# Universal Rigidity: Towards Accurate and Efficient Localization of Wireless Networks

Zhisu Zhu
ICME
Stanford University
Stanford, CA 94305
Email: zhuzhisu@stanford.edu

Anthony Man–Cho So
Dept. of SE&EM
The Chinese University of Hong Kong
Shatin, N. T., Hong Kong
Email: manchoso@se.cuhk.edu.hk

Yinyu Ye
Dept. of MS&E
Stanford University
Stanford, CA 94305
Email: yyye@stanford.edu

***Abstract*—A fundamental problem in wireless ad–hoc and sensor networks is that of determining the positions of nodes. Often, such a problem is complicated by the presence of nodes whose positions cannot be uniquely determined. Most existing work uses the notion of *global rigidity* from rigidity theory to address the non–uniqueness issue. However, such a notion is not entirely satisfactory, as it has been shown that even if a network localization instance is known to be globally rigid, the problem of determining the node positions is still intractable in general. In this paper, we propose to use the notion of *universal rigidity* to bridge such disconnect. Although the notion of universal rigidity is more restrictive than that of global rigidity, it captures a large class of networks and is much more relevant to the efficient solvability of the network localization problem. Specifically, we show that both the problem of *deciding* whether a given network localization instance is universally rigid and the problem of *determining* the node positions of a universally rigid instance can be solved efficiently using semidefinite programming (SDP). Then, we give various constructions of universally rigid instances. In particular, we show that trilateration graphs are generically universally rigid, thus demonstrating not only the richness of the class of universally rigid instances, but also the fact that trilateration graphs possess much stronger geometric properties than previously known. Finally, we apply our results to design a novel edge sparsification heuristic that can reduce the size of the input network while *provably* preserving its original localization properties. One of the applications of such heuristic is to speed up existing convex optimization–based localization algorithms. Simulation results show that our speedup approach compares very favorably with existing ones, both in terms of accuracy and computation time.**

## I. Introduction

Thanks in part to recent advances in technology and production capabilities, the deployment of large–scale wireless ad–hoc and sensor networks is becoming increasingly feasible and common. These networks can be used to perform a great variety of tasks, such as target tracking and detection, data collection and habitat monitoring, just to name a few. However, in order to fully realize the potential of such networks, it is essential that the positions of individual nodes can be determined with reasonable accuracy. Of course, this can be achieved in a straightforward manner by either manually configuring the nodes or equipping each node with a Global Positioning System (GPS). However, as is well–known (see, e.g., [25]), such a solution is often impractical or even infeasible. Therefore, the *network localization problem*—namely, the problem of determining the positions of nodes in a network—has attracted much research interest over the years (see, e.g., [21, Chapter 17] and the references therein). One of the most common settings under which the problem is studied is the so–called *fine–grained localization*. In that setting, it is assumed that all nodes can measure the distance between themselves and their neighboring nodes (this can be achieved using various hardware ranging techniques). Furthermore, some of the nodes are designated as *anchors*, which means that their positions are known (say, by manual configuration or through GPS). Hence, the network localization problem becomes that of determining the positions of non–anchor nodes so that they satisfy the given distance measurements (for simplicity, we shall refer to a non–anchor node as *sensor* in the sequel). Many solution approaches have been proposed for this problem; see, e.g., [3], [4], [6], [10], [22]–[25], [27], [31]. However, it should be noted that the problem as stated could be ill–posed, as there may be more than one non–congruent localization of the sensors satisfying the given distance measurements. Given the importance of accurate node positions for many applications, it is thus natural to ask whether a given instance of the network localization problem is uniquely localizable, and if so, whether that localization can be found efficiently.

In [11] (see also [1]), Eren et al. proposed to use tools from rigidity theory (see [16] for an introduction) to address these issues. In particular, they showed that for any $d \geq 1$, a generic network localization instance is uniquely localizable in $\mathbb{R}^d$ iff its associated *grounded graph* is generically globally rigid in $\mathbb{R}^d$. Such a connection allows one to utilize powerful results in rigidity theory to tackle the issue of unique network localizability. For instance, when combined with a recent result of Gortler et al. [15], one immediately obtains an efficient algorithm for checking whether a given network localization instance is uniquely localizable in $\mathbb{R}^d$ [1]. Moreover, it spurs a flurry of work on rigidity–based approaches for testing network localizability; see, e.g., [12], [14], [33]. It should be noted, however, that most of the latter approaches apply only to the two–dimensional case.

[1]The original localizability test described in [11] applies only to instances in $\mathbb{R}^2$, as it is based on a characterization of generically globally rigid graphs in $\mathbb{R}^2$ due to Hendrickson [17] and Jackson and Jordán [18]. However, its extension to higher dimensions is trivial in view of Gortler et al.'s result.

Although the approach proposed by Eren et al. [11] goes a long way towards resolving the issue of unique network localizability, a nagging question remains: Suppose one knows that a given network localization instance is uniquely localizable in $\mathbb{R}^d$ for some $d \geq 1$. Can one efficiently find the (unique) localization, or declare that the given instance is infeasible? Afterall, we are most interested in the positions of nodes in the network. Unfortunately, the notion of global rigidity seems ill–suited for answering this question. Indeed, unless $\mathcal{RP} = \mathcal{NP}$, there does not exist an efficient randomized algorithm that, given an instance of the network localization problem with a unique localization in $\mathbb{R}^2$, finds that localization [2]. In other words, the knowledge that an instance has a unique localization in $\mathbb{R}^d$ does not make the task of finding that localization any easier (see [1], [7], [26] for related results on the complexity of the network localization problem). Such a disconnect motivates the following question:

$(*)$ *Is there an alternative to the notion of global rigidity that admits both efficient* verification *and* realization*?*

In an attempt to answer this question, So and Ye [29] introduced the notion of *unique $d$–localizability* and showed that the problem of deciding whether a given network localization instance is uniquely $d$–localizable, as well as the problem of determining the node positions of a uniquely $d$–localizable instance, can be solved efficiently using semidefinite programming (SDP). However, unlike the notion of unique network localizability introduced by Eren et al. [11], which is defined in terms of global rigidity, the notion of unique $d$–localizability is defined directly in terms of the uniqueness of solution to a certain system of quadratic equations. Consequently, the connection between the results of [11] and that of [29] is not entirely obvious. Moreover, it is not clear how the notion of unique $d$–localizability fits into the framework of rigidity theory, and whether some of the rigidity–theoretic constructions can be used to study properties of uniquely $d$–localizable instances.

In this paper, we remedy the situation and provide further theoretical foundation for the notion of unique $d$–localizability. Specifically, we establish, for the first time, the precise conditions under which the notion of unique $d$–localizability and the rigidity–theoretic notion of universal rigidity are equivalent. Using this equivalence, we are able to resolve the question $(*)$ in the affirmative. In particular, even though universal rigidity is a more restrictive notion than global rigidity, it has much more favorable computational properties and is arguably more relevant to the efficient solvability of the network localization problem. Moreover, our equivalence result provides a connection between optimization theory and rigidity theory. This opens up the possibility of using tools from the former to tackle problems in the latter.

Now, given the nice algorithmic features of the notion of universal rigidity, it is natural to ask whether it captures a sufficiently rich class of graphs. We answer this question in the affirmative by showing that the trilateration graphs studied in [11] are in fact *generically universally rigid*. This sharpens the result in [11] and shows that trilateration graphs possess much stronger geometric properties than previously known. Moreover, when combined with our equivalence result, it demonstrates the existence of *sparse* (i.e. with only $O(|V|)$ edges) uniquely $d$–localizable instances for any fixed $d \geq 1$, thus refuting a common belief that such instances must be dense (e.g., with $\Omega(|V|^2)$ edges). As trilateration graphs and trilateration–based schemes are used in many localization algorithms, we believe that our result will have interesting implications on the performance of such algorithms.

In recent years, there has also been great interest in applying convex optimization techniques, particularly that of SDP relaxation, to tackle the network localization problem (see, e.g., [3], [4], [10], [20], [28]–[30], [32]). Although convex optimization–based localization algorithms can usually produce highly accurate results, they are also computationally demanding. One way to speed up such algorithms is to reduce the number of edges (and hence distance measurements) in the input network (i.e. to *sparsify* the network). Unfortunately, existing sparsification procedures (see, e.g., [20], [32]) do not preserve the localization properties of the input. It could happen, for example, that the sparsified instance is not uniquely $d$–localizable, even though the original instance is so. To further demonstrate the power of our results, we show how the generic universal rigidity of trilateration graphs can be used to circumvent this problem. Specifically, we propose a novel edge sparsification heuristic that can sparsify the input instance while provably preserving its localization properties. Such a heuristic is based on an efficient decomposition procedure that identifies trilateration subgraphs in the grounded graph associated with the input. Simulation results show that our speedup approach compares very favorably with existing ones, both in terms of accuracy and computation time.

The rest of the paper is organized as follows. In Section II, we formulate the network localization problem and briefly review the definition and properties of the notion of unique $d$–localizability. In Section III, we introduce the notion of universal rigidity and establish its relationship to that of unique $d$–localizability. In Section IV, we give various constructions of generically universally rigid graphs. In Section V, we apply our results to develop a novel edge sparsification heuristic and show how it can be used to speed up convex optimization–based localization algorithms without sacrificing their accuracy. In Section VI, we provide simulation results of our edge sparsification heuristic. Finally, we end with some closing remarks and future directions in Section VII.

## II. Problem Formulation and Background

We begin by giving a formal definition of the network localization problem. Let $d \geq 1$ be the dimension in which the nodes reside. (In practice, we usually have $d = 2$ or 3. However, our results apply to any dimension.) Let $G = (V, E)$ be the given network. Without loss of generality, we assume that $G$ is connected. The nodes of $G$ are partitioned into two sets: the set $V_s = \{1, \ldots, n\}$ of *sensors*, and the set $V_a = \{n+1, \ldots, n+m\}$ of *anchors*. Together, they induce

two subsets $E_{ss}, E_{sa} \subset E$, which are defined as follows:

$$\begin{aligned} E_{ss} &= \{(i,j) \in E : i,j \in V_s\} \\ E_{sa} &= \{(i,j) \in E : i \in V_s, j \in V_a\} \end{aligned}$$

Now, for each anchor $i \in V_a$, its position $a_i \in \mathbb{R}^d$ is assumed to be known. Moreover, each edge $(i,j) \in E_{ss}$ (resp. $(i,j) \in E_{sa}$) is given a positive weight $d_{ij}$ (resp. $\bar{d}_{ij}$), which can be viewed as the distance between sensor $i$ and sensor $j$ (resp. sensor $i$ and anchor $j$). In particular, the existence of an edge between $i,j \in V$ means that the distance between $i$ and $j$ is known. This allows us to assume without loss that $E_{aa} = \{(i,j) : i,j \in V_a\} \subset E$, as the distance between any two anchors is trivially known. In summary, an instance of the network localization problem is given by a 4–tuple $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a}, d)$, where $G = ((V_s, V_a), (E_{ss}, E_{sa}, E_{aa}))$ is the input network, $\mathbf{d} = (d_{ij})_{(i,j)\in E_{ss}} \in \mathbb{R}_+^{|E_{ss}|}$ and $\bar{\mathbf{d}} = (\bar{d}_{ij})_{(i,j)\in E_{sa}} \in \mathbb{R}_+^{|E_{sa}|}$ are the vectors of available distance measurements, $\mathbf{a} = (a_i)_{i\in V_a} \in \mathbb{R}^{dm}$ is the vector containing the positions of the anchors, and $d \geq 1$ is the given dimension. For simplicity's sake, suppose that the given distance data are exact. Then, the goal of the network localization problem is to find an assignment of coordinates $\tilde{\mathbf{x}} = (\tilde{x}_1; \ldots; \tilde{x}_n) \in \mathbb{R}^{dn}$ to the nodes in $V_s$ (which, together with $\mathbf{a} \in \mathbb{R}^{dm}$, is called a *localization* or *realization* of $G$) such that $\tilde{\mathbf{x}}$ satisfies the following system:

$$\begin{aligned} \|x_i - x_j\|_2^2 &= d_{ij}^2 \qquad \text{for } (i,j) \in E_{ss} \\ \|a_i - x_j\|_2^2 &= \bar{d}_{ij}^2 \qquad \text{for } (i,j) \in E_{sa} \\ x_i &\in \mathbb{R}^d \qquad \text{for } i = 1, \ldots, n \end{aligned} \tag{1}$$

In general, there could be multiple solutions to the above system. However, if the graph $G$ is generically globally rigid in $\mathbb{R}^d$ and if the instance $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a}, d)$ has at least $d+1$ anchors and admits a generic[2] localization $\mathbf{p} = (\tilde{\mathbf{x}}; \mathbf{a}) \in \mathbb{R}^{d(n+m)}$, then there is a unique solution to the above system; see, e.g., [11]. Moreover, there is an efficient algorithm for verifying whether $G$ is generically globally rigid in $\mathbb{R}^d$ [15]. Unfortunately, such an efficient *verification* procedure does not translate into an efficient *realization* procedure. In particular, even if we know that $G$ is generically globally rigid in $\mathbb{R}^d$, it is still intractable to find the solution to Problem (1) (or declare that no such solution exists) [2], [7].

In an attempt to circumvent this problem, So and Ye [29] introduced the notion of unique $d$–localizability. Specifically, an instance $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a}, d)$ of the network localization problem is said to be *uniquely $d$–localizable* if (i) the system (1) has a unique solution $\tilde{\mathbf{x}} = (\tilde{x}_1; \ldots; \tilde{x}_n) \in \mathbb{R}^{dn}$, and (ii) for any $l > d$, $((\tilde{x}_1; \mathbf{0}); (\tilde{x}_2; \mathbf{0}); \ldots; (\tilde{x}_n; \mathbf{0})) \in \mathbb{R}^{ln}$ is the unique solution to the following system:

$$\begin{aligned} \|x_i - x_j\|_2^2 &= d_{ij}^2 \qquad \text{for } (i,j) \in E_{ss} \\ \|(a_i; \mathbf{0}) - x_j\|_2^2 &= \bar{d}_{ij}^2 \qquad \text{for } (i,j) \in E_{sa} \\ x_i &\in \mathbb{R}^l \qquad \text{for } i = 1, \ldots, n \end{aligned} \tag{2}$$

[2]We say that a point $\mathbf{p} = (p_1, \ldots, p_l) \in \mathbb{R}^{dl}$ is *generic* if there does not exist a non–zero polynomial $h : \mathbb{R}^{dl} \to \mathbb{R}$ with integer coefficients such that $h(p_1, \ldots, p_l) = 0$.

Geometrically, the above conditions state that the given instance has a unique localization in $\mathbb{R}^l$ for all $l \geq d$, and that localization is given by $(\tilde{\mathbf{x}}, \mathbf{a}) \in \mathbb{R}^{dn} \times \mathbb{R}^{dm}$. We should emphasize that the notion of unique $d$–localizability depends both on the combinatorial structure of $G$ and the geometric information embedded in $(\mathbf{d}, \bar{\mathbf{d}}) \in \mathbb{R}_+^{|E_{ss}|} \times \mathbb{R}_+^{|E_{sa}|}$ and $\mathbf{a} \in \mathbb{R}^{dm}$.

As shown in [29], the problem of deciding whether a given network localization instance is uniquely $d$–localizable, as well as the problem of determining the node positions of a uniquely $d$–localizable instance, can be solved efficiently by solving the following SDP:

$$\begin{aligned} (\mathbf{0}; e_i - e_j)(\mathbf{0}; e_i - e_j)^T \bullet Z &= d_{ij}^2 \qquad \text{for } (i,j) \in E_{ss} \\ (a_i; -e_j)(a_i; -e_j)^T \bullet Z &= \bar{d}_{ij}^2 \qquad \text{for } (i,j) \in E_{sa} \\ Z_{1:d,1:d} &= I_d \\ Z &\in \mathcal{S}_+^{d+n} \end{aligned} \tag{3}$$

Here, $e_i \in \mathbb{R}^n$ is the $i$–th standard basis vector (where $i = 1, \ldots, n$), $I_d$ is the $d \times d$ identity matrix, $\mathcal{S}_+^{d+n}$ is the set of $(d+n)\times(d+n)$ symmetric positive semidefinite matrices, $A \bullet B = \mathrm{tr}(AB)$ is the trace inner product of two symmetric matrices $A, B$ of the same dimension, and the constraint $Z_{1:d,1:d} = I_d$ implies that the matrix $Z$ has the form:

$$Z = \begin{bmatrix} I_d & X \\ X^T & Y \end{bmatrix} \tag{4}$$

where $X \in \mathbb{R}^{d\times n}$ and $Y \in \mathbb{R}^{n\times n}$. Specifically, So and Ye proved the following theorem:

*Theorem 1:* (cf. [29, Theorem 2]) Suppose that the input graph $G$ is connected. Then, the following are equivalent:

1) The instance $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a}, d)$ is uniquely $d$–localizable.
2) The solution matrix $Z$ to (3), as given by (4), satisfies $Y = X^T X$.

Thus, in order to check whether a given instance is uniquely $d$–localizable, it suffices to solve the SDP (3) and check whether the solution matrix $Z$ has the form $Y = X^T X$. Moreover, if this is the case, then the unique localization of the $i$–th sensor is given by the $i$–th column of the matrix $X$, where $i = 1, \ldots, n$. We refer the reader to [29] for further details.

## III. Unique $d$–Localizability and Its Rigidity–Theoretic Counterpart

At this point, it is not clear how the notion of unique $d$–localizability fits into the framework of rigidity theory. As a result, the connection between the approach of [29] and other rigidity–based approaches (see, e.g., [11], [12], [14], [33]) for testing network localizability is not entirely obvious. We now remedy this situation by establishing the precise relationship between the notion of unique $d$–localizability and the rigidity–theoretic notion of universal rigidity. We begin with a definition.

*Definition 1:* Let $d, l \geq 1$ be integers. Let $G = (V, E)$ be an $l$–vertex graph, and let $\mathbf{p} = (p_1; \ldots; p_l) \in \mathbb{R}^{dl}$ be its

realization in $\mathbb{R}^d$. We say that $(G, \mathbf{p})$ is *universally rigid* in $\mathbb{R}^d$ if for any realization $\mathbf{q} = (q_1; \ldots; q_l) \in \mathbb{R}^{hl}$ of $G$ in $\mathbb{R}^h$ (where $h \geq 1$ is arbitrary), we have:

$$\Big[\|p_i - p_j\|_2 = \|q_i - q_j\|_2 \text{ for } (i,j) \in E\Big]$$
$$\implies \Big[\|p_i - p_j\|_2 = \|q_i - q_j\|_2 \text{ for } 1 \leq i < j \leq l\Big]$$

In other words, $\mathbf{p}$ is the unique (up to congruence) realization of $G$ in *any* Euclidean space. If $(G, \mathbf{p})$ is universally rigid in $\mathbb{R}^d$ for all generic realizations $\mathbf{p} \in \mathbb{R}^{dl}$, then we say that $G$ is *generically universally rigid* in $\mathbb{R}^d$.

The reader should contrast this definition with that of global rigidity in $\mathbb{R}^d$, which only requires that $\mathbf{p}$ is the unique (up to congruence) realization of $G$ in $\mathbb{R}^d$; see, e.g., [8]. In particular, the notion of universal rigidity is more restrictive than that of global rigidity. However, its advantage would become clear after we establish the following equivalence result:

*Theorem 2:* Suppose that $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a}, d)$, where $G = ((V_s, V_a), (E_{ss}, E_{sa}, E_{aa}))$, is a uniquely $d$–localizable instance of the network localization problem, with $\mathbf{p} = (\tilde{\mathbf{x}}; \mathbf{a}) \in \mathbb{R}^{d(|V_s|+|V_a|)}$ being its unique localization in $\mathbb{R}^l$ for all $l \geq d$. Then, $(G, \mathbf{p})$ is universally rigid in $\mathbb{R}^d$.

Conversely, let $G = (V, E)$ be a graph with $|V| \geq d+1$, and let $\mathbf{p} \in \mathbb{R}^{d|V|}$ be a realization of $G$ in $\mathbb{R}^d$. Suppose that $(G, \mathbf{p})$ is universally rigid in $\mathbb{R}^d$. Then, whenever $V_s, V_a \subset V$ are such that $|V_a| \geq d+1$, $V_s = V \backslash V_a$, and there exist $d+1$ affinely independent vectors in the family $\{p_i\}_{i \in V_a}$, the instance $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{p}|_{V_a}, d)$, where $\mathbf{d} = (\|p_i - p_j\|_2)_{(i,j) \in E_{ss}}$, $\bar{\mathbf{d}} = (\|p_i - p_j\|_2)_{(i,j) \in E_{sa}}$, and $\mathbf{p}|_{V_a} = (p_i)_{i \in V_a}$, is uniquely $d$–localizable.

*Proof:* Let $\mathbf{q} \in \mathbb{R}^{h(|V_s|+|V_a|)}$ be a realization of $G$ in $\mathbb{R}^h$ for some $h \geq 1$. Since $G$ contains the complete subgraph $G_a = (V_a, E_{aa})$, we can apply an isometry $T$ to $\mathbf{q}$ so that $T(q_i) = a_i$ for all $i \in V_a$. However, since $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a}, d)$ is uniquely $d$–localizable with $\mathbf{p} = (\tilde{\mathbf{x}}; \mathbf{a}) \in \mathbb{R}^{d(|V_s|+|V_a|)}$ being its unique localization in $\mathbb{R}^l$ for all $l \geq d$, we must have $T(\mathbf{q}) = \mathbf{p}$. Since $T$ is an isometry, it follows that $\|p_i - p_j\|_2 = \|q_i - q_j\|_2$ for $1 \leq i < j \leq |V_s| + |V_a|$, whence $(G, \mathbf{p})$ is universally rigid in $\mathbb{R}^d$.

Conversely, suppose that $(G, \mathbf{p})$ is universally rigid in $\mathbb{R}^d$, and that $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{p}|_{V_a}, d)$ is not uniquely $d$–localizable. Then, there exists another realization $\mathbf{p}' \in \mathbb{R}^{h|V|}$ of $G$ in $\mathbb{R}^h$ for some $h \geq d$, such that $p'_i = (p_i; \mathbf{0}) \in \mathbb{R}^h$ for all $i \in V_a$. Since $(G, \mathbf{p})$ is universally rigid in $\mathbb{R}^d$, there exists an isometry $T$ such that $T(\mathbf{p}') = \mathbf{p}$. On the other hand, since $T$ fixes $\mathbf{p}|_{V_a}$ and there are $d+1$ affinely independent vectors in $\{p_i\}_{i \in V_a}$, we conclude that $T$ must be the identity. In particular, this implies that $\mathbf{p}' = \mathbf{p}$, which is a contradiction. Hence, the instance $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{p}|_{V_a}, d)$ is uniquely $d$–localizable, as desired. ■

Theorem 2 has important algorithmic consequences. Indeed, let $G = (V, E)$ be an $l$–vertex graph and $\mathbf{p} \in \mathbb{R}^{dl}$ be a realization of $G$ in $\mathbb{R}^d$. Suppose that the family $\{p_i\}_{i \in V}$ contains $d+1$ affinely independent vectors. Then, by Theorems 1 and 2, we see that it is possible to check whether $(G, \mathbf{p})$ is universally rigid in $\mathbb{R}^d$ efficiently using SDP. Moreover, if $(G, \mathbf{p})$ is indeed universally rigid in $\mathbb{R}^d$, then the unique realization $\mathbf{p}$ can also be found efficiently using SDP. By contrast, it is $\mathcal{NP}$–hard to check whether $(G, \mathbf{p})$ is globally rigid in $\mathbb{R}^d$ [26]. Moreover, unless $\mathcal{RP} = \mathcal{NP}$, there does not exist an efficient algorithm for finding the unique realization $\mathbf{p}$ of a globally rigid instance $(G, \mathbf{p})$ [2]. Thus, even though universal rigidity is a more restrictive notion than global rigidity, it is much more amenable to algorithmic treatment.

## IV. Construction of Universally Rigid Instances

The notion of universal rigidity would not be very interesting if it does not capture a sufficiently rich class of graphs. In this section, we prove that trilateration graphs are generically universally rigid. Furthermore, we show how generically universally rigid graphs can be constructed in an incremental manner. These results show that the notion of universal rigidity is indeed non–trivial.

### A. Generic Universal Rigidity of Trilateration Graphs

Let us begin with the definition of trilateration graphs.

*Definition 2:* Let $d, l \geq 1$ be integers with $l \geq d+1$. An $l$–vertex graph $G = (V, E)$ is called a *$d$–trilateration graph* if there exists an ordering $\{1, 2, \ldots, l\}$ of the vertices in $V$ (called a *$d$–trilateration ordering*) such that (i) the first $d+1$ vertices $1, 2, \ldots, d+1$ form a complete graph, and (ii) every vertex $j \geq d+2$ is connected to at least $d+1$ of the vertices $1, 2, \ldots, j-1$.

The family of trilateration graphs was studied in [1], [11], where it was shown that a $d$–trilateration graph is generically globally rigid in $\mathbb{R}^d$. This implies that if $G$ is an $(n+m)$–vertex $d$–trilateration graph with $n \geq 1$ sensors and $m \geq d+1$ anchors, then the instance $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a}, d)$ has a unique localization in $\mathbb{R}^d$ (i.e. there is a unique solution to the system (1)) whenever $(\mathbf{d}, \bar{\mathbf{d}})$ and $\mathbf{a}$ induce a generic realization of $G$ in $\mathbb{R}^d$. We now sharpen this result by showing that the instance $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a}, d)$ is in fact uniquely $d$–localizable, i.e. it has a unique localization in $\mathbb{R}^l$ for all $l \geq d$.

*Theorem 3:* Let $G$ be an $(n+m)$–vertex $d$–trilateration graph with $n \geq 1$ sensors and $m \geq d+1$ anchors. Given a network localization instance $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a}, d)$, let $\tilde{\mathbf{x}} \in \mathbb{R}^{dn}$ be a feasible realization of the sensors. Suppose that the realization $(\tilde{\mathbf{x}}, \mathbf{a}) \in \mathbb{R}^{dn} \times \mathbb{R}^{dm}$ of $G$ is generic. Then, the instance $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a}, d)$ is uniquely $d$–localizable, and $(\tilde{\mathbf{x}}, \mathbf{a})$ is its unique localization in $\mathbb{R}^d$.

*Proof:* Let $V_s = \{1, \ldots, n\}$ be the set of sensors and $V_a = \{n+1, \ldots, n+m\}$ be the set of anchors. Since $G$ is a $d$–trilateration graph, there exists an ordering $\pi$ of the vertices that satisfies the conditions in Definition 2. Let us first consider the case where the first $d+1$ vertices under $\pi$ are all anchors and show that the desired conclusion holds. Towards that end, consider the vertex $\pi(d+2)$, which is the $(d+2)$–nd vertex under $\pi$. If it is an anchor, then there is nothing to argue. Hence, suppose that it is a sensor. Since it is connected to vertices $\pi(1), \ldots, \pi(d+1)$, we have the following constraints in the SDP (3), where $i = 1, \ldots, d+1$:

$$\|a_{\pi(i)}\|_2^2 - 2a_{\pi(i)}^T x_{\pi(d+2)} + Y_{\pi(d+2),\pi(d+2)} = \bar{d}^2_{\pi(i),\pi(d+2)}$$

Here, $x_{\pi(d+2)}$ is the $\pi(d+2)$–nd column of the matrix $X \in \mathbb{R}^{d\times n}$ in (3), and $Y_{\pi(d+2),\pi(d+2)}$ is the $(\pi(d+2),\pi(d+2))$–nd entry of the matrix $Y \in \mathbb{R}^{n\times n}$ in (3). Upon eliminating $Y_{\pi(d+2),\pi(d+2)}$, we obtain the following system of $d$ linear equations in $x_{\pi(d+2)} \in \mathbb{R}^d$ (here, we have $i = 1,\ldots,d$):

$$(a_{\pi(i)} - a_{\pi(d+1)})^T x_{\pi(d+2)} =$$

$$\frac{1}{2}\left(\|a_{\pi(i)}\|_2^2 - \|a_{\pi(d+1)}\|_2^2 + \bar{d}^2_{\pi(d+1),\pi(d+2)} - \bar{d}^2_{\pi(i),\pi(d+2)}\right)$$

Since $(\tilde{\mathbf{x}}, \mathbf{a})$ is generic, the vectors $\{a_{\pi(i)} - a_{\pi(d+1)}\}_{i=1,\ldots,d}$ are linearly independent. Thus, there is a unique solution to the above system, namely, $x_{\pi(d+2)} = \tilde{x}_{\pi(d+2)}$. This in turn implies that $Y_{\pi(d+2),\pi(d+2)} = \|\tilde{x}_{\pi(d+2)}\|_2^2$. Moreover, since $Y - X^TX \succeq \mathbf{0}$ by the Schur complement and $(Y - X^TX)_{\pi(d+2),\pi(d+2)} = 0$, we conclude that $Y_{\pi(d+2),l} = Y_{l,\pi(d+2)} = \tilde{x}^T_{\pi(d+2)}x_l$ for $l = 1,\ldots,n$.

Now, suppose that for $i = d+2,\ldots,j$ with $\pi(i) \in V_s$, we have $x_{\pi(i)} = \tilde{x}_{\pi(i)}$, $Y_{\pi(i),\pi(i)} = \|\tilde{x}_{\pi(i)}\|_2^2$, and $Y_{\pi(i),l} = Y_{l,\pi(i)} = \tilde{x}^T_{\pi(i)}x_l$ for $l = 1,\ldots,n$. Consider vertex $\pi(j+1)$. Without loss of generality, we may assume that it is a sensor. Then, we have the following constraints in the SDP (3):

$$\|\tilde{x}_{\pi(l)}\|_2^2 - 2\tilde{x}^T_{\pi(l)}x_{\pi(j+1)} + Y_{\pi(j+1),\pi(j+1)} = d^2_{\pi(l),\pi(j+1)}$$

for $(\pi(l),\pi(j+1)) \in E_{ss}$, $l = 1,\ldots,j$; and

$$\|a_{\pi(l)}\|_2^2 - 2a^T_{\pi(l)}x_{\pi(j+1)} + Y_{\pi(j+1),\pi(j+1)} = \bar{d}^2_{\pi(j+1),\pi(l)}$$

for $(\pi(j+1),\pi(l)) \in E_{sa}$, $l = 1,\ldots,j$. Since vertex $\pi(j+1)$ is connected to at least $d+1$ of the vertices $\pi(1),\ldots,\pi(j)$, there are at least $d+1$ linear equations in the above system. Since $(\tilde{\mathbf{x}}, \mathbf{a})$ is generic, we can extract $d+1$ independent linear equations from it. Then, by using the same argument as before, we see that $x_{\pi(j+1)} = \tilde{x}_{\pi(j+1)}$, $Y_{\pi(j+1),\pi(j+1)} = \|\tilde{x}_{\pi(j+1)}\|_2^2$, and $Y_{\pi(j+1),l} = Y_{l,\pi(j+1)} = \tilde{x}^T_{\pi(j+1)}x_l$ for $l = 1,\ldots,n$. Hence, the inductive step is completed. In particular, we conclude that $Y = \tilde{X}^T\tilde{X}$, where $\tilde{X} = [\ \tilde{x}_1 \ \cdots \ \tilde{x}_n\ ]$, is the unique solution to (3). It then follows from Theorem 1 that $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a}, d)$ is uniquely $d$–localizable.

In summary, we have shown that if $\mathbf{p} \in \mathbb{R}^{dl}$ is a generic realization of an $l$–vertex $d$–trilateration graph $G = (V, E)$ (where $l \geq d+1$), and if the positions of the first $d+1$ vertices under the $d$–trilateration ordering of $V$ are given, then $\mathbf{p}$ can be found by solving systems of linear equations whenever the distances $d_{ij} = \|p_i - p_j\|_2$ (for $(i,j) \in E$) are given. Moreover, it will be the unique realization of $G$ (with respect to the positions of the first $d+1$ vertices) in $\mathbb{R}^h$ for all $h \geq d$.

Now, let us consider the case where the first $d+1$ vertices under $\pi$ are not all anchors. Suppose that there are two generic realizations $(\tilde{\mathbf{x}}, \mathbf{a})$ and $(\tilde{\mathbf{y}}, \mathbf{a})$ of $G$. Note that by our remark in the preceding paragraph, both realizations must lie in $\mathbb{R}^d$ and are unique with respect to the positions of the first $d+1$ vertices under $\pi$. This implies that there exists an isometry $T : \mathbb{R}^d \to \mathbb{R}^d$ that maps $(\tilde{\mathbf{x}}, \mathbf{a})$ to $(\tilde{\mathbf{y}}, \mathbf{a})$. However, since the $m \geq d+1$ generically positioned anchors are fixed by $T$, it follows that $T$ must be the identity. In particular, we have $(\tilde{\mathbf{x}}, \mathbf{a}) = (\tilde{\mathbf{y}}, \mathbf{a})$, and the desired conclusion in the theorem statement follows. ■

An important consequence of Theorem 3 is that it establishes the existence of sparse uniquely $d$–localizable instances (i.e. those whose graphs have only $O(|V|)$ edges) for any fixed $d \geq 1$. This refutes a common belief that every uniquely $d$–localizable instance must have $\Omega(|V|^2)$ edges. Moreover, by combining Theorems 2 and 3, we obtain the following important corollary:

*Corollary 1:* $d$–trilateration graphs are generically universally rigid in $\mathbb{R}^d$ for all $d \geq 1$.

### *B. Incremental Constructions of Generically Universally Rigid Graphs*

Having demonstrated the existence of generically universally rigid graphs, we now show how they can be used to construct larger generically universally rigid graphs. The first construction is based on the idea of trilateration extension introduced in [11]. We begin with the definition.

*Definition 3:* Let $d \geq 1$ be an integer. We say that a graph $G' = (V', E')$ is a *$d$–trilateration extension* of another graph $G = (V, E)$, where $|V| \geq d+1$, if $V' = V \cup \{w\}$ with $w \notin V$, $E \subset E'$, and $|\{(w,v) \in E' : v \in V\}| \geq d+1$. In other words, $G'$ is obtained from $G$ by adding a new vertex to $G$ and connecting it to at least $d+1$ vertices of $G$.

*Theorem 4:* Let $G = (V, E)$ be generically universally rigid in $\mathbb{R}^d$ with $|V| \geq d+1$, and let $G' = (V', E')$ be a $d$–trilateration extension of $G$. Then, $G'$ is generically universally rigid in $\mathbb{R}^d$.

The proof of Theorem 4 follows a similar argument as in the proof of Theorem 3. Due to space limitation, we shall defer it to the full version of this paper [34].

Our second construction is based on taking the union of generically universally rigid graphs. Specifically, we have the following theorem:

*Theorem 5:* Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two generically universally rigid graphs in $\mathbb{R}^d$ with $|V_1 \cap V_2| \geq d+1$. Then, the graph $G = (V_1 \cup V_2, E_1 \cup E_2)$ is also generically universally rigid in $\mathbb{R}^d$.

*Proof:* Let $\mathbf{p} \in \mathbb{R}^{d|V_1\cup V_2|}$ be a generic realization of $G$ in $\mathbb{R}^d$, and let $\mathbf{q} \in \mathbb{R}^{h|V_1\cup V_2|}$ be an arbitrary realization of $G$ in $\mathbb{R}^h$ for some $h \geq 1$. Since $G_1$ is generically universally rigid in $\mathbb{R}^d$, there exists an isometry $T_1$ such that $T_1(\mathbf{q})|_{G_1} = \mathbf{p}|_{G_1}$. Here, $T_1(\mathbf{q})|_{G_1}$ (resp. $\mathbf{p}|_{G_1}$) denotes the restriction of $T_1(\mathbf{q})$ (resp. $\mathbf{p}$) onto the coordinates corresponding to the vertices of $G_1$. By a similar argument, there exists an isometry $T_2$ such that $T_2(\mathbf{q})|_{G_2} = \mathbf{p}|_{G_2}$. Now, observe that there exists an isometry $T$ such that $T(T_1(\mathbf{q}))|_{G_2} = T_2(\mathbf{q})|_{G_2} = \mathbf{p}|_{G_2}$. Since $T$ fixes the generically positioned vertices in $V_1 \cap V_2$ and since $|V_1 \cap V_2| \geq d+1$ by assumption, it follows that $T$ must be the identity. In particular, since $T_1(\mathbf{q})|_{G_1} = \mathbf{p}|_{G_1}$ and $T_1(\mathbf{q})|_{G_2} = \mathbf{p}|_{G_2}$, we conclude that $T_1(\mathbf{q}) = \mathbf{p}$. This completes the proof. ■

## V. Edge Sparsification and Application to SDP Speedup

Due in part to their simplicity and nice theoretical properties, trilateration graphs and trilateration–based schemes have

been used in many localization algorithms. In this section, we demonstrate yet another application of trilateration graphs, namely, to speed up existing convex optimization–based localization algorithms, while at the same time preserving their accuracy.

To motivate this application and for the sake of concreteness, let us consider again the SDP (3). As demonstrated in [3], [4], the SDP (3) can often produce highly accurate results. (In fact, this is quite typical of all recent convex optimization–based localization algorithms; see, e.g., [3], [4], [20], [30], [32].) However, a major drawback of (3) is that it is computationally demanding. Indeed, the SDP (3) contains $(d+n)(d+n+1)/2$ variables and $d(d+1)/2+|E_{ss}|+|E_{sa}|$ equality constraints. As such, an instance of (3) with only a few hundred nodes is considered to be a challenge for standard SDP solvers. A natural way to speed up the solution time of (3) is to reduce the number of constraints in it. In other words, we can preprocess the input before solving the SDP (3) by removing edges from the input network. Such an edge sparsification approach has indeed been pursued by several researchers; see, e.g., [20], [32]. However, the proposed schemes are quite ad hoc in nature and may remove edges that are crucial to an accurate localization of the sensors. It could happen, for example, that the sparsified instance is not uniquely $d$–localizable, even though the original instance is so. Thus, it is natural to ask whether one can develop an efficient edge sparsification heuristic that can *provably* preserve the localization properties of the input.

Using the generical universal rigidity of trilateration graphs, we answer this question in the affirmative. The high–level idea is simple: decompose the input graph into a number of $d$–trilateration subgraphs and remove any redundant edges in those subgraphs. For concreteness' sake, let us describe our edge sparsification heuristic—which we call the *Equivalent Edge Sparsification (EES)* heuristic—for the case where $d = 2$ (which, of course, is a case of practical interest); see Algorithm 1. The generalization to $d \geq 3$ will be straightforward.

Note that in line 8 of Algorithm 1, there is some freedom in choosing which vertex $i$ is to enter $S$ and which three edges that connect $i$ to $S$ are to be included in $E'$. In Section VI, we shall discuss the specific selection rules used in our simulations. For now, let us assume that they are chosen in an arbitrary fashion.

**Algorithm 1** EES HEURISTIC

**Input:** A graph $G = (V, E)$.
**Output:** An edge–sparsified graph $\text{EES}(G)$.
1: Set $S \leftarrow \emptyset$, $T \leftarrow V$, and $E' \leftarrow \emptyset$.
2: **if** there exists a 3–clique $K_3 = \{i_1, i_2, i_3\}$ in $G$ **then**
3: set $S \leftarrow S \cup K_3$, $T \leftarrow T \backslash K_3$, and $E' \leftarrow \{(i_{j_1}, i_{j_2}) : 1 \leq j_1 < j_2 \leq 3\}$
4: **else**
5: return $G$ and stop
6: **end if**
7: **while** $|S| < |V|$ **do**
8: Let $i \in T$ be such that $|\{(i,j) \in E : j \in S\}| \geq 3$. Set $S \leftarrow S \cup \{i\}$, $T \leftarrow T \backslash \{i\}$, and $E' \leftarrow E' \cup \{(i, j_k) : j_k \in S,\ k = 1, 2, 3\}$. The vertices $i \in T$ and $j_1, j_2, j_3 \in S$ are chosen according to some pre–specified rule.
9: **if** $|\{(i,j) \in E : j \in S\}| < 3$ for all $i \in T$ **then**
10: let $G_T = (T, E_T)$, where $E_T = \{(i,j) \in E : i, j \in T\}$, and compute $(T, E'_T) = \text{EES}(G_T)$
11: set $E' \leftarrow E' \cup \{(i,j) \in E : i \in S,\ j \in T\} \cup E'_T$, $S \leftarrow S \cup T$, and $T \leftarrow \emptyset$
12: **end if**
13: **end while**
14: return $(V, E')$

### A. Analysis of the EES Heuristic

Naturally, we would be interested in determining various theoretical properties of the EES heuristic. Towards that end, let us first analyze the structure of the graph produced by the heuristic. We begin with a definition.

*Definition 4:* Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be vertex–disjoint subgraphs of a graph $G = (V, E)$. The *graph join* of $G_1$ and $G_2$ under $G$ is the graph $G_1 +^G G_2$ whose vertex set is $V_1 \cup V_2$ and edge set is $E_1 \cup E_2 \cup \{(u,v) \in E : u \in V_1,\ v \in V_2\}$.

*Theorem 6:* Let $\text{EES}(G)$ be the output of the EES heuristic when applied to the input graph $G = (V, E)$. Then, we have the following decomposition:

$$\text{EES}(G) = G_0 +^G G_1 +^G G_2 +^G \cdots +^G G_k \tag{5}$$

Here, $k \geq 0$ is some integer, $G_i = (V_i, E_i)$ is a minimal 2–trilateration graph (i.e. $G'_i = (V_i, E'_i)$ is not a 2–trilateration graph for any $E'_i \subsetneq E_i$) for $i = 0, 1, \ldots, k-1$, $G_k = (V_k, E_k)$ is either a minimal 2–trilateration graph or a triangle–free graph, and $V_0, \ldots, V_k$ are mutually disjoint with $V = \cup_{i=0}^k V_i$. In particular, each $G_i$ is a subgraph of $G$, and the number of edges in $\text{EES}(G)$ is bounded above by $O(k|V|)$ if $G_k$ is a minimal 2–trilateration graph, and by $O(k|V| + |V_k|^2)$ if $G_k$ is a triangle–free graph.

*Proof:* The desired decomposition follows directly from the definition of the EES heuristic. To determine the number of edges in $\text{EES}(G)$, we first note that the number of edges in an $l$–vertex minimal 2–trilateration graph is $O(l)$. Thus, if $G_k$ is a minimal 2–trilateration graph, then the number of edges in $\text{EES}(G)$ can be bounded above by $\sum_{i=0}^k O(|V_i|) + 2\sum_{j=1}^k \sum_{i=j}^k O(|V_i|) = O(k|V|)$ (since $|V| = \sum_{i=0}^k |V_i|$). On the other hand, if $G_k$ is a triangle–free graph, then by Turán's theorem (see, e.g., [5, Chapter VI, Theorem 1.1]), we have $|E_k| = O(|V_k|^2)$. Thus, it follows that the number of edges in $\text{EES}(G)$ is bounded above by $\sum_{i=0}^{k-1} O(|V_i|) + O(|V_k|^2) + 2\sum_{j=1}^{k-1} \left(\sum_{i=j}^{k-1} O(|V_i|) + O(|V_k|)\right) = O(k|V| + |V_k|^2)$. This completes the proof. ∎

Theorem 6 shows that the graph returned by the EES heuristic does indeed have a small number of edges in general. Next, we come to the most important feature of the EES heuristic, namely that it preserves the localization properties of the input instance. We first prove the following theorem:

*Theorem 7:* Let $G = ((V_s, V_a), (E_{ss}, E_{sa}, E_{aa}))$, and let $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a}, 2)$ be an arbitrary feasible network localization instance. Suppose that $\mathbf{p} = (\tilde{\mathbf{x}}; \mathbf{a}) \in \mathbb{R}^{2(|V_s|+|V_a|)}$ is a generic realization of the instance $(\text{EES}(G), (\mathbf{d}, \bar{\mathbf{d}})|_{EES(G)}, \mathbf{a}, 2)$ in $\mathbb{R}^2$, where $(\mathbf{d}, \bar{\mathbf{d}})|_{EES(G)}$ is the restriction of $(\mathbf{d}, \bar{\mathbf{d}})$ onto the edges of $\text{EES}(G)$. Then, $\mathbf{p}$ is also a generic realization of $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a}, 2)$ in $\mathbb{R}^2$.

*Remark.* Since $\text{EES}(G)$ is a subgraph of $G$, it is clear that *any* realization $\mathbf{p} \in \mathbb{R}^{2(|V_s|+|V_a|)}$ of $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a}, 2)$ will also be a realization of $(\text{EES}(G), (\mathbf{d}, \bar{\mathbf{d}})|_{EES(G)}, \mathbf{a}, 2)$. Thus, the converse to Theorem 7 holds even without the genericity assumption.

*Proof:* Let $E \equiv E_{ss} \cup E_{sa} \cup E_{aa}$, and consider the decomposition of $\text{EES}(G)$ given by (6). Recall that by Theorem 6, $G_0, \ldots, G_{k-1}$ are 2–trilateration graphs. Thus, by Corollary 1 and the genericity of $\mathbf{p}$, we see that $(G_j, \mathbf{p}|_{G_j})$ is universally rigid in $\mathbb{R}^2$ for $j = 0, 1, \ldots, k-1$, where $\mathbf{p}|_{G_j}$ is the restriction of $\mathbf{p}$ onto the vertices of $G_j$. In particular, since $\mathbf{p}$ satisfies the distance constraints corresponding to the edges in $E_j$ and $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a}, 2)$ is feasible, we see that $\mathbf{p}$ must also satisfy the distance constraints corresponding to the edges in $(V_j \times V_j) \cap E$ for $j = 0, 1, \ldots, k-1$. This, together with the definition of the EES heuristic, implies that $\mathbf{p}$ satisfies the distance constraints corresponding to the edges in $((V \backslash V_k) \times V) \cap E$.

Now, if $G_k$ is a 2–trilateration graph, then by the above argument, we see that $\mathbf{p}$ also satisfies the distance constraints corresponding to the edges in $(V_k \times V) \cap E$. Otherwise, $G_k$ is a triangle–free graph, and we have $E_k = (V_k \times V_k) \cap E$ by definition of the EES heuristic. In particular, we see that $\mathbf{p}$ again satisfies the distance constraints corresponding to the edges in $(V_k \times V) \cap E$. Thus, in both cases, we can conclude that $\mathbf{p}$ satisfies the distance constraints corresponding to the edges in $\big[((V \backslash V_k) \times V) \cap E\big] \cup \big[(V_k \times V) \cap E\big] = (V \times V) \cap E = E$. In other words, $\mathbf{p}$ is a generic realization of $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a}, 2)$ in $\mathbb{R}^2$, as desired. ∎

In a similar fashion, we can prove the following theorem:

*Theorem 8:* Suppose that $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a}, 2)$, where $G = ((V_s, V_a), (E_{ss}, E_{sa}, E_{aa}))$, is a uniquely 2–localizable instance of the sensor network localization problem, with $\mathbf{p} = (\tilde{\mathbf{x}}; \mathbf{a}) \in \mathbb{R}^{2(|V_s|+|V_a|)}$ being its unique localization in $\mathbb{R}^l$ for all $l \geq 2$. Furthermore, suppose that $\mathbf{p}$ is generic. Then, the instance $(\text{EES}(G), (\mathbf{d}, \bar{\mathbf{d}})|_{EES(G)}, \mathbf{a}, 2)$ is also uniquely 2–localizable, and $\mathbf{p}$ is its unique localization in $\mathbb{R}^l$ for all $l \geq 2$.

The proof of Theorem 8 is similar to that of Theorem 7 and is deferred to the full version of this paper [34].

We remark that although Theorems 6 to 8 are stated for the case where $d = 2$, they can be generalized to other values of $d$ in a straightforward manner.

Finally, let us discuss the implementation and analyze the computational complexity of the EES heuristic. At the beginning, we construct a global list $\mathcal{L}$ of all 3–cliques in the input graph $G$ and maintain, for each vertex $i \in V$, a list $\mathcal{L}_i$ of 3–cliques to which it belongs. All of these can be done in $O(|V|^3)$ time. Lines 1 and 8 can be implemented in $O(|V|)$ and $O(|S| \cdot |V|)$ time, respectively. Given the list $\mathcal{L}$, line 2 can be implemented in $O(1)$ time. Now, if the condition in line 9 is satisfied, then before proceeding to line 10, we will first delete from $\mathcal{L}$ those 3–cliques that have at least one vertex in $S$ using the lists $\{\mathcal{L}_i\}_{i \in S}$. Such a pruning step allows us to conclude that throughout the course of the algorithm, each element in $\mathcal{L}$ is accessed at most a constant number of times. As a result, the total runtime of the EES heuristic can be bounded by $O(|V|^3)$. Note that the complexity of the EES heuristic is much lower than that required for solving the SDP (3). This justifies the use of the EES heuristic as a preprocessing procedure for speeding up the solution time of (3).

### B. Further Processing of the SDP

To summarize, we have just developed a novel edge sparsification heuristic that can provably preserve the localization properties of the input. To the best of our knowledge, this is the first heuristic with such a theoretical guarantee. The instance obtained after we apply the edge sparsification heuristic typically has fewer distance constraints, and hence its associated SDP can already be solved faster than the one associated with the original instance. It turns out that it is possible to further improve the computational efficiency of solving those SDPs by using recently developed general–purpose speedup techniques (i.e. these techniques apply to general SDPs and do not necessarily take advantage of the structure of the network localization problem; see, e.g., [13], [19], [20]). Moreover, besides improving the *solution time* of the SDP (3), one can also improve the *accuracy* of the solution by using a gradient descent procedure (see, e.g., [3]). Specifically, after solving the SDP (3) and obtaining a solution $X^*$, one can use it as a starting point for local gradient descent procedures. Due to space limitation, we refer the reader to [3], [13], [19], [20] for details of these approaches.

## VI. Simulation Results

In this section, we present some preliminary computational results to show the effectiveness of our edge sparsification approach in solving the network localization problem. All simulations are ran on a 2.8GHz CPU PC with 2 GB memory.

### A. Effectiveness of the EES Heuristic

To demonstrate the effectiveness of the EES heuristic, we randomly place 500 nodes over the unit square $[0, 1] \times [0, 1]$. Two vertices are connected by an edge if their distance is at most $\rho$, where $\rho = 0.05, 0.06, \ldots, 0.15$ (we shall refer to the parameter $\rho$ as the *radio range* of the sensors). Moreover, in line 8 of Algorithm 1, we choose the vertices $i \in T$ and $j_1, j_2, j_3 \in S$ according to the following strategy. First, we label the vertices using the Reverse Cuthill–McKee Ordering [9] (this typically can speed up matrix computations). Then, we pick the vertices $i \in T$ and $j_1, j_2, j_3 \in S$ so that $\{j_1, j_2, j_3\}$ forms a 3–clique. If this is not possible, then we just pick the vertex $i \in T$ with the smallest label; and among the neighbors of $i$ that are in $S$, we choose three that have the largest labels.

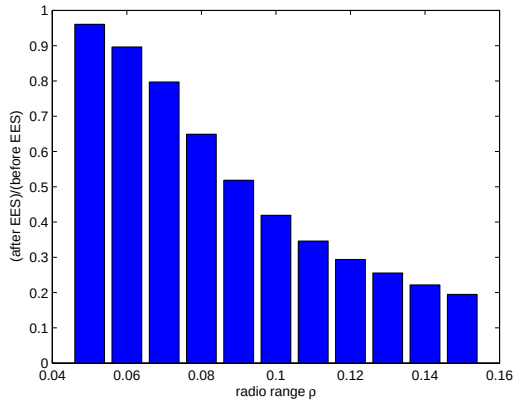


Fig. 1. The ratios of the number of edges before and after applying the EES heuristic to randomly generated 500–node sensor networks.

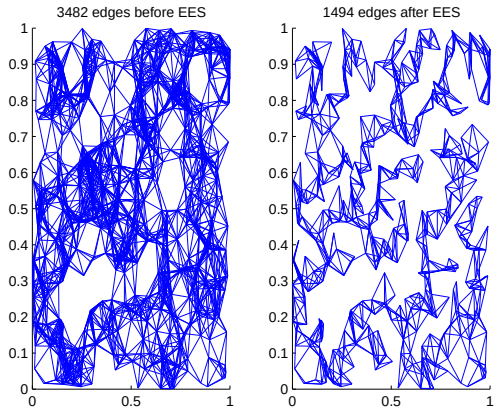


Fig. 2. Effect of the EES heuristic on a randomly generated 500–node sensor network.

The percentage of edges kept by the EES heuristic is shown in Figure 1. As the plot demonstrates, the reduction in the number of edges becomes more substantial as the radio range increases. One particular example is shown in Figure 2. There are 3482 edges in the 500–node network before we apply the EES heuristic. After we apply the heuristic, only 1494 edges are left.

### *B. Performance of EES–Preprocessed SDP–Based Localization Algorithms*

Next, we investigate the performance of various SDP–based localization algorithms when the input instance is first sparsified by the EES heuristic. To begin, we generate random sensor networks over the unit square with 90% sensors and 10% anchors, where the total number $n$ of nodes ranges from 100 to 1600. We set $\rho = 2/\sqrt{n}$ and solve the resulting network localization problems using four different schemes:

- FSDP: the SDP formulation (3) of Biswas and Ye [4]
- SSDP: the so–called *sparse SDP* formulation of Kim et al. [20]

| $n$ | FSDP | EES–FSDP | SSDP | EES–SSDP |
|---|---|---|---|---|
| 100 | 4.8 | 3.8 | 5.9 | 3.8 |
| 200 | 24.0 | 14.4 | 23.9 | 8.7 |
| 400 | 262.8 | 118.2 | 110.9 | 27.6 |
| 800 | 2439.3 | 1116.9 | 674.7 | 115.1 |
| 1600 | * | * | * | 639.1 |

TABLE I
COMPUTATION TIME COMPARISON (IN SECONDS); "*"=OUT OF MEMORY; RADIO RANGE $\rho = 2/\sqrt{n}$

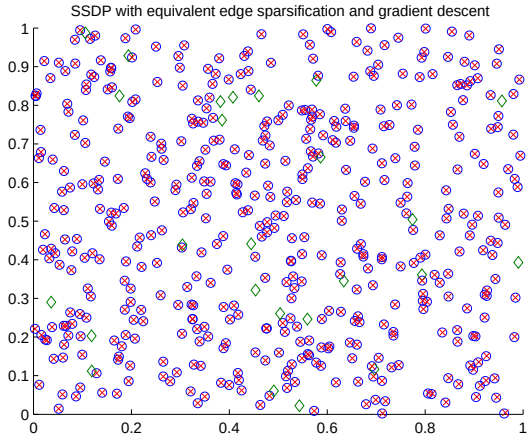


Fig. 3. Localization ($\times$) of a sensor network with 475 sensors ($\circ$), 25 anchors ($\Diamond$) and $\rho = 0.1$, computed by EES–SSDP with gradient descent.

- EES–FSDP: FSDP after applying the EES heuristic
- EES–SSDP: SSDP after applying the EES heuristic

The average computation time of each of these schemes is given in Table I. As one can see, the EES heuristic does help to significantly reduce the computation time of the SDP formulations in question. For the case where $n = 1600$, both FSDP and SSDP without EES give rise to very large SDPs, and our computer ran out of memory when solving them.

Figure 3 shows one of the results produced by EES–SSDP with gradient descent (see the discussion in Section V-B). We use "$\circ$" to denote the true location of a sensor (whose position is not known to the SDP) and "$\times$" to denote the location of a sensor computed by the algorithm. As can be seen from the figure, EES–SSDP with gradient descent can give a very accurate localization of the sensors.

To further evaluate the accuracy of the above schemes, we follow [4], [20], [30], [32] and use the Root Mean Square Distance (rmsd) to measure the discrepancy between the computed locations and true locations of the sensors, i.e. $rmsd = \left(n^{-1}\sum_{i=1}^{n}\|x_i - \bar{x}_i\|^2\right)^{1/2}$. Here, $x_i \in \mathbb{R}^d$ is the position of sensor $i$ as computed by any one of the schemes above, and $\bar{x}_i \in \mathbb{R}^d$ is its true position. As an illustration, we record the corresponding average rmsd values of FSDP and EES–FSDP in Table II. The discrepancy in the average rmsd values between FSDP and EES–FSDP can be attributed to the fact that the input instances are not uniquely 2–localizable. In

| $n$ | 100 | 200 | 400 | 800 |
|---|---|---|---|---|
| FSDP | 2.10e-02 | 3.23e-02 | 1.85e-02 | 1.17e-02 |
| EES–FSDP | 1.99e-02 | 3.18e-02 | 2.06e-02 | 1.99e-02 |

TABLE II
RMSD COMPARISON BETWEEN FSDP AND EES–FSDP

particular, the SDP (3) is simply a *relaxation* of the original problem and may not return an exact localization. Thus, the localizations returned by FSDP and EES–FSDP could be different.

## VII. Conclusion and Future Directions

In this paper, we studied the notion of universal rigidity and showed that it is not only more amenable to algorithmic treatment, but also more relevant to the efficient solvability of the network localization problem than the notion of global rigidity. Furthermore, we demonstrated the richness of the class of universally rigid instances. In particular, we proved that trilateration graphs, which are widely used in the design of localization algorithms, are generically universally rigid. Finally, we developed a novel edge sparsification heuristic that can reduce the number of edges (and hence distance measurements) in the input network while preserving its localization properties. We used such a heuristic to speed up existing convex optimization–based localization algorithms, and simulation results showed that our approach is promising.

In the future, it would be interesting to further investigate the properties of universally rigid instances and their algorithmic implications, as they are still not fully understood. Moreover, in view of the complexity results in [1], [2], [7], [26], it would be worthy to identify other classes of efficiently realizable *globally rigid* instances.

## Acknowledgments

The authors would like to thank Professor Wenan Zang for discussions related to the $d$–trilateration graphs. Anthony Man–Cho So's research is supported by Hong Kong Research Grants Council (RGC) General Research Fund (GRF) Project No. CUHK 416908.

## References

[1] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. O. Anderson, and P. N. Belhumeur, "A Theory of Network Localization," *IEEE Trans. Mobile Comput.*, vol. 5, no. 12, pp. 1663–1678, 2006.

[2] J. Aspnes, D. Goldenberg, and Y. R. Yang, "On the Computational Complexity of Sensor Network Localization," in *Proc. ALGOSENSORS 2004*, LNCS 3121, pp. 32–44, 2004.

[3] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye, "Semidefinite Programming Based Algorithms for Sensor Network Localization," *ACM Trans. Sensor Networks*, vol. 2, no. 2, pp. 188–220, 2006.

[4] P. Biswas and Y. Ye, "Semidefinite Programming for Ad Hoc Wireless Sensor Network Localization," in *Proc. IPSN 2004*, 2004, pp. 46–54.

[5] B. Bollobás, *Extremal Graph Theory*. London: Academic Press, 1978.

[6] J. Bruck, J. Gao, and A. A. Jiang, "Localization and Routing in Sensor Networks by Local Angle Information," *ACM Trans. Sensor Networks*, vol. 5, no. 1, Article 7, 2009.

[7] M. Bădoiu, E. D. Demaine, M. Hajiaghayi, and P. Indyk, "Low–Dimensional Embedding with Extra Information," *Disc. Comput. Geom.*, vol. 36, no. 4, pp. 609–632, 2006.

[8] R. Connelly, "Generic Global Rigidity," *Disc. Comput. Geom.*, vol. 33, no. 4, pp. 549–563, 2005.

[9] E. Cuthill and J. McKee, "Reducing the Bandwidth of Sparse Symmetric Matrices," in *Proc. 1969 24th Nat'l Conf.*, 1969, pp. 157–172.

[10] L. Doherty, K. S. J. Pister, and L. El Ghaoui, "Convex Position Estimation in Wireless Sensor Networks," in *IEEE INFOCOM 2001*, vol. 3, 2001, pp. 1655–1663.

[11] T. Eren, D. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. O. Anderson, and P. N. Belhumeur, "Rigidity, Computation, and Randomization in Network Localization," in *IEEE INFOCOM 2004*, vol. 4, 2004, pp. 2673–2684.

[12] Z. Fekete and T. Jordán, "Uniquely Localizable Networks with Few Anchors," in *Proc. ALGOSENSORS 2006*, LNCS 4240, pp. 176–183, 2006.

[13] M. Fukuda, M. Kojima, K. Murota, and K. Nakata, "Exploiting Sparsity in Semidefinite Programming via Matrix Completion I: General Framework," *SIAM J. Opt.*, vol. 11, no. 3, pp. 647–674, 2000.

[14] D. K. Goldenberg, A. Krishnamurthy, W. C. Maness, Y. R. Yang, A. Young, A. S. Morse, A. Savvides, and B. D. O. Anderson, "Network Localization in Partially Localizable Networks," in *IEEE INFOCOM 2005*, vol. 1, 2005, pp. 313–326.

[15] S. J. Gortler, A. D. Healy, and D. P. Thurston, "Characterizing Generic Global Rigidity," 2007, preprint.

[16] J. Graver, B. Servatius, and H. Servatius, *Combinatorial Rigidity*, ser. Graduate Studies in Mathematics. Providence, Rhode Island: American Mathematical Society, 1993, vol. 2.

[17] B. Hendrickson, "Conditions for Unique Graph Realizations," *SIAM J. Comput.*, vol. 21, no. 1, pp. 65–84, 1992.

[18] B. Jackson and T. Jordán, "Connected Rigidity Matroids and Unique Realizations of Graphs," *J. Comb. Theory, Ser. B*, vol. 94, no. 1, pp. 1–29, 2005.

[19] S. Kim, M. Kojima, M. Mevissen, and M. Yamashita, "Exploiting Sparsity in Linear and Nonlinear Matrix Inequalities via Positive Semidefinite Matrix Completion," Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh–Okayama, Meguro–ku, Tokyo 152–8552, Japan, Tech. Rep. B–452, 2009.

[20] S. Kim, M. Kojima, and H. Waki, "Exploiting Sparsity in SDP Relaxation for Sensor Network Localization," *SIAM J. Opt.*, vol. 20, no. 1, pp. 192–215, 2009.

[21] X. Li, *Wireless Ad Hoc and Sensor Networks: Theory and Applications*. New York: Cambridge University Press, 2008.

[22] H. Lim and J. C. Hou, "Distributed Localization for Anisotropic Sensor Networks," *ACM Trans. Sensor Networks*, vol. 5, no. 2, Article 11, 2009.

[23] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) Using AOA," in *IEEE INFOCOM 2003*, vol. 3, 2003, pp. 1734–1743.

[24] N. B. Priyantha, H. Balakrishnan, E. D. Demaine, and S. Teller, "Mobile–Assisted Localization in Wireless Sensor Networks," in *IEEE INFOCOM 2005*, vol. 1, 2005, pp. 172–183.

[25] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic Fine–Grained Localization in Ad–Hoc Networks of Sensors," in *Proc. ACM MobiCom 2001*, 2001, pp. 166–179.

[26] J. B. Saxe, "Embeddability of Weighted Graphs in $k$–Space is Strongly NP–Hard," in *Proc. 17th Allerton Conf. Commun., Control, and Comput.*, 1979, pp. 480–489.

[27] Y. Shang and W. Ruml, "Improved MDS–Based Localization," in *IEEE INFOCOM 2004*, vol. 4, 2004, pp. 2640–2651.

[28] A. M.-C. So and Y. Ye, "A Semidefinite Programming Approach to Tensegrity Theory and Realizability of Graphs," in *Proc. 17th ACM–SIAM SODA*, 2006, pp. 766–775.

[29] ——, "Theory of Semidefinite Programming for Sensor Network Localization," *Math. Prog., Ser. B*, vol. 109, no. 2, pp. 367–384, 2007.

[30] P. Tseng, "Second–Order Cone Programming Relaxation of Sensor Network Localization," *SIAM J. Opt.*, vol. 18, no. 1, pp. 156–185, 2007.

[31] C. Wang and L. Xiao, "Locating Sensors in Concave Areas," in *IEEE INFOCOM 2006*, 2006, pp. 1–12.

[32] Z. Wang, S. Zheng, Y. Ye, and S. Boyd, "Further Relaxations of the Semidefinite Programming Approach to Sensor Network Localization," *SIAM J. Opt.*, vol. 19, no. 2, pp. 655–673, 2008.

[33] Z. Yang, Y. Liu, and X.-Y. Li, "Beyond Trilateration: On the Localizability of Wireless Ad–Hoc Networks," in *IEEE INFOCOM 2009*, 2009, pp. 2392–2400.

[34] Z. Zhu, A. M.-C. So, and Y. Ye, "Universal Rigidity and Edge Sparsification for Sensor Network Localization," 2009, preprint, available at http://www.se.cuhk.edu.hk/~manchoso/papers/unirigid.pdf.