network becomes denser, even if only the anchors run BB, the increased number of localizable nodes is close to the results when all nodes run BB. The reason is that distributed localizable node detection algorithms generally start from anchors and conduct extension operations to find localizable nodes incrementally. Even the anchor-based BB extension can greatly improve the node localizability detection performance.

B. Mining Strongly Localizable Area With Rangeable Edges

For network localization in unevenly distributed networks, even if the localization result of the whole network is not satisfactory, there are still some dense components whose local realization results are accurate [14], [34]. We define the concept of *StrONGLy Localizable Area (SOLA)* to characterize such

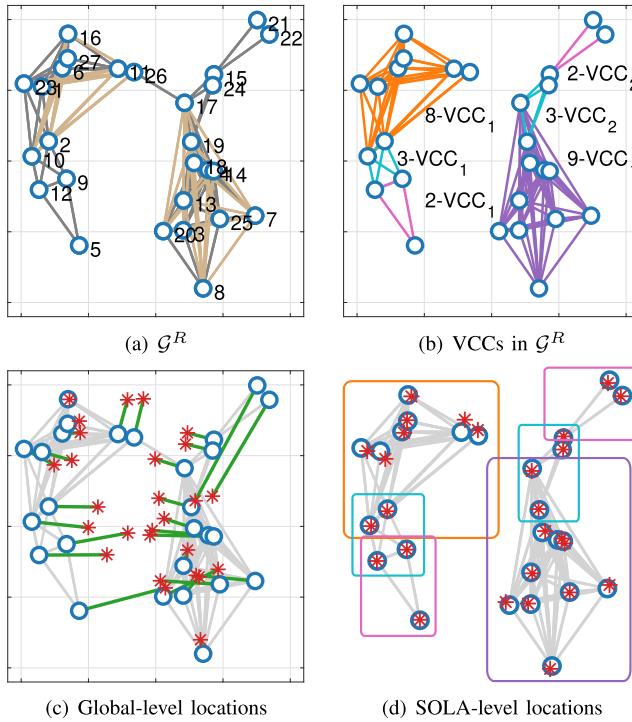


Fig. 6. Effectiveness of SOLA.

substructures. The aim is to calculate the local relative node locations. The local location results can be transformed into global locations with synchronization methods [12]. SOLA is not strictly defined. It refers to the subgraph with satisfactory connectivity and therefore has the potential of good local realization accuracy. The inferred rangeable edges enable us to easily detect SOLA structures.

1) *Visualizing Effectiveness of SOLA*: We discuss detecting SOLA using *k*-vertex connected component (*k*-VCC) [39] as an example, which is a widely-applied cohesive subgraph metric that can guarantee good local edge density. A *k*-VCC of \mathcal{G} is a connected subgraph in which the removal of any $k - 1$ vertices will not disconnect the subgraph. Let \mathcal{G}^R denote the graph \mathcal{G} along with its rangeable edges. \mathcal{G}^R can be decomposed into a sequence of VCCs $\{1\text{-VCC}_1, \dots, 1\text{-VCC}_{n_1}, \dots, k\text{-VCC}_1, \dots, k\text{-VCC}_{n_k}\}$ in a polynomial time [39], where n_k is the number of *k*-VCCs.

Fig. 6(a) shows \mathcal{G}^R of a network with 30 nodes, where the original edges and rangeable edges are in black lines and brown lines, respectively. In Fig. 6(b), \mathcal{G}^R is decomposed into 6 VCCs. Then, locations of all nodes are calculated with g^2o [40] algorithm using inter-node distances. We call them global-level locations. In Fig. 6(c), the calculated locations are denoted by red asterisks, and the true locations are in blue circles. The green line connecting the true location and the calculated location indicates the localization error of a certain node. In Fig. 6(d), the locations are calculated in the respective local coordinate system of each VCC, so we call them SOLA-level locations. Comparing the localization errors in Fig. 6(c) and Fig. 6(d), SOLA has the potential for much lower localization errors. Through the rangeable edges, we can get a clearer picture about which subgraphs contain

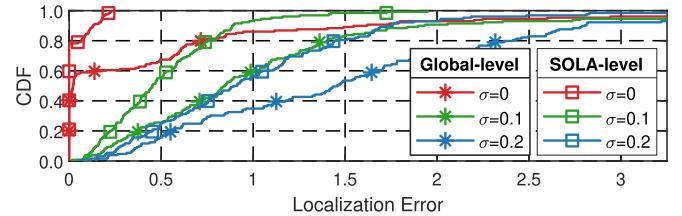


Fig. 7. Global-level localization errors and SOLA-level localization errors.

more distance constraints and thus find better localizable subgraphs.

2) *Statistical Results*: In Fig. 7, the distance noise setting varies in $\{0, 0.1, 0.2\}$. 100,000 records are collected for both global-level and SOLA-level localization errors in each setting. The cumulative distribution function (CDF) of the localization error is given. It is shown that the localization error at the SOLA-level is significantly lower than that at the global-level.

C. Enriching Distance Information With Rangeable Edges

A scheme to calculate their specific lengths is further investigated to enrich graph distance information.

1) *Calculating the Length of Rangeable Edge*: Given a distance graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{d}\}$, the node locations $\{\hat{\mathbf{p}}_1, \dots, \hat{\mathbf{p}}_n\}$ can be calculated using the g^2o algorithm [40]. Then, for each unmeasured edge $(i, j) \notin \mathcal{E}$, an estimated length \hat{d}_{ij} can be calculated as $\hat{d}_{ij} = \|\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j\|_2$.

Considering that not all estimates are reliable, along with the calculation, we define a concept of *stability index* to evaluate the reasonableness of a distance. In specific, the *stability index* is assigned as $|d_{ij} - \hat{d}_{ij}|$ for an edge (i, j) , where d_{ij} is the distance between i and j in the ground truth network topology. The stability index indicates the difference from the estimated distance to the ground truth distance. Due to the property as in Definition 1, the edge length of a rangeable edge should be unique in any realization of the graph. Thus, its stability index is expected to be minor. Next, we verify this property.

2) *Statistical Results*: A sparse network setting ($AvgDeg = 6$) and a dense setting ($AvgDeg = 4$) are considered for statistical results. Under each setting, 100,000 records are collected for each kind of edge, and the CDF of the stability index is plotted in Fig. 8. The measurable edges, i.e., $(i, j) \in \mathcal{E}$, have the smallest stability index because they are used as explicit constraints. For the unmeasurable edges, the rangeable edges have a similar stability index with the measurable edges, especially in dense networks. This verifies the property of the rangeable edges. The rigid unmeasured edges and flexible unmeasured edges, are less stable. The detected rigid edges are more stable than the flexible edges because they have a discrete number of possible realizations. The verification shows that even if the rangeable unmeasured edges are not utilized in the graph optimization model, their lengths remain to be consistent with the ground truth in the graphs' realization results. So we can use the edge connectivity information as if the rangeable unmeasured edges exist. If their length information is needed, we can use the lengths estimated by g^2o confidently.

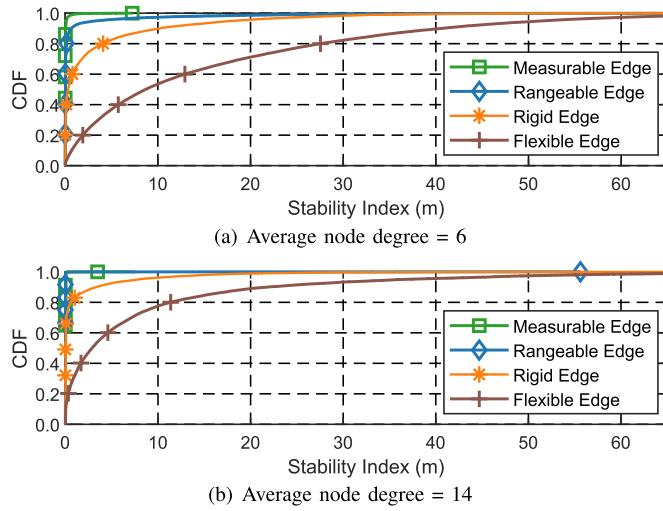


Fig. 8. Distribution of edge stability index.



Fig. 9. Finding localizable node with rigid edge.

Other potential applications of rangeable edges include indicating the 3D relative relations among UAVs; 3D protein structure determination with nuclear magnetic resonance distance [41], [42], singularity analysis [43], chemical molecular conformation [44], etc.

D. Applications of Rigid and Flexible Edges

Localizability is based on edge-counting. Rangeable edges can increase the edge count so that more localizable nodes can be found. It has been proposed that a node is localizable if it is connected to 3 distinct anchors by three measurable edges in \mathbb{R}^2 [20]. We relax this condition with rigid edges.

Theorem 3: In a graph \mathcal{G} , a node i is localizable if it is connected to three distinct anchors by two measurable edges and one rigid edge, where the two measurable edges are not included in the MRB of the rigid edge.

Proof: Let a , b , and c be three distinct anchors. Without loss of generality, suppose that (i, a) and (i, b) are measurable, and (i, c) is rigid. Let $\mathbf{D}_{ic} = \{d_{ic}^1 \dots d_{ic}^k\}$ be the possible lengths of (i, c) , whose elements are determined by the MRB between i and c . Let d_{ic} be the distance in the ground truth realization of \mathcal{G} , $d_{ic} \in \mathbf{D}_{ic}$. Then, we prove that d_{ic} is the only element in \mathbf{D}_{ic} . Since d_{ia} , d_{ib} , d_{ab} , d_{bc} , and d_{ac} are available, as in Fig. 9, two possible lengths of (i, c) can be calculated with our previous method [12], denoted by $\hat{\mathbf{D}}_{ic} = \{d_{ic}^+, d_{ic}^-\}$. It has been proved that $\hat{\mathbf{D}}_{ic}$ covers all the possible lengths of (i, c) [12], $d_{ic} \in \hat{\mathbf{D}}_{ic}$. Since the measurable edges are not included in the MRB between i and c , the values in $\mathbf{D}_{ic} \setminus d_{ic}$ and $\hat{\mathbf{D}}_{ic} \setminus d_{ic}$ are independent in the distance space where \mathbf{D}_{ic} and $\hat{\mathbf{D}}_{ic}$ have measure zero. Therefore, any element in $\mathbf{D}_{ic} \setminus d_{ic}$ has zero probability to appear in $\hat{\mathbf{D}}_{ic} \setminus d_{ic}$. It has

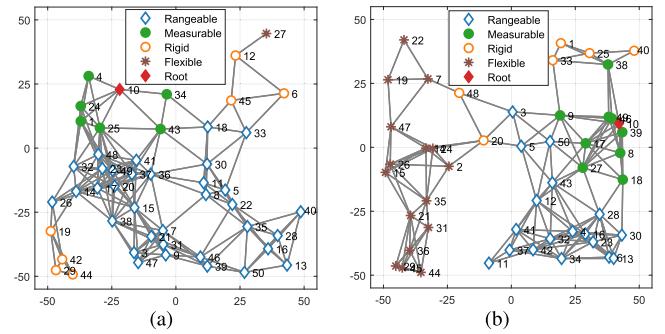


Fig. 10. A visualization of the rigid and rangeable edge inference of BB.

zero probability that there is another consensus distance for (i, c) instead of d_{ic} . Based on the uniqueness of d_{ia} , d_{ib} , and d_{ic} , it can be concluded that i is localizable. \square

The flexible edges cannot provide deterministic or discrete distance constraints like the rangeable edges or rigid edges. But they can help to understand the network distribution. For example, the set of flexible edges indicates the weakness of the network. Such knowledge provides hints on network attack/defense operations or node deployment optimization.

VI. PERFORMANCE EVALUATION

Extensive evaluations are conducted to show the validity and effectiveness of the proposed methods. An overall comparison with existing methods is summarized.

A. Visualizing the Rigid and Rangeable Edges

Fig. 10(a) and Fig. 10(b) visualize the rigid and rangeable edges detected by the BB algorithm. For clarity, only the edges rooted from a single root node are plotted. The root node is represented by a red diamond. The edge between the root and a solid circle is a measurable edge. The edges between the root node and a hollow diamond, a hollow circle, and an asterisk are rangeable, rigid, and flexible, respectively. It is shown that edges with different properties are essentially separated by binary vertex cuts to the root. In Fig. 10(b), there exists a global binary vertex cut due to an obstacle, making all the nodes in the left part of the network can not find disjoint MRB to the root. Then, all the edges between the root node and the left part of the network are flexible.

B. Statistical Results on Edge Proportion

The proportion distribution of different types of edges is evaluated. Node's sensing radius, the number of nodes, and the topology seeds are controlled so that the random networks are generated with different typologies and sparsity. In each network, a random root node i is selected and the properties of its edges to all other nodes are identified by DRE. In Fig. 11, the average node degree ($AvgDeg$) varies from 4 to 16. For each $AvgDeg$, 1,000 networks are randomly generated. Since the lengths of measurable edges and rangeable edges are unique, their proportions are stacked. The number of rangeable edges from i to other nodes improves quickly with the increase of $AvgDeg$. Rigid edges take more than 10% proportion when

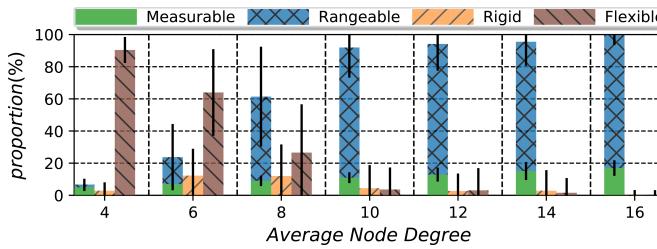


Fig. 11. Distribution of different types of edges w.r.t. single root node.

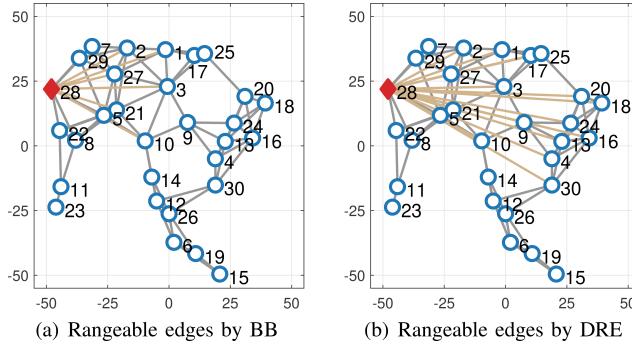


Fig. 12. The rangeable edges detected by the distributed algorithm BB and the centralized algorithm DRE.

AvgDeg is 6 and 8. Rangeable edges take the most proportion when *AvgDeg* is beyond 8. It is shown that rich hidden knowledge exists in generic graphs with different sparsity, so it is crucial to utilize these hidden knowledge for graph structure analysis, e.g., localizable node detection.

C. Distributed Rangeable Edge Detection v.s. Centralized Rangeable Edge Detection

The BB algorithm utilizes triangle extension for distributed construction of minimal rigid branches, so some rangeable edges may not be detected by BB. The case when the distributed BB algorithm finds less rangeable edges than the centralized DRE algorithm is illustrated in Fig. 12. For clarity, only the rangeable edges rooted at node 28 are given. Fig. 12(a) shows rangeable edges detected by BB. Because nodes 9 and 20 cannot be extended from either node 3 or node 25, making all the MRBs share a common vertex cut (9, 10), BB fails to detect rangeable edges beyond (9, 10).

Fig. 12(b) shows rangeable edges detected by DRE. Although the triangle extension is restricted by (9, 10), the three vertex-disjoint path property is not affected, enabling DRE can find rangeable edges crossing (9, 10). Thus, the defect of distributed rangeable edge detection is mainly caused by the limitation of triangle extension for finding MRBs. But later extensive experiments will show that the number of rangeable edges detected by BB is very close to DRE, showing the effectiveness of the distributed algorithm in practice.

D. Rangeable Edges v.s. Other Existing Methods

The number of edges inferred by DRE and BB are evaluated together and compared with existing methods, including RR-cut edge [24], Avoiding Flip Ambiguities (AFA)

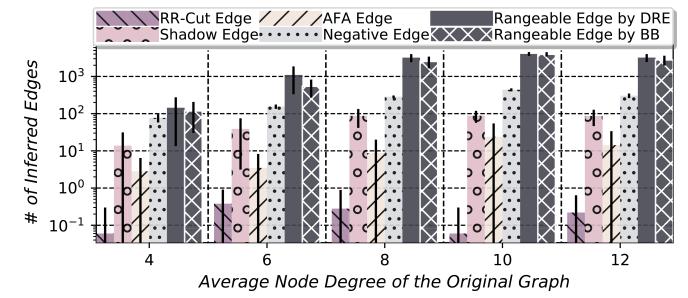


Fig. 13. The number of edges inferred by different algorithms.

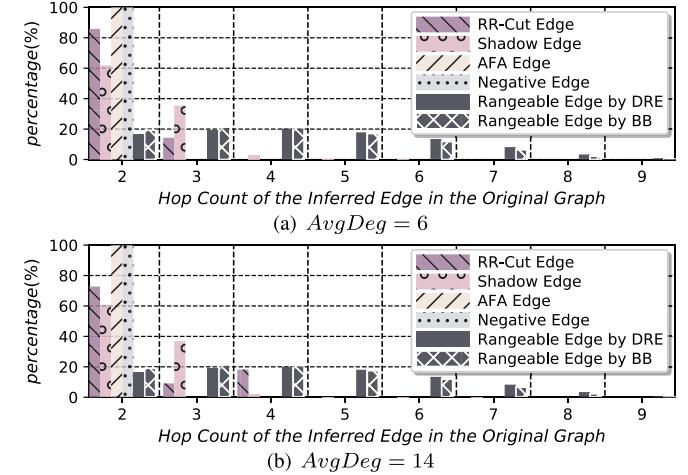


Fig. 14. The hop counts of the inferred edges in the original graph.

edge [32], shadow edge [31], and negative edge [12]. The *AvgDeg* varies from 4 to 12 to control the edge densities. 1,000 random networks are generated for each setting. The number of different kinds of inferred edges are presented in Fig. 13.

From Fig. 13, DRE and BB can infer much more unique-length edges than other methods. Note that the vertical axis is in a exponential form. The number of rangeable edges detected by BB is less than DRE due to the limitation of triangle extension. Nevertheless, both BB and DRE find several orders of magnitude more edges than other algorithms. The advantage is especially prominent in sparse networks since the proposed algorithms are based on branch extension, which works satisfactorily even the network is sparse. The number of inferred edges increases with *AvgDeg* when *AvgDeg* $<= 10$. Because more disjoint minimal rigid branches can be found in denser networks. When *AvgDeg* > 10 , fewer edges can be detected, since most of the edges are already measurable.

E. On the Diversity of the Inferred Edges

Apart from the advantage of detecting more unique-length edges, we further evaluate the difference in “diversity”. In specific, for an AFA edge [32] or a negative edge [12] (i, j) , the hop count between i and j in the original graph is always 2, due to the principle restriction. In contrast, the hop count of other inferred edges is more diverse. In Fig. 14, two *AvgDeg* are presented, and 1,000 networks are generated under each *AvgDeg* and the hop counts of the inferred edges are counted.

TABLE I
THE COMPARISON OF EDGE INFERENCE METHODS

	Hop Count	Anchor Information	Distance Information	UDG	Edge Inference Without Location Information	Category
RR-Cut Edge [24]	≥ 2	not required	not required	\times	support	centralized
Shadow Edge [31]	≥ 2	required	required	\checkmark	not support	centralized
AFA Edge [32]	≥ 2	not required	required	\checkmark	not support	centralized
Negative Edge[12]	2	not required	required	\checkmark	support	distributed
Rangeable Edge by DRE (Ours)	≥ 2	not required	not required	\times	support	centralized
Rangeable Edge by BB (Ours)	≥ 2	not required	not required	\times	support	distributed

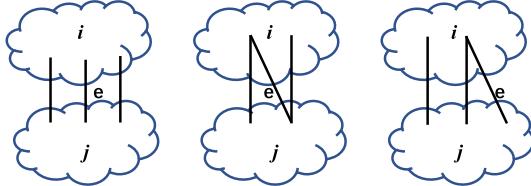


Fig. 15. Non-rigid virtual edge, rigid virtual edge, and rangeable virtual edge.

It is shown that 100% AFA edge and negative edge are 2 hops in the original graph. The RR-Cut edge and the shadow edge mainly distribute between 2 hops and 4 hops. In contrast, the rangeable edges detected by DRE and BB vary from 2 hops to 9 hops, which is close to the maximum number of hops of the original graph. Thus, the rangeable edges are not only much more numerous, but also much diverse.

F. Comparison With Existing Methods

Table I summarizes existing methods for inferring unmeasured edges in distance graphs. As introduced in Section II-B, the shadow edge, AFA edge, and negative edge methods mainly utilize the UDG constraint to infer the unmeasured edges. They all require the distance information of the edges, so there are inevitably wrongly inferred edges due to the ranging noise. Shadow edge and AFA edge don't support edge inference without knowing location information and they are centralized methods. Negative edge supports edge inference without knowing location information, and it is fully distributed. Shadow edge requires anchor information, but AFA edge and negative edge don't require anchor information. Among the three methods, negative edge method can directly give the length of the inferred edge during inference, while the other two (shadow edge and AFA edge) can only calculate the edge length based on the calculated node locations.

The RR-Cut edge method is different. It does not use the UDG constraint but finds the vertex cuts in the redundantly rigid components. It doesn't need the edge length information or the anchor information and supports designated edge inference. It is also a centralized method.

Compared with these existing methods, DRE and BB don't use UDG constraints; don't need anchor information, or edge length information. They support multi-hop and edge inference without location information.

VII. CONCLUSION

This paper investigates the properties of unmeasurable edges in generic graphs, i.e., rangeability, rigidity, and flexibility.

The set of rangeable edges is identified since these edges can provide deterministic knowledge as if they are measured. Centralized and distributed conditions to detect rangeable edges are proposed, as well as the centralized algorithm DRE and the distributed algorithm BB. BB can also output rigid and flexible edges. We show that the detected edges are valuable in various applications. Extensive simulation results show that rich hidden knowledge can be discovered by the proposed methods. DRE and BB can detect much more rangeable edges than existing methods. In future work, we will further investigate the value of the rigid and flexible edges.

APPENDIX A PRELIMINARIES ON RIGIDITY AND LOCALIZABILITY

Different levels of rigidity theory are introduced, as well as localizability theory and algorithms.

A. Basic Concepts in Graph Rigidity

In \mathbb{R}^2 , Laman [16] gave the concept of *minimally rigid graph* (also called Laman graph).

Definition 6 (Minimally Rigid Graph/Laman graph): In \mathbb{R}^2 , a graph \mathcal{G} with n vertices is minimally rigid if \mathcal{G} has $2n - 3$ edges and any k vertex sub-graph has at most $2k - 3$ edges.

A Laman graph will become non-rigid after the removal of any edge. A graph is rigid if it has a spanning Laman graph. Redundant rigidity is a higher level of rigidity.

Definition 7 (Redundant Rigid Graph [15]): A graph G is redundantly rigid if it remains rigid after removing any edge.

M-Circuit characterizes the minimal redundant rigid graph.

Definition 8 (*M*-Circuit [45]): A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in \mathbb{R}^2 is an *M*-circuit if $|\mathcal{E}| = 2|\mathcal{V}| - 2$ and $|\mathcal{E}(\mathcal{X})| \leq 2|\mathcal{X}| - 3$ for all $\mathcal{X} \subset \mathcal{V}$ with $2 \leq |\mathcal{X}| \leq |\mathcal{V}| - 1$; or equivalently, $|\mathcal{E}| = 2|\mathcal{V}| - 2$ and $\mathcal{G} - e$ is minimally rigid for any $e \in \mathcal{E}$.

Beyond redundant rigidity, global rigidity considers the unique recovery of a graph from the given set of distance measurements. A globally rigid graph can be fully reconstructed from the set of edge lengths in \mathbf{d} , which is called the *realization* of the graph. In \mathbb{R}^2 , the globally rigid graph is characterized by edge counting-based methods [15], [33].

Definition 9 (Globally Rigid [15], [33]): A graph \mathcal{G} is globally rigid in \mathbb{R}^2 if and only if \mathcal{G} is 3-connected and redundantly rigid, or \mathcal{G} is a complete graph with at most three vertices.

Overall, rigidity properties at different levels are mainly defined on graphs by counting edges, characterizing whether a graph has a discrete or unique number of realizations.

B. Basic Concepts in Localizability

Given a distance graph, *network localizability* returns a yes-or-no answer to describe whether the entire network can be uniquely localized. *Node localizability* is more practical and can find out which nodes can be uniquely localized. On the theoretical aspect, a network is localizable if and only if it is globally rigid and has at least three nonlinear anchors. To find localizable nodes, Yang et al. [24] gave a sufficient condition.

Lemma 6: (Sufficient condition for node localizability): RR3P [24]: A node is localizable in \mathbb{R}^2 if it is in a redundant rigid component C with at least three non-collinear anchors, and it has three vertex-disjoint paths to three anchors in C .

On the algorithmic aspect, some methods can find localizable nodes in a distributed manner. The most classic approach is based on trilateration, such as the Triangle Protocol (TP) [20]. A trilateration graph in \mathbb{R}^2 is a graph with a trilateration ordering, $\pi = (u_1, u_2, \dots, u_n)$ where u_1, u_2, u_3 are three anchors forming a K_3 and every $u_i (i > 3)$ is adjacent to at least three nodes before u_i in π . But not all localizable networks admit trilateration ordering. An improvement to TP is Wheel Extension (WE) [21], [22]. A wheel graph is formed by a cycle and a vertex adjacent to all vertices on the cycle. If a wheel graph contains three anchors, all the nodes extended by the wheel extension are localizable. Goldenberg et al. [46] proposed a SWEEP localization algorithm based on bilateration extension. It can localize more localizable nodes than TP. Wu et al. [26] proposed Triangle Extension (TE), which further improved TP and WE. TE starts from two anchor nodes (i, j) to conduct bilateration extension. If the extension reaches another anchor node, then all the nodes in this extended graph are localizable. These extension-based localizability detection and localization algorithms use only the knowledge of the measured edges.

APPENDIX B PROOF OF LEMMA 1

Proof: The Lemma can be proved by contradiction. If $d(v) = 1$ for any v , $\mathbf{B}(i, j)$ is not rigid, which is contradicted by $\mathbf{B}(i, j)$ being minimally rigid, so all nodes in an MRB have degree ≥ 2 ; If $d(v) = 2$ for $v \neq i, j$, if v and the two edges connecting to v are removed, edges among other nodes will not be impacted. The remained graph has $n - 1$ vertices and $2n - 3 - 2 = 2(n - 1) - 3$ edges. Any subgraph with k vertices has no more than $2k - 3$ edges, so the remaining graph is still minimally rigid, which contradicts that $\mathbf{B}(i, j)$ is an MRB. \square

APPENDIX C PROOF OF LEMMA 2

Proof: Let $\mathcal{G}(\mathcal{V}, \mathcal{E}) = \mathbf{B}(i, j) + (i, j)$. Since $\mathbf{B}(i, j)$ is a MRB, it is minimally globally rigid and have $2|\mathcal{V}| - 3$ edges, so $\mathbf{B}(i, j) + (i, j)$ has $2|\mathcal{V}| - 2$ edges. From the definition of M-Circuit in Definition 8, we prove that for all $\mathcal{X} \subset \mathcal{V}$ with $2 \leq |\mathcal{X}| \leq |\mathcal{V}| - 1$, $|\mathcal{E}(\mathcal{X})| \leq 2|\mathcal{X}| - 3$.

For any subgraph $G(\mathcal{X})$ with $2 \leq |\mathcal{X}| \leq |\mathcal{V}| - 1$, because any node in a MRB except the end nodes i, j has at least two parents and one child, it has node degree ≥ 3 . Then the removal of any vertex except i, j removes at least three edges;

the removal of i or j also removes at least three edges from \mathcal{G} , since i, j both have degree ≥ 2 . After $k = |\mathcal{V}| - |\mathcal{X}|$ vertices are removed, suppose the subgraph induced by the removed vertices is denoted by \mathcal{G}' . If there are s connected components in \mathcal{G}' , the number of vertices in each connected component is denoted by k_1, k_2, \dots, k_s . Then in a connected component with k_i vertices there is at least $k_i - 1$ edges for being connected and at most $2k_i - 3$ edges for being minimally rigid. So the total number of removed edges with the k vertices are at least $3k - \sum_{i=1}^s (2k_i - 3) + \sum_{i=1}^s (k_i - 1) \geq 2k + 2$, because $\sum_{i=1}^s k_i \leq k$. So the remained number of edges is at most $2|\mathcal{V}| - 2 - (2k + 2) \leq 2|\mathcal{X}| - 3$. \square

APPENDIX D PROOF OF LEMMA 3

Proof: Because $\mathbf{B}_1(i, j)$ and $\mathbf{B}_2(i, j)$ don't share a common vertex cut set, the path from i to j cannot be cut by removing any two vertices. So there are at least three vertex disjoint paths between i and j . \square

APPENDIX E PROOF OF LEMMA 4

Proof: Let $\mathcal{M} = (\mathcal{V}, \mathcal{E}) = \mathbf{B}_1(i, j) \cup \mathbf{B}_2(i, j)$, which is merged by identifying common edges and vertices. It is clear that $i, j \in \mathcal{V}$ but $(i, j) \notin \mathcal{E}$. Lemma 3.6.16 in [45] gives the following fact for a rigid graph: When $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $i, j \in \mathcal{V}$ but $(i, j) \notin \mathcal{E}$, then i, j must be in a redundantly rigid component if and only if i, j is in a rigid component of $\mathcal{G} - e$ for any $e \in \mathcal{E}$.

Let's firstly consider to remove any edge e from $\mathcal{M} = \mathbf{B}_1(i, j) \cup \mathbf{B}_2(i, j)$. If e is in $\mathcal{M} \setminus \mathbf{B}_1(i, j)$, after removing e , there is still a rigid graph $\mathbf{B}_1(i, j)$ constraining $\{i, j\}$. The same holds for $e \in \mathcal{M} \setminus \mathbf{B}_2(i, j)$. So we only need to consider to remove any edge e from $\mathbf{B}_1(i, j) \cap \mathbf{B}_2(i, j)$. From Lemma 3, the graph is at least 2-connected after removing any shared edge. Note that a 2-connected graph is flexible only when it has a 2-edge-cut, i.e., the removal of the two edges disconnects the graph. But if there is a 2-edge-cut in $\mathcal{M} - e$, it must because $\mathcal{M} - e$ has at least two binary vertex cut sets. In such case, Fig. 15 shows the three possible relationships between e and the 2-edge-cut. The first case appears only when \mathbf{B}_1 and \mathbf{B}_2 both contain the three edges. For being minimally rigid, $\mathbf{B}_1 \cup \mathbf{B}_2$ must contain a binary cut set, which is contradict to the property that \mathbf{B}_1 and \mathbf{B}_2 are disjoint. The second and the third cases are contradict with that \mathcal{M} doesn't have a binary vertex cut set. So i, j must be within a rigid component after removing any e from \mathcal{M} . From Lemma 3.6.16 in [45], i, j is constrained in a redundantly rigid component. \square

REFERENCES

- [1] C. Ma, B. Wu, S. Poslad, and D. Selviah, "Wi-Fi RTT ranging performance characterization and positioning system design," *IEEE Trans. Mobile Comput.*, vol. 21, no. 2, pp. 740–756, Feb. 2022.
- [2] FiRa Consortium. (2023). *Localization Cases Based on Distance Measurement Using UWB Technology*. [Online]. Available: <https://www.firaconsortium.org/discover/use-cases#location>
- [3] Q. Liu, X. Bai, X. Gan, and S. Yang, "LoRa RTT ranging characterization and indoor positioning system," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–10, Mar. 2021.

- [4] Y. Schröder and L. Wolf, "InPhase: Phase-based ranging and localization," *ACM Trans. Sensor Netw.*, vol. 18, no. 2, pp. 1–39, May 2022.
- [5] N. Semiconductors. (2020). *Uwb for Social Distancing: Take the Guesswork Out of Staying Safe*. [Online]. Available: <https://www.nxp.com/company/blog/uwb-for-social-distancing-take-the-guesswork-out-of-staying-safe>: BL-UWB-SOCIAL-DISTANCING
- [6] Wikipedia. (2024). *List of Uwb-enabled Mobile Devices*. [Online]. Available: https://en.wikipedia.org/wiki/List_of_UWB-enabled_mobile_devices
- [7] S. Wang, Y. Wang, X. Bai, and D. Li, "Communication efficient, distributed relative state estimation in UAV networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 1151–1166, Apr. 2023.
- [8] X. Fang, X. Li, H. Huang, and L. Xie, "Multisensor-based distributed localization in multi-agent systems," in *Proc. IEEE 17th Int. Conf. Control Autom. (ICCA)*, Jun. 2022, pp. 449–453.
- [9] S. Wang, Y. Wang, D. Li, and Q. Zhao, "Distributed relative localization algorithms for multi-robot networks: A survey," *Sensors*, vol. 23, no. 5, p. 2399, Feb. 2023.
- [10] H. Huang, W. Miao, G. Min, C. Huang, X. Zhang, and C. Wang, "Resilient range-based d-dimensional localization for mobile sensor networks," *IEEE/ACM Trans. Netw.*, vol. 28, no. 5, pp. 2037–2050, Oct. 2020.
- [11] S. Ji, K.-F. Sze, Z. Zhou, A. M. So, and Y. Ye, "Beyond convex relaxation: A polynomial-time non-convex optimization approach to network localization," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 2499–2507.
- [12] H. Ping, Y. Wang, D. Li, and T. Sun, "Flipping free conditions and their application in sparse network localization," *IEEE Trans. Mobile Comput.*, vol. 21, no. 3, pp. 986–1003, Mar. 2022.
- [13] H. Ping, Y. Wang, D. Li, and W. Chen, "Understanding node localizability in barycentric linear localization," *IEEE/ACM Trans. Netw.*, vol. 31, no. 3, pp. 1353–1368, Jun. 2023.
- [14] T. Sun, Y. Wang, D. Li, Z. Gu, and J. Xu, "WCS: Weighted component stitching for sparse network localization," *IEEE/ACM Trans. Netw.*, vol. 26, no. 5, pp. 2242–2253, Oct. 2018.
- [15] B. Jackson and T. Jordán, "Connected rigidity matroids and unique realizations of graphs," *J. Combinat. Theory B*, vol. 94, no. 1, pp. 1–29, May 2005.
- [16] G. Laman, "On graphs and rigidity of plane skeletal structures," *J. Eng. Math.*, vol. 4, no. 4, pp. 331–340, Oct. 1970.
- [17] D. J. Jacobs and B. Hendrickson, "An algorithm for two-dimensional rigidity percolation: The pebble game," *J. Comput. Phys.*, vol. 137, no. 2, pp. 346–365, Nov. 1997.
- [18] X. Bai, Y. Wang, H. Ping, X. Xu, D. Li, and S. Wang, "InferLoc: Hypothesis-based joint edge inference and localization in sparse sensor networks," *ACM Trans. Sensor Netw.*, vol. 20, no. 1, pp. 1–28, Jan. 2024.
- [19] L. Mo et al., "Canopy closure estimates with GreenOrbs: Sustainable sensing in the forest," in *Proc. 7th ACM Conf. Embedded Networked Sensor Syst.*, Nov. 2009, pp. 99–112.
- [20] T. Eren et al., "Rigidity, computation, and randomization in network localization," in *Proc. IEEE INFOCOM*, Mar. 2004, pp. 2673–2684.
- [21] Z. Yang, Y. Liu, and X.-Y. Li, "Beyond trilateration: On the localizability of wireless ad-hoc networks," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 2392–2400.
- [22] Z. Yang, Y. Liu, and X.-Y. Li, "Beyond trilateration: On the localizability of wireless ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 6, pp. 1806–1814, Dec. 2010.
- [23] Z. Yang and Y. Liu, "Understanding node localizability of wireless ad-hoc networks," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.
- [24] Z. Yang and Y. Liu, "Understanding node localizability of wireless ad hoc and sensor networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 8, pp. 1249–1260, Aug. 2012.
- [25] Y. Zhang, Q. Wei, Y. Wang, H. Ping, and D. Li, "GPART: Partitioning maximal redundant rigid and maximal global rigid components in generic distance graphs," *ACM Trans. Sensor Netw.*, vol. 19, no. 4, pp. 1–26, Jun. 2023.
- [26] H. Wu, A. Ding, W. Liu, L. Li, and Z. Yang, "Triangle extension: Efficient localizability detection in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7419–7431, Nov. 2017.
- [27] H. Ping, Y. Wang, X. Shen, D. Li, and W. Chen, "On node localizability identification in barycentric linear localization," *ACM Trans. Sensor Netw.*, vol. 19, no. 1, pp. 1–26, Feb. 2023.
- [28] Y. Liu, Z. Yang, X. Wang, and L. Jian, "Location, localization, and localizability," *J. Comput. Sci. Technol.*, vol. 25, no. 2, pp. 274–297, Mar. 2010.
- [29] B. Jackson, T. Jordán, and Z. Szabadka, "Globally linked pairs of vertices in equivalent realizations of graphs," *Discrete Comput. Geometry*, vol. 35, no. 3, pp. 493–512, Mar. 2006.
- [30] B. Jackson, T. Jordán, and Z. Szabadka, "Globally linked pairs of vertices in rigid frameworks," in *Rigidity and Symmetry*. New York, NY, USA: Springer, 2014, pp. 177–203.
- [31] G. Oliva, S. Panzieri, F. Pascucci, and R. Setola, "Sensor networks localization: Extending trilateration via shadow edges," *IEEE Trans. Autom. Control*, vol. 60, no. 10, pp. 2752–2755, Oct. 2015.
- [32] Q. Guo, Y. Zhang, J. Lloret, B. Kantarci, and W. K. G. Seah, "A localization method avoiding flip ambiguities for micro-UAVs with bounded distance measurement errors," *IEEE Trans. Mobile Comput.*, vol. 18, no. 8, pp. 1718–1730, Aug. 2019.
- [33] R. Connelly, "Generic global rigidity," *Discrete Comput. Geometry*, vol. 33, no. 4, pp. 549–563, Apr. 2005.
- [34] Y. Wang, T. Sun, G. Rao, and D. Li, "Formation tracking in sparse airborne networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2000–2014, Sep. 2018.
- [35] Y. Zhang, S. Liu, X. Zhao, and Z. Jia, "Theoretic analysis of unique localization for wireless sensor networks," *Ad Hoc Netw.*, vol. 10, no. 3, pp. 623–634, May 2012.
- [36] C. Gutwenger and P. Mutzel, "A linear time implementation of SPQR-trees," in *Graph Drawing* (Lecture Notes in Computer Science), vol. 1984. Berlin, Germany: Springer, 2001, pp. 77–90.
- [37] J. E. Hopcroft and R. E. Tarjan, "Dividing a graph into triconnected components," *SIAM J. Comput.*, vol. 2, no. 3, pp. 135–158, Sep. 1973.
- [38] X. Fang, X. Li, and L. Xie, "3-D distributed localization with mixed local relative measurements," *IEEE Trans. Signal Process.*, vol. 68, pp. 5869–5881, 2020.
- [39] D. Wen, L. Qin, Y. Zhang, L. Chang, and L. Chen, "Enumerating k-Vertex connected components in large graphs," in *Proc. IEEE 35th Int. Conf. Data Eng. (ICDE)*, Apr. 2019, pp. 52–63.
- [40] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g²o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 3607–3613.
- [41] C. Lavor et al., "Minimal NMR distance information for rigidity of protein graphs," *Discrete Appl. Math.*, vol. 256, pp. 91–104, Mar. 2019.
- [42] R. Labiak, C. Lavor, and M. Souza, "Distance geometry and protein loop modeling," *J. Comput. Chem.*, vol. 43, no. 5, pp. 349–358, Feb. 2022.
- [43] J. M. Porta, N. Rojas, and F. Thomas, "Distance geometry in active structures," in *Mechatronics for Cultural Heritage and Civil Engineering*. Cham, Switzerland: Springer, 2018, pp. 115–136.
- [44] E. Estrada and D. Bonchev, *Chemical Graph Theory*. Boca Raton, FL, USA: CRC Press, 2013, pp. 1538–1558.
- [45] T. Jordán, "II-Combinatorial rigidity: Graphs and matroids in the theory of rigid frameworks," in *Discrete Geometric Analysis*, vol. 34. Budapest, Hungary: Egerváry Research Group, 2016, pp. 33–112.
- [46] J. Fang, M. Cao, A. S. Morse, and B. Anderson, "Localization of sensor networks using sweeps," in *Proc. 45th IEEE Conf. Decis. Control*, Dec. 2006, pp. 4645–4650.



Haodi Ping received the Ph.D. degree from the Department of Computer Sciences, Renmin University of China, in 2023. He visited the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 2022. He is currently an Assistant Professor with the College of Computer Science, Faculty of Information Technology, Beijing University of Technology. His research interests include network localization algorithms, graph optimization, and applications.



Yongcai Wang (Member, IEEE) received the B.S. and Ph.D. degrees from the Department of Automation Sciences and Engineering, Tsinghua University, in 2001 and 2006, respectively. He was an Associate Researcher with the NEC Laboratories China, from 2007 to 2009. He was a Research Scientist with the Institute for Interdisciplinary Information Sciences (IIIS), Tsinghua University, from 2009 to 2015. He was a Visiting Scholar with Cornell University in 2015. He is currently an Associate Professor with the Department of Computer Sciences, Renmin University of China. His research interests include the Internet of Things, spatial computing, spatial perception, and graph computing.



Deying Li (Member, IEEE) received the M.S. degree in mathematics from Huazhong Normal University in 1988 and the Ph.D. degree in computer science from the City University of Hong Kong in 2004. She is currently a Professor with the Department of Computer Science, Renmin University of China. Her research interests include wireless networks, mobile computing, and algorithm design and analysis.



Yu Zhang received the B.S. degree from the Department of Mathematics and Computer Science, Anhui Normal University, in 2016. She is currently pursuing the master's degree with the Department of Computer Sciences, Renmin University of China. Her research interests include network localization and applications.



Lihua Xie (Fellow, IEEE) received the B.E. and M.E. degrees in electrical engineering from the Nanjing University of Science and Technology in 1983 and 1986, respectively, and the Ph.D. degree in electrical engineering from the University of Newcastle, Australia, in 1992. Since 1992, he has been with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, where he is currently a Professor and the Director of the Delta-NTU Corporate Laboratory for Cyber-Physical Systems and the Center for Advanced Robotics Technology Innovation. He was the Head of the Division of Control and Instrumentation, Nanyang Technological University, from July 2011 to June 2014. His research interests include robust control and estimation, networked control systems, multi-agent control, and unmanned systems.

He is a Fellow of the Academy of Engineering Singapore, the International Federation of Automatic Control (IFAC), and the Chinese Automation Association (CAA). He is the Editor-in-Chief of *Unmanned Systems*. He has served as an Editor for IET book series in control; and an Associate Editor for a number of journals, including IEEE TRANSACTIONS ON AUTOMATIC CONTROL, *Automatica*, IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS.