Rigid Network Design Via Submodular Set Function Optimization

Iman Shames and Tyler H. Summers

Abstract—We consider the problem of constructing networks that exhibit desirable algebraic rigidity properties, which can provide significant performance improvements for associated formation shape control and localization tasks. We show that the network design problem can be formulated as a submodular set function optimization problem and propose greedy algorithms that achieve global optimality or an established near-optimality guarantee. We also consider the separate but related problem of selecting anchors for sensor network localization to optimize a metric of the error in the localization solutions. We show that an interesting metric is a modular set function, which allows a globally optimal selection to be obtained using a simple greedy algorithm. The results are illustrated via numerical examples, and we show that the methods scale to problems well beyond the capabilities of current state-of-the-art convex relaxation techniques.

Index Terms—Network problems, optimization

1 Introduction

Accelerating advances in communication, computation, and sensing technologies are allowing networks of low-cost interconnected nodes to provide unprecedented data streams about physical processes, and position control and localization of (mobile sensor) node positions is often required to meaningfully interpret and utilize the data. There are several broad categories of measurement techniques that can be used for position localization and control, including angle of arrival, received signal strength profiling, and distance related measurements. In this paper, we will focus on distance-based scenarios in the plane; however, similar results can be obtained for other types of measurements and in higher dimensions.

The application of rigid graph theory to distanced-based motion control in mobile robot formations and localization problems in sensor networks has been the focus of many recent studies [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]. In formation shape control problems, the literature is focused on characterizing rigid graph properties that allow distributed control of any desired configuration [1], [2], [3] and associated control law design [4], [5], [6]. Similarly, the sensor network localization literature has focused on characterizing rigid graph properties that allow unique localization solutions [7], [8], [10], [12], [13], [14] and associated localization algorithms [9], [10], [11].

While many aspects of distanced-based motion control and localization problems have been studied in detail, very few papers consider design of the network itself. In particular, the distances to be actively controlled in a formation control problem or the distance measurements in a localization problem are typically taken as given. However, a network designer can choose which set of distances to control or which set of measurements to utilize given knowledge of the desired formation shape or estimated sensor configuration. An intelligent choice of these sets may yield drastic performance improvements of certain tasks, such as localization or formation control.

Design of rigid networks to optimize metrics associated with the graph edges is considered in [15], [16], [17], [18]. In [15], the total sum of the edge lengths, which is roughly related to communication cost, is minimized using decentralized methods. A more general state space setting is considered in [16], and an algorithm is developed to minimize the system $\mathcal{H}_2$ norm of the network associated with an exogenous disturbance input. In [17], a sum of generic weight functions associated with edges is minimized using decentralized methods. In all cases, additional edges increase the cost, so the optimal networks structures are shown to be minimally rigid networks (rigid networks with the minimum number of edges), and the algorithms are closely related to standard algorithms for finding minimum weight spanning trees.

It has also been observed that the performance of the rigidity based localization and control techniques are closely related to the algebraic properties of the network [19], [20], [21]. There are combinatorial methods to construct networks to satisfy certain rigidity properties, namely, Henneberg sequences [22], [23], but there are no methods that take into account the algebraic properties of the network, arising from the relative positions of the nodes, to maximize these performance-based objective functions in the network design. The metrics that we will consider are different from those in [15], [16], [17]. In contrast to [15], [16], [17], our metrics are motivated by performance of specific localization and control tasks and are *improved* by the addition of edges. This implies that minimally rigid networks are not necessarily optimal in our setting. An interesting extension for

- *I. Shames is with the Department of Electrical and Electronic Engineering, University of Melbourne, Australia. E-mail: iman.shames@unimelb.edu.au.*
- *T. H. Summers is with the Department of Mechanical Engineering, University of Texas at Dallas. E-mail: tyler.summers@utdallas.edu.*

Manuscript received 16 July 2014; revised 2 Sept. 2015; accepted 13 Sept. 2015. Date of publication 17 Sept. 2015; date of current version 13 Oct. 2015. Recommended for acceptance by C. Caramanis.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TNSE.2015.2480247

future work would be to simultaneously consider both types of objectives and study the corresponding tradeoffs. The recent work [18] formulates a related edge weight maximization problem but does not make connections to algebraic rigidity properties.

A related network design problem in the context of sensor network position localization is anchor selection. It is well known that the absolute positions of at least three sensors, called *anchors*, are required to uniquely localize a network in the plane. However, there are typically many possible choices for anchors, each of which results in different properties of the position estimates of the remaining sensors. The optimal selection of anchors has been considered in [19], which uses a convex relaxation heuristic to minimize a worst-case measure of estimation error covariance. While this method results in a convex optimization problem, there are no approximation guarantees, and the convex optimization problem can still be difficult to solve for very large networks. The problem of anchor selection or placement in sensor networks has been studied in different context in the last few years, however, to the best of our knowledge none of the existing methods have taken the approach presented in this paper and often are not applicable to large scale networks, e.g., see [24], [25], [26], [27], [28].

In this paper we first consider the problem of rigid network design where in a given configuration the edges are selected to optimize an algebraic rigidity-related performance index, based on a *rigidity Gramian*, while satisfying a rank condition that ensures rigidity of the network. In particular, we show that networks with desirable algebraic rigidity properties can be constructed using a simple greedy algorithm. Moreover, we establish that certain cost functions that capture the algebraic properties of the networks are modular or submodular set functions, which allows us to provably obtain globally optimal or near optimal edge selections. We then revisit the problem of optimal anchor selection that was introduced in [19] and propose an alternative solution based on set function optimization. In particular, we show that an interesting metric associated with the estimation error covariance is a modular set function. Again, the implication of this is that a simple greedy algorithm for anchor selection will provide a globally optimal anchor selection. The results are illustrated with numerical examples, and we show that our methods scale to problems well beyond the capabilities of current state-of-the-art convex relaxation techniques. The problems have a similar mathematical structure to other recently introduced Gramian-based set function optimization problems linking submodularity with controllability [29], [30], [31] and submodularity with network coherence [32]. Other problems involving submodularity in networked control systems are studied in [33].

The rest of the paper is organized as follows. Section 2 describes the necessary background information. In Section 3 the problem of constructing a rigid network through applying submodular optimization techniques is considered. In Section 4 we revisit the problem of anchor selection in a network in the context of network localization. We present numerical examples in Section 5. Concluding remarks are given in Section 6.

2 Preliminaries

This section provides background on rigid graph theory and on set function submodularity and matroids. We introduce an important matrix, called the *rigidity Gramian*, which is constructed from the well-known rigidity matrix and quantifies algebraic rigidity properties of a network. The network design problems we consider in the following section can be cast as matroid constrained submodular set function optimization problems.

2.1 Rigid Graph Theory

Let us call a *network* a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the vertex set and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set with $\{i, j\}$ denoting the undirected edge incident at i and j, together with a map $\mathbf{p} : \mathcal{V} \rightarrow \mathbb{R}^{2|\mathcal{V}|}$, with $p_i \in \mathbb{R}^2$ denoting the coordinate vector associated with vertex $i \in \mathcal{V}$. A network is thus denoted by a tuple $(\mathcal{G}, \mathbf{p})$. Suppose there is a set of non-negative real numbers representing intersensor distances $\mathcal{D} = \{d_{ij} : \{i, j\} \in \mathcal{E}\}$. The network is a *realization* of $\mathcal{D}$ if $||p_i - p_j|| = d_{ij}$ for any $\{i, j\} \in \mathcal{E}$.

The two networks $(\mathcal{G}, \mathbf{p})$ and $(\mathcal{G}, \mathbf{q})$ are *equivalent* if $||p_i - p_j|| = ||q_i - q_j||$ for any $\{i, j\} \in \mathcal{E}$. The two networks $(\mathcal{G}, \mathbf{p})$ and $(\mathcal{G}, \mathbf{q})$ are *congruent* if $||p_i - p_j|| = ||q_i - q_j||$ for all pairs i, j whether or not $\{i, j\} \in \mathcal{E}$. This is equivalent to saying that $(\mathcal{G}, \mathbf{p})$ can be obtained from $(\mathcal{G}, \mathbf{q})$ by an isometry of $\mathbb{R}^2$, i.e., a combination of translation, rotation and reflection.

Roughly speaking, a network is rigid when it cannot flex, i.e., its shape cannot be changed via continuous motions of vertex positions while keeping distances associated with edges constant to become incongruent to its starting position. More precisely, we have the following definition.

Definition 1. *A network $(\mathcal{G}, \mathbf{p})$ is rigid if there exists some positive ϵ such that if $(\mathcal{G}, \mathbf{p})$ and $(\mathcal{G}, \mathbf{q})$ are equivalent and $||p_i - q_i|| < \epsilon$ for all $i \in \mathcal{V}$, then the two networks are congruent.*

A network is *minimally rigid* when it is rigid but the deletion of any single edge from the associated graph results in a nonrigid network. The underlying graph of any minimally rigid network can be constructed via applying the following two operations, called *Henneberg sequences* [23], to a rigid graph on two vertices (viz., two vertices connected by an edge, the smallest rigid graph):

1) (Vertex Addition) Addition of a new vertex to the graph along with edges connecting it to two previously existing vertices.
2) (Edge Splitting) Addition of a new vertex to the graph along with two edges connecting it to two previously existing vertices that share a common edge, removing the common edge, and addition of another edge to any other vertex in the graph.

Any non-minimally rigid graph can be obtained from a minimally rigid graph by simply adding edges. An equivalent graph theoretic characterization is given by Laman's Theorem [34], which states roughly that rigid graphs have at least $2|V| - 3$ well-distributed edges.

It turns out that there exist rigid networks $(\mathcal{G}, \mathbf{p})$ and $(\mathcal{G}, \mathbf{q})$ which are equivalent but not congruent. In fact, any minimally rigid network with more than three vertices is equivalent to another such network to which it is not congruent, due to so-called *flip* and *flex ambiguities* [20]. A network $(\mathcal{G}, \mathbf{p})$ is *globally rigid* when every network equivalent to $(\mathcal{G}, \mathbf{p})$ is also congruent to it. Such a network is uniquely realisable given distances associated with the edges, and fixing the coordinates of at least three noncollinear vertices results in a unique position map $\mathbf{p}$ that satisfies the given distance constraints; see [12], [13], [14] for more details.

Rigidity and global rigidity for a network in $\mathbb{R}^2$ are generic properties, in the sense that if a network $(\mathcal{G}, \mathbf{p})$ has either of these properties, then the network $(\mathcal{G}, \bar{\mathbf{p}})$ will also have the property for generic values of the position coordinates $\bar{\mathbf{p}}$, i.e., for all values save possibly for those contained in a set of measure zero involving an algebraic dependence over the rationals of the coordinates.[1] However, the positions in a realization of a network, not just the underlying graph, can significantly affect the performance of algorithms for robotic formation shape control and sensor localization that relate to rigidity, so algebraic properties of rigidity are of substantial interest.

Rigidity can also be fully characterized algebraically. Consider a network $(\mathcal{G}, \mathbf{p})$ in the plane, and let the coordinate vector p_j of vertex j be $p_j = [x_j, y_j]^\top$. The rigidity matrix, denoted $R_{(\mathcal{G},\mathbf{p})}$ is defined with an arbitrary ordering of the vertices and edges, and has $2|\mathcal{V}|$ columns and $|\mathcal{E}|$ rows. Each edge gives rise to a row, and if the edge links vertices j and κ, the nonzero entries of the row of the matrix are in columns $2j-1$, $2j$, $2k-1$ and $2k$, and are respectively $x_j - x_k$, $y_j - y_k$, $x_k - x_j$, $y_k - y_j$. Note that the entries of the rigidity matrix do not depend on the absolute value of the positions of the nodes, and they only depend on the relative positions of the nodes from each other. A graph is generically rigid if and only if for generic vertex positions, the rigidity matrix has rank $2|\mathcal{V}| - 3$ [23]. There are zero-measure sets of non-generic vertex positions for which the rigidity matrix can be rank deficient, leading to the separate concept of *infinitesimal rigidity* [35]. In this paper, we assume that vertices are in generic positions to avoid degenerate situations in subsequent optimization problems involving the subtle distinction between generic and infinitesimal rigidity.

Assumption 1. *The positions associated with vertices are generic.*

The rigidity matrix contains much more quantitative information about rigidity than just the rank. In particular, the singular values of the rigidity matrix provide a measure of the algebraic quality of a network. Accordingly, we define the *edge rigidity Gramian* for a network $(\mathcal{G}, \mathbf{p})$ as

$$X_{\mathcal{E}}^{(\mathcal{G},\mathbf{p})} = R_{(\mathcal{G},\mathbf{p})} R_{(\mathcal{G},\mathbf{p})}^\top \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|} \tag{1}$$

and the *vertex rigidity Gramian* as

$$X_{\mathcal{V}}^{(\mathcal{G},\mathbf{p})} = R_{(\mathcal{G},\mathbf{p})}^\top R_{(\mathcal{G},\mathbf{p})} \in \mathbb{R}^{|2\mathcal{V}|\times 2|\mathcal{V}|}. \tag{2}$$

Networks whose Gramians have large eigenvalues have superior algebraic rigidity properties than those with smaller eigenvalues. These advantages translate to faster local convergence rates in formation shape control problems and lower estimation error covariance in localization problems. Note that the vertex and edge Gramians have the same non-zero eigenvalues, but that they have different null space dimension and different eigenvectors. Various scalarizations of the Gramian can be considered, which trade off in different ways rigidity across the network.

2.2 Submodularity and Matroids

We will formulate rigid network design problems as set function optimization problems. For a given finite set $\mathcal{V} = \{1, ..., N\}$, which will represent a set of edges or nodes in a network, a set function $f : 2^{\mathcal{V}} \to \mathbf{R}$ assigns a real number to each subset of V, which will represent a scalar metric of the network rigidity. Cardinality constrained set function optimization problems have the form

$$\underset{\mathcal{S}\subseteq\mathcal{V},\ |\mathcal{S}|\leq\kappa}{\text{maximize}} \quad f(\mathcal{S}). \tag{3}$$

This finite combinatorial optimization problem can by solved by brute force by evaluating f all possible subsets of size κ and selecting the maximizing subset. However, this approach quickly becomes infeasible even for moderate values of N and κ. When f has a property called *submodularity*, although the problem remains hard, a simple greedy algorithm can be used to obtain near optimal subsets.

Definition 2 (Submodularity). *Let $\mathcal{V}$ be a finite set and let $f : 2^{\mathcal{V}} \to \mathbb{R}$ be a set function on $\mathcal{V}$. Then f is called submodular if for every $\mathcal{A}, \mathcal{B} \subseteq \mathcal{V}$ it holds that*

$$f(\mathcal{A}) + f(\mathcal{B}) \geq f(\mathcal{A}\cup\mathcal{B}) + f(\mathcal{A}\cap\mathcal{B}). \tag{4}$$

Submodularity can be informally described as a diminishing returns property; that is, adding an element to a smaller set gives a larger gain than adding one to a larger set. In particular, we have the following definition and result.

Definition 3 (Set Function Monotonicity). *A set function $f : 2^{\mathcal{V}} \to \mathbb{R}$ is called monotone increasing if for all subsets $\mathcal{A}, \mathcal{B} \subseteq \mathcal{V}$ it holds that*

$$\mathcal{A}\subseteq\mathcal{B} \Rightarrow f(\mathcal{A}) \leq f(\mathcal{B}) \tag{5}$$

and is called monotone decreasing if for all subsets $\mathcal{A}, \mathcal{B} \subseteq \mathcal{V}$ it holds that

$$\mathcal{A}\subseteq\mathcal{B} \Rightarrow f(\mathcal{A}) \geq f(\mathcal{B}). \tag{6}$$

Theorem 1 ([36]). *A set function $f : 2^{\mathcal{V}} \to \mathbb{R}$ is submodular if and only if the derived set functions $f_a : 2^{\mathcal{V}\backslash\{a\}} \to \mathbb{R}$*

$$f_a(\mathcal{X}) = f(\mathcal{X}\cup\{a\}) - f(\mathcal{X})$$

are monotone decreasing for all $a \in \mathcal{V}$.

A set function is called *supermodular* if the reversed inequality in (4) holds, and is called *modular* if it is both sub- and supermodular. Modular functions have the following simple characterization.

Definition 4 (Modularity). *A function f is modular if for any $\mathcal{A} \subseteq \mathcal{V}$:*

[1] An example for such non-generic situations is the case where the coordinates are collinear in $\mathbb{R}^2$.

$$f(\mathcal{A}) = g(\emptyset) + \sum_{i\in\mathcal{A}} g(i). \tag{7}$$

One can see that optimizing modular set functions is easy because each element of a subset gives an independent contribution to the function values. Thus, (3) is solved by evaluating the set function for each individual element and choosing the top κ individual elements to obtain the best size κ subset.

Consider the submodular function optimization problem (3) where f is a monotone increasing submodular function, κ is a constant, and $\mathcal{V}$ is a given set. Algorithm 1 outlines a greedy algorithm to this problem. We have the following approximation result for (3) and the greedy Algorithm 1.

Theorem 2 ([37]). *Algorithm 1 gives a $(1-1/e)$-approximation for the problem (3), i.e., $(1-1/e)f(\mathcal{S}_{OPT}) \leq f(\mathcal{S}^\star)$, where $\mathcal{S}_{OPT}$ is the global optimizer of (3) and $\mathcal{S}^\star$ is the output of Algorithm 1.*

Algorithm 1. A Greedy Solution to (3)

```
$\mathcal{S} \leftarrow \emptyset$
while $|\mathcal{S}| \leq \kappa$ do
    $e^\star = \mathrm{argmax}_{e\in\mathcal{V}\setminus\mathcal{S}} \quad [f(\mathcal{S}\cup\{e\}) - f(\mathcal{S})]$
    $\mathcal{S} \leftarrow \mathcal{S}\cup\{e^\star\}$
end while
$\mathcal{S}^\star \leftarrow \mathcal{S}$
```

Cardinality constrained set function optimization problems can be considered as a special case of more general matroid constrained set function optimization problems. Matroids can be used to encode more complicated constraints: for example, in our setting one might like to construct a network to optimize an algebraic metric of rigidity subject to a constraint that the underlying graph is rigid, which can be described as a matroid constraint [14]. Matroids generalize the notion of linear independence in vector spaces and are defined as follows.

Definition 5 (Matroid). *A matroid is an ordered pair $(\mathcal{V}, \mathcal{I})$ consisting of a finite set $\mathcal{V}$ and a collection $\mathcal{I}$ of subsets of $\mathcal{V}$ (called the independent sets) having the following properties:*

1) $\emptyset \in \mathcal{I}$
2) *If $\mathcal{A} \in \mathcal{I}$ and $\mathcal{B} \subset \mathcal{A}$, then $\mathcal{B} \in \mathcal{I}$.*
3) *If $\mathcal{A}$ and $\mathcal{B}$ are in $\mathcal{I}$ and $|\mathcal{A}| < |\mathcal{B}|$, then there is an element x of $\mathcal{B} \setminus \mathcal{A}$ such that $\mathcal{A}\cup\{x\} \in \mathcal{I}$.*

Definition 6 (Independence Oracle). *For the matroid $(\mathcal{V}, \mathcal{I})$, an independence oracle is a function $\Psi(\mathcal{X})$ for all $\mathcal{X} \subseteq \mathcal{V}$ such that:*

$$\Psi(\mathcal{X}) = \begin{cases} \text{True} & \text{if } \mathcal{X} \in \mathcal{I} \\ \text{False} & \text{if } \mathcal{X} \notin \mathcal{I}. \end{cases}$$

Now consider the following matroid constrained optimization problem:

$$\underset{\mathcal{S}\subseteq\mathcal{V},\ \mathcal{S}\in\mathcal{I}}{\text{maximize}} \quad f(\mathcal{S}), \tag{8}$$

where the ordered pair $(\mathcal{V}, \mathcal{I})$ is a matroid. Similar to above, Algorithm 2 provides a greedy solution to this problem and the following result holds for the solution obtained from it.

Theorem 3 ([38]). *Algorithm 2 gives a $1/2$-approximation for the problem (8), i.e., $f(\mathcal{S}_{OPT})/2 \leq f(\mathcal{S}^\star)$ where $\mathcal{S}_{OPT}$ is the global optimizer of (8) and $\mathcal{S}^\star$ is the output of Algorithm 2.*

Observe that calculating $\mathcal{O}$ at each iteration requires access to an independent oracle. Moreover, we note that, there are recently developed and more sophisticated algorithms that improve the performance guarantee to $(1-1/e)$ $f(\mathcal{S}_{OPT}) \leq f(\mathcal{S}^\star)$, the same approximation guarantee that can be achieved for the special case of a cardinality constraints [39]. However, in this paper we focus on simple greedy algorithms of the type presented in Algorithm 2 to simplify the presentation.

Algorithm 2. A Greedy Solution to (8)

```
Require: An independence oracle checking $e \in \mathcal{I}$.
    $\mathcal{S} \leftarrow \emptyset$
    $\mathcal{O} \leftarrow \{e \in \mathcal{V}\setminus\mathcal{S} : \mathcal{S}\cup\{e\} \in \mathcal{I}\}$
    while $\mathcal{O} \neq \emptyset$ do
        $e^\star = \mathrm{argmax}_{e\in\mathcal{O}} \quad [f(\mathcal{S}\cup\{e\}) - f(\mathcal{S})]$
        $\mathcal{S} \leftarrow \mathcal{S}\cup\{e^\star\}$
        $\mathcal{O} \leftarrow \{e \in \mathcal{V}\setminus\mathcal{S} : \mathcal{S}\cup\{e\} \in \mathcal{I}\}$
    end while
    $\mathcal{S}^\star \leftarrow \mathcal{S}$
```

We conclude this section by emphasizing that for the case that f is a modular function both algorithms result globally optimal solutions such that $f(\mathcal{S}_{OPT}) = f(\mathcal{S}^\star)$.

3 Rigid Network Construction

In this section, we consider the problem of constructing rigid networks given a set of vertices and their positions in the plane by choosing which pairs of nodes should be connected by an edge. We focus on optimizing a performance index that captures the algebraic quality of the rigidity of the network (though other cost functions associated with the selected edges can be incorporated). Since rigidity is monotonic in the number of edges, we limit the number of edges in the network, subject to a constraint that the underlying graph is rigid; from Laman's theorem [34] it is necessary that $|\mathcal{E}| \geq 2|\mathcal{V}| - 3$. As mentioned earlier, there are many sets of edges that could be added to achieve rigidity of the underlying graph (namely, any set obtained from a Henneberg sequence [23]), but the algebraic rigidity of the resulting networks can be drastically different. To illustrate this, we present two examples and explain the implications with respect to localization and formation shape control scenarios.

3.1 Motivating Examples

Example 1. Consider Fig. 1. Both networks depicted in Fig. 1 are constructed by the application of the Henneberg sequence and as a result are generically minimally rigid. Now, consider Fig. 2, where the networks have the same underlying graph as those in Fig. 1 with the difference that nodes 1 and 4 are nearly collocated. In the new configuration, the network depicted on the left exhibits weak

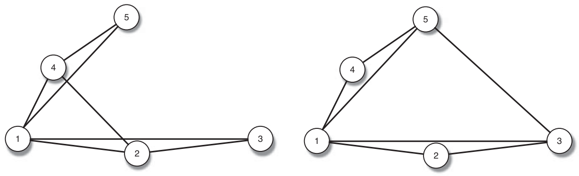

Fig. 1. Two different networks obtained from the application of the Henneberg sequence. Hence, the underlying graphs are generically rigid.

algebraic rigidity properties: one can see that if node 4 is placed on top of node 1, node 5 can take infinitely many positions satisfying the edge length constraints $\|p_1 - p_5\| = d_{15}$ and $\|p_4 - p_5\| = d_{45}$. This corresponds to a rank deficiency in the rigidity Gramian. If the goal was to localize the nodes given noisy distance measurements, the variance of the position estimate for node 5 would approach infinity as the distance between nodes 1 and 4 goes to zero. The network on the right does not suffer from this problem even if nodes 1 and 4 are collocated. In terms of algebraic rigidity, the graph structure on the right is highly preferred given the positions in Fig. 2.

Example 2. In this example we again consider the networks depicted in Fig. 1. Suppose that the nodes now represent mobile robots and it is desired for the robots to move to a formation such that $\|p_i - p_j\| = d_{ij}$ for some desired d_{ij} and $i, j \in \mathcal{V}$. Moreover, suppose that the nodes have single integrator dynamics $\dot{p}_i = u_i$ and to achieve the desired formation, the input of each node u_i is chosen to be the control law proposed in [4], yielding the closed-loop dynamics

$$\dot{p}_i = \frac{1}{2} \sum_{j \in \mathcal{V}_i} \left(d_{ij}^2 - \|p_j - p_i\|^2\right)(p_j - p_i), \tag{9}$$

where $\mathcal{V}_i$ is the set of nodes sharing an edge with i. As in the previous example, consider a situation where d_{14} is small. Fig. 3 depicts the control performance for the left and right networks in Fig. 1. The plot shows the time evolution of the shape error function $s_e = \sum_{i,j \in \mathcal{V}} |d_{ij} - \|p_j - p_i\||$, which captures convergence of the nodes to the desired configuration starting from the same initial positions. The error decreases much faster when the nodes have the underlying graph in the right network in Fig. 1.

The difference in performance in both examples can be traced to the eigenvalues of the rigidity Gramian. In particular, the eigenvalues of the Gramian on the right network in Fig. 1 are larger than those of the left network, and this leads to better performance in both localization and formation shape control scenarios. These examples motivate the consideration of node coordinates and associated algebraic rigidity properties in constructing rigid networks.

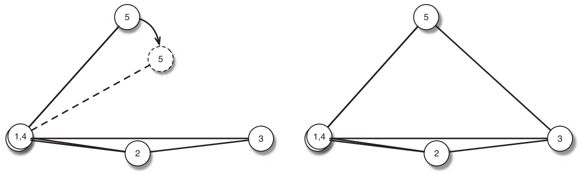

Fig. 2. Although the underlying graphs are generically rigid, the relative node positions can lead to significantly different algebraic rigidity properties. The network on the left ceases to be rigid when nodes 1 and 4 become collocated (and have very low algebraic rigidity when nearly collocated). Thus, the network on the right is highly preferred for these particular relative positions.

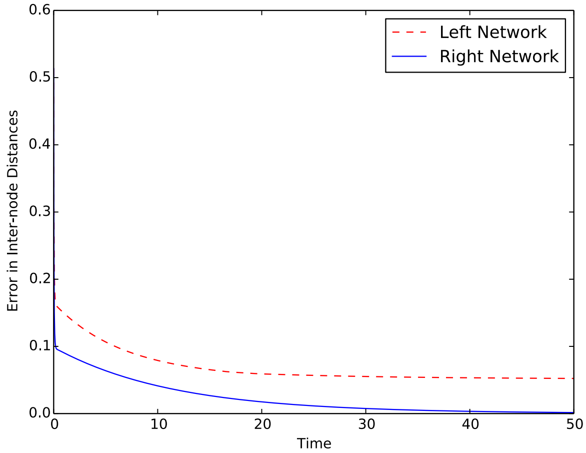

Fig. 3. The performance of a formation shape control strategy for two different networks.

3.2 Problem Statement

The following statement formalizes the problem of interest.

Problem 1. *Given a finite set of nodes $\mathcal{V}$ and a generic coordinate mapping $\mathbf{p} \in \mathbb{R}^{2|\mathcal{V}|}$, solve the following optimization problem*

$$\begin{array}{ll} \underset{\mathcal{E}}{\text{maximize}} & f(\mathcal{E}) \\ \text{subject to} & (\mathcal{G}, \mathbf{p}) \text{ is rigid} \\ & \mathcal{G} = (\mathcal{V}, \mathcal{E}), \quad |\mathcal{E}| \leq \kappa, \end{array} \tag{10}$$

where $\mathcal{G}$ is the underlying graph with vertex set $\mathcal{V}$ and the variable edge set $\mathcal{E}$, $\mathbf{p}$ corresponds to the nodes coordinates, $f : 2^{\mathcal{V}} \rightarrow \mathbb{R}$ is a monotone increasing set function that quantifies the algebraic rigidity of the network, and $\kappa \geq 2|\mathcal{V}| - 3$ is a given constant.

This problem is an NP-hard combinatorial optimization problem. However, we will show that several scalar functions of the rigidity Gramian quantifying algebraic rigidity are modular or submodular set functions. The problem can be split into two stages. In the first stage, a minimally rigid graph is constructed while optimizing an algebraic rigidity metric, which can be cast as a matroid constrained submodular maximization problem. In the second stage, after minimal rigidity has been achieved, an algebraic rigidity metric is further optimized while the remaining edges are added, which can be cast as a cardinality constrained submodular maximization problem. As a consequence of the modularity and submodularity properties, each stage comes with global optimality or near optimality guarantees per Theorems 2 and 3 via simple greedy algorithms.

First, we rewrite the rigidity constraint of the network $(\mathcal{G}, \mathbf{p})$ in terms of the rank of the associated rigidity matrix $R_{(\mathcal{G},\mathbf{p})}$. Thus (10) can be rewritten as

$$\begin{aligned} \underset{\mathcal{E}}{\text{maximize}} \quad & f(\mathcal{E}) \\ \text{subject to} \quad & \text{rank}(R_{(\mathcal{G},\mathbf{p})}) = 2|\mathcal{V}|-3 \\ & \mathcal{G}=(\mathcal{V},\mathcal{E}), \quad |\mathcal{E}|\le\kappa. \end{aligned} \tag{11}$$

The proposed solution to (11) consists of two stages. In the first stage, we consider a special case of (10), where $(\mathcal{G},\mathbf{p})$ is required to be minimally rigid, in which case $|\mathcal{E}|=2|\mathcal{V}|-3$:

$$\begin{aligned} \underset{\mathcal{E}}{\text{maximize}} \quad & f(\mathcal{E}) \\ \text{subject to} \quad & \text{rank}(R_{(\mathcal{G},\mathbf{p})}) = 2|\mathcal{V}|-3 \\ & \mathcal{G}=(\mathcal{V},\mathcal{E}), \quad |\mathcal{E}| = 2|\mathcal{V}|-3. \end{aligned} \tag{12}$$

After a minimally rigid graph has beed constructed, the rank constraint is unnecessary. So in the second stage, the remaining $\kappa-(2|\mathcal{V}|-3)$ edges are added by solving

$$\begin{aligned} \underset{\mathcal{E}}{\text{maximize}} \quad & f(\mathcal{E}) \\ \text{subject to} \quad & \mathcal{E}_{S1,OPT}\subset\mathcal{E} \\ & \mathcal{G}=(\mathcal{V},\mathcal{E}), \quad |\mathcal{E}|\le\kappa, \end{aligned} \tag{13}$$

where $\mathcal{E}_{S1,OPT}$ is the result of the algorithm for (12). In what follows, we elaborate on our proposed algorithms for solving (12) and (13).

3.3 Solution to the First Stage as a Matroid Constrained Optimization Problem

Let

$$\begin{aligned} \overline{\mathcal{M}} \triangleq \{M : M\subset\mathcal{E}_c, |M| = 2|\mathcal{V}|-3, \mathcal{G}_M \\ = (\mathcal{V},M), \text{rank}(R(\mathcal{G}_M,\mathbf{p})) = 2|\mathcal{V}|-3\}, \end{aligned} \tag{14}$$

where $\mathcal{E}_c$ is the set of the edges of the complete graph on $\mathcal{V}$, and let

$$\mathcal{F}\triangleq\bigcup_{\mathcal{M}\in\overline{\mathcal{M}}} 2^{\mathcal{M}}. \tag{15}$$

We have the following.

Proposition 1. *Consider the following optimization problem:*

$$\underset{\mathcal{E}\in\mathcal{F}}{\text{maximize}} \quad f(\mathcal{E}) \tag{16}$$

where $\mathcal{F}$ is described by (15). The following statements hold:

1) *The pair $(\mathcal{E}_c,\mathcal{F})$ is a matroid.*
2) *Any solution to (12) is a solution to (16), and vice versa.*

Proof. To prove 1), first label the edges of $\mathcal{E}_c$ from 1 to $|\mathcal{V}|(|\mathcal{V}|-1)/2$. Next, consider the mapping $\eta:\mathcal{C}\mapsto\mathcal{E}_c$ that relates each of the numbers to its corresponding edge, where $\mathcal{C}=\{1,\dots,|\mathcal{V}|(|\mathcal{V}|-1)/2\}$. Furthermore, let $S(\mathcal{V},\mathbf{p})$ be the rigidity matrix associated with network $(\mathcal{V},\mathcal{E}_c,\mathbf{p})$ with s_i its ith row. By definition (15), $\mathcal{F}$ is the set of all subsets $\mathcal{Y}$ of $\mathcal{E}_c$ such that $s_{\eta(e)}$, $\forall e\in\mathcal{Y}$, are linearly independent. Then according to [40, Proposition 1.1.1] the pair $(\mathcal{E}_c,\mathcal{F})$ is a matroid.

To prove 2), let $\widetilde{\mathcal{E}}$ and $\widehat{\mathcal{E}}$ be the solutions to (12) and (16) respectively where $|\widetilde{\mathcal{E}}|\le 2|\mathcal{V}|-3$ and $|\widehat{\mathcal{E}}|\le 2|\mathcal{V}|-3$. First, because the rank condition cannot be satisfied with $|\widetilde{\mathcal{E}}|<2|\mathcal{V}|-3$ then $|\widetilde{\mathcal{E}}|=2|\mathcal{V}|-3$ and since f is nondecreasing increasing then it is required that $|\widehat{\mathcal{E}}|=2|\mathcal{V}|-3$. This in turn means $\widehat{\mathcal{E}}\in\overline{\mathcal{M}}$ and equivalently $\text{rank}(R((\mathcal{V}, \widehat{\mathcal{E}}),\mathbf{p}) = 2|\mathcal{V}|-3$, thus $\widehat{\mathcal{E}}$ is a solution to (12). Now consider $\widetilde{\mathcal{E}}$. Since $\text{rank}(R(\widetilde{V},\widehat{\mathcal{E}}),\mathbf{p}) = 2|\mathcal{V}|-3$ and $|\widetilde{\mathcal{E}}|=2|\mathcal{V}|-3$, by definition $\widetilde{\mathcal{E}}\in\overline{\mathcal{M}}$ and consequently $\widetilde{\mathcal{E}}\in\mathcal{F}$. Since any member of $\mathcal{F}$ that maximizes f is a solution to (16), $\widetilde{\mathcal{E}}$ is a solution to (16) as well. □

The matroid $(\mathcal{E}_c,\mathcal{F})$ is closely related to the infinitesimal rigidity matroid discussed in [41].

A greedy algorithm for solving the matroid constrained optimization problem (16) similar to the one outlined in Algorithm 2 can be employed where $\mathcal{V}$, $\mathcal{S}$ and $\mathcal{I}$ are replaced by $\mathcal{E}_c$, $\mathcal{E}$, and $\mathcal{F}$, respectively. The algorithm starts with an empty set of edges and iteratively adds edges to the set to maximize the marginal benefit of each added edge given the previously added edges, while maintaining independence of the set according to the matroid $\mathcal{F}$.

Note that constructing $\mathcal{O}$ in Algorithm 2 requires access to an independence oracle for testing independence of the edge set at each iteration. The following result defines such an oracle based on checking the rank of the associated rigidity matrices.

Proposition 2. *For a given $\mathcal{E}\in\mathcal{F}$ with $|\mathcal{E}|<2|\mathcal{V}|-3$, let $\mathcal{O}\triangleq\{e\in\mathcal{E}_c\setminus\mathcal{E} : \mathcal{E}\cup\{e\}\in\mathcal{F}\}$ and $\Omega\triangleq\{e\in\mathcal{E}_c\setminus\mathcal{E} : \text{rank}(R((\mathcal{V},\mathcal{E}\cup\{e\}),\mathbf{p})) > \text{rank}(R_{((\mathcal{V},\mathcal{E}),\mathbf{p})})\}$, where $\mathcal{E}_c$ is the set of the edges of the complete graph on $\mathcal{V}$. Then, $\Omega=\mathcal{O}\in\mathcal{F}$.*

Proof. First, note that for any $\mathcal{E}\in\mathcal{F}$, $|\mathcal{E}|=\text{rank}(R_{((\mathcal{V},\mathcal{E}),\mathbf{p})})$ by definition. Consequently, if for some $e\in\mathcal{E}_c\setminus\mathcal{E}$, $\mathcal{E}\cup\{e\}\in\mathcal{F}$, then $\text{rank}(R((\mathcal{V},\mathcal{E}\cup\{e\}),\mathbf{p})) = |\mathcal{E}|+1 > \text{rank}(R_{((\mathcal{V},\mathcal{E}),\mathbf{p})})\}$. Thus, for any $e\in\mathcal{O}$, also $e\in\Omega$. The proof for the reverse direction follows in a similar way. □

In the light of Proposition 2, we propose Algorithm 3, which makes independence oracle calls explicit by using the rank of the rigidity matrix.

We now present the main result of this section on the trace of the rigidity Gramian, which is one important way of quantifying algebraic rigidity of a network.

Algorithm 3. Minimally Rigid Network Construction: A Greedy Solution to (16)

```
Require: $\mathcal{V}$, $\mathbf{p}$
  $\mathcal{E} \leftarrow \emptyset$
  $\overline{\mathcal{E}} \leftarrow \mathcal{E}_c$
  while rank$\left[R_{((\mathcal{V},\mathcal{E}),\mathbf{p})}\right] < 2|\mathcal{V}|-3$ do
    $e^\star = \text{argmax}_{e\in\overline{\mathcal{E}}} \; [f(\mathcal{E}\cup\{e\}) - f(\mathcal{E})]$
    if rank$\left[R_{((\mathcal{V},\mathcal{E}),\mathbf{p})}\right] <$ rank$\left[R((\mathcal{V},\mathcal{E}\cup\{e^\star\}),\mathbf{p})\right]$ then
      $\mathcal{E} \leftarrow \mathcal{E}\cup\{e^\star\}$
    end if
    $\overline{\mathcal{E}} \leftarrow \overline{\mathcal{E}}\setminus\{e^\star\}$
  end while
  $\mathcal{E}_{S1} \leftarrow \mathcal{E}$
```

Theorem 4. *Let $\mathcal{E}\subseteq\mathcal{E}_c$ and let $X_{\mathcal{E}} = R^\top_{((\mathcal{V},\mathcal{E}),\mathbf{p})}R_{((\mathcal{V},\mathcal{E}),\mathbf{p})}$ be the rigidity Gramian associated with $\mathcal{E}$. Then the set function defined by $f(\mathcal{E}) = \text{trace}(X_{\mathcal{E}})$ is modular and monotone* *increasing. As a consequence, when Algorithm 3 is used to optimize this function, a globally optimal set of edges is obtained.*

Proof. Recall the labeling of edges in $\mathcal{E}_c$ from 1 to $|\mathcal{V}|(|\mathcal{V}|-1)/2$ and the mapping $\eta : \mathcal{C} \to \mathcal{E}_c$ that relates each of the numbers to its corresponding edge, where $\mathcal{C} = \{1, \ldots, |\mathcal{V}|(|\mathcal{V}|-1)/2\}$. Furthermore, let $R((\mathcal{V}, \mathcal{E}_c), \mathbf{p})$ be the rigidity matrix associated with network $((\mathcal{V}, \mathcal{E}_c), \mathbf{p})$ with s_i its ith row. Note that

$$\begin{aligned} f(\mathcal{E}) &= \operatorname{trace}(R^\top_{((\mathcal{V},\mathcal{E}),\mathbf{p})} R_{((\mathcal{V},\mathcal{E}),\mathbf{p})}) \\ &= \operatorname{trace}\left(\sum_{\eta(i)\in\mathcal{E}} s_i^\top s_i\right) = \sum_{\eta(i)\in\mathcal{E}} s_i s_i^\top. \end{aligned}$$

Thus, from Definition 4, $f(\mathcal{E}) = \operatorname{trace}(X_\mathcal{E})$ is modular. That $f(\mathcal{E})$ is monotone increasing follows from the additivity of the Gramian in the edges, which means that adding an edge to the graph cannot reduce the algebraic rigidity of the network. Global optimality is obtained from Algorithm 3 since each edge contributes independently to the function value for any given set. □

For the case that f is modular it is easy to see that $f(\mathcal{E} \cup \{e\}) - f(\mathcal{E}) = f(\{i\})$. As a result, $e^\star = \operatorname{argmax}_{e\in\overline{\mathcal{E}}} [f(\mathcal{E} \cup \{e\}) - f(\mathcal{E})]$ in Algorithm 3 can be replaced by $e^\star = \operatorname{argmax}_{e\in\overline{\mathcal{E}}} \quad f(\{i\})$. This results in a more computationally efficient algorithm since f is evaluated at most $|\mathcal{E}_c|$ times.

If f is submodular, the set $\mathcal{E}_{S1}$ obtained from Algorithm 3 satisfies the performance guarantee $f(\mathcal{E}_{S1,OPT})/2 \leq f(\mathcal{E}_{S1})$, where $\mathcal{E}_{S1,OPT}$ is the global maximizer of f in the first stage problem (12) [37]. One can imagine alternative spectral functions of the rigidity Gramian to optimize, such as trace of the pseudoinverse or the product of non-zero eigenvalues. Since in Algorithm 3 the rigidity matrix changes rank at each iteration, it is not clear how to prove if these or other functions are submodular. However, we will see in the next section that in the second stage we can prove submodularity of other functions.

3.4 Solution to the Second Stage as a Cardinality Constrained Optimization Problem

In the first stage, a minimally rigid graph is constructed by Algorithm 3, which returns edge set $\mathcal{E}_{S1}$. In the second stage the optimization problem (13) is to be solved, in which the remaining edges are added to $\mathcal{E}_{S1}$. A greedy algorithm for solving (13) is outlined in Algorithm 4.

Algorithm 4. Minimally Rigid Network Construction: A Greedy Solution to (13)

```
Require: $\mathcal{V}$, $\mathbf{p}$, $\mathcal{E}_{S1}$ from Algorithm 3.
  $\mathcal{E} \leftarrow \mathcal{E}_{S1}$
  $\overline{\mathcal{E}} \leftarrow \mathcal{E}_c \setminus \mathcal{E}_{S1}$
  while $|\mathcal{E}| < \kappa$ do
    $e^\star = \operatorname{argmax}_{e\in\overline{\mathcal{E}}} \quad [f(\mathcal{E} \cup \{e\}) - f(\mathcal{E})]$
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{e^\star\}$
    $\overline{\mathcal{E}} \leftarrow \overline{\mathcal{E}} \setminus \{e^\star\}$
  end while
  $\mathcal{E}_{S2} \leftarrow \mathcal{E}$
```

Since the trace of the rigidity Gramian is a modular set function, one also obtains a globally optimal solution for this stage and a globally optimal solution for the whole problem. We will now show that during this stage, in which there is already a minimally rigid edge set, two other spectral Gramian metrics, namely the trace of the Gramian pseudoinverse and the log product of non-zero eigenvalues, correspond to submodular set functions and thus come with an approximation guarantee for this stage.

To motivate the use of the trace and alternative metrics, we now describe how the entire Gramian spectrum affects the performance of formation control and localization tasks. In a formation control context, if the dynamics of each agent is driven by stochastic process noise (e.g., a team of autonomous aircraft under a random wind disturbance), then the agents do not deterministically converge locally to the desired shape, but instead stochastically fluctuate around the desired shape. The *variance* of the linearized positions around the desired shape converges locally to a steady state value. The rigidity Gramian can be interpreted as defining an ellipsoid that quantifies variance of linearized relative position control errors, and all eigenvalues of the rigidity Gramian contribute to errors in various directions. The trace of the Gramian pseudoinverse is then proportional to the average value of the error variance in different directions, and the log product of non-zero eigenvalues of the Gramian is a volumetric quantification of the steady state errors. In a localization setting, the rigidity Gramian and its pseudoinverse are closely related to the Fisher Information Matrix and estimation error covariance associated with linearized position estimates, discussed in more detail below. Similarly, the Gramian defines an uncertainty ellipsoid, and all eigenvalues of the Gramian contribute to estimation errors in various directions; the trace is related to Fisher Information, the trace of the pseudoinverse is proportional to average variance, and the eigenvalue product has volumetric and entropy interpretations.

We have the following result.

Theorem 5. *Let $\mathcal{E} \subseteq \mathcal{E}_c$ and let $X_\mathcal{E} = R^\top_{((\mathcal{V},\mathcal{E}),\mathbf{p})} R_{((\mathcal{V},\mathcal{E}),\mathbf{p})}$ be the rigidity Gramian associated with $\mathcal{E}$. Moreover, assume that there exists an edge set $\mathcal{E}^\star$ such that $((\mathcal{V}, \mathcal{E}^\star), \mathbf{p})$ is minimally rigid, i.e., $\operatorname{rank}((\mathcal{V}, \mathcal{E}^\star), \mathbf{p}) = 2|\mathcal{V}| - 3$, and $\mathcal{E}^\star \subseteq \mathcal{E}$. The following set functions are submodular and monotone increasing*

1) $f_1(\mathcal{E}) = -\operatorname{trace}(X_\mathcal{E}^\dagger)$, *where $X_\mathcal{E}^\dagger$ denotes the Moore-Penrose pseudoinverse of $X_\mathcal{E}$.*

2) $f_2(\mathcal{E}) = \log\left(\prod_{i=1}^{\operatorname{rank}(X_\mathcal{E})} \lambda_i(X_\mathcal{E})\right)$ *where $\lambda_i(X_\mathcal{E})$ is the ith eigenvalue of $X_\mathcal{E}$ and $\lambda_1(X_\mathcal{E}) \geq \ldots \geq \lambda_n(X_\mathcal{E})$.*

As a consequence, when Algorithm 4 is used to optimize these functions, the approximation guarantees provided by Theorems 2 and 3 are obtained.

Proof. We will use the characterization in Theorem 1 to prove the result. Similar to the preceding proof of modularity for the trace metric, the key structure that can be exploited is the additivity of the rigidity Gramian with respect to elements of $\mathcal{E}$.

Trace of the inverse. Take an arbitrary $e \in \mathcal{E}_c$ and consider the derived set functions $f_e : 2^{\mathcal{E}_c\setminus\{e\}} \to \mathbb{R}$ given by

$$\begin{aligned} f_e(\mathcal{E}) &= -\mathrm{trace}(X^{\dagger}_{\mathcal{E}\cup\{e\}}) + \mathrm{trace}(X^{\dagger}_{\mathcal{E}}) \\ &= -\mathrm{trace}((X_{\mathcal{E}} + X_e)^{\dagger}) + \mathrm{trace}(X^{\dagger}_{\mathcal{E}}). \end{aligned}$$

Take any $\mathcal{E}_1 \subseteq \mathcal{E}_2 \subseteq \mathcal{E}_c \setminus \{e\}$. By the additivity property of the Gramian, it is clear that $\mathcal{E}_1 \subseteq \mathcal{E}_2 \Rightarrow X_{\mathcal{E}_1} \preceq X_{\mathcal{E}_2}$. Now define $X(t) = X_{\mathcal{E}_1} + t(X_{\mathcal{E}_2} - X_{\mathcal{E}_1})$ for $t \in [0,1]$. Obviously, $X(0) = X_{\mathcal{E}_1}$ and $X(1) = X_{\mathcal{E}_2}$. Now define

$$\hat{f}_e(X(t)) = -\mathrm{trace}((X(t) + X_e)^{\dagger}) + \mathrm{trace}(X(t)^{\dagger}).$$

Note that $\hat{f}_e(X(0)) = f_e(\mathcal{E}_1)$ and $\hat{f}_e(X(1)) = f_e(\mathcal{E}_2)$. We have

$$\begin{aligned} \frac{d}{dt}\hat{f}_e(X(t)) &= \frac{d}{dt}\Big[-\mathrm{trace}((X(t)+X_e)^{\dagger}) + \mathrm{trace}(X(t)^{\dagger})\Big] \\ &= \mathrm{trace}\Big[(X(t)+X_e)^{\dagger}(X_{\mathcal{E}_2} - X_{\mathcal{E}_1})(X(t)+X_e)^{\dagger}\Big] \\ &\quad - \mathrm{trace}\Big[X(t)^{\dagger}(X_{\mathcal{E}_2} - X_{\mathcal{E}_1})X(t)^{\dagger}\Big] \\ &= \mathrm{trace}\Big[\Big((X(t)+X_e)^{\dagger,2} - X(t)^{\dagger,2}\Big)(X_{\mathcal{E}_2} - X_{\mathcal{E}_1})\Big] \leq 0. \end{aligned}$$

To obtain the second equality we used the matrix derivative formula $\frac{d}{dt}\mathrm{trace}(X(t)^{\dagger}) = \mathrm{trace}(X(t)^{\dagger}\frac{d}{dt}(X(t))\, X(t)^{\dagger})$ which holds whenever $X(t)$ has constant rank for all t, which we have here due to the assumption that there already exists a set of edges that provides rigidity. To obtain the third equality we used the cyclic property of trace. Since $(X(t)+X_e)^{\dagger,2} - X(t)^{\dagger,2} \preceq 0$ and $X_{\mathcal{E}_2} - X_{\mathcal{E}_1} \succeq 0$, the last inequality holds because the trace of the product of a positive and negative semidefinite matrix is non-positive. Since

$$\hat{f}_e(X(1)) = \hat{f}_e(X(0)) + \int_0^1 \frac{d}{dt}\hat{f}_e(X(t))dt,$$

it follows that $\hat{f}_e(X(1)) = f_e(\mathcal{E}_2) \leq \hat{f}_e(X(0)) = f_e(\mathcal{E}_1)$. Thus, f_e is monotone decreasing, and f_1 is submodular by Theorem 1.

Finally, it can be seen from additivity of the rigidity Gramian that f is monotone increasing, which just means that adding an edge to the graph cannot decrease its rigidity.

Product of non-zero eigenvalues. The proof for the product of non-zero eigenvalues has the same structure. Take any $e \in \mathcal{E}_c$. Since $\mathrm{rank}(X_{\mathcal{E}}) = \mathrm{rank}(X_{\mathcal{E}\cup\{e\}})$, from Lemma 1 in the appendix we know that there exists a P such that

$$\log\Bigg(\prod_{i=1}^{\mathrm{rank}(X_{\mathcal{E}})} \lambda_i(X_{\mathcal{E}})\Bigg) = \log\,\det(\widetilde{X}_{\mathcal{E}})$$

and

$$\begin{aligned} \log\Bigg(\prod_{i=1}^{\mathrm{rank}(X_{\mathcal{E}\cup\{e\}})} \lambda_i(X_{\mathcal{E}\cup\{e\}})\Bigg) &= \log\Bigg(\prod_{i=1}^{\mathrm{rank}(X_{\mathcal{E}})} \lambda_i(X_{\mathcal{E}\cup\{e\}})\Bigg) \\ &= \log\,\det(\widetilde{X}_{\mathcal{E}\cup\{e\}}), \end{aligned}$$

where $\widetilde{X}_{\mathcal{E}} = X_{\mathcal{E}} + P$ and $\widetilde{X}_{\mathcal{E}\cup\{e\}} = X_{\mathcal{E}\cup\{e\}} + P$. So, it is enough to show the submodularity of $\log\det(\widetilde{X}_{\mathcal{E}})$. Consider the derived set functions $f_e : 2^{\mathcal{E}_c\setminus\{e\}} \to \mathbb{R}$ given by

$$\begin{aligned} f_e(\mathcal{E}) &= \log\,\det\widetilde{X}_{\mathcal{E}\cup\{e\}} - \log\,\det\widetilde{X}_{\mathcal{E}} \\ &= \log\,\det(\widetilde{X}_{\mathcal{E}} + X_e) - \log\,\det\widetilde{X}_{\mathcal{E}}. \end{aligned}$$

Take any $\mathcal{E}_1 \subseteq \mathcal{E}_2 \subseteq \mathcal{E}_c \setminus \{e\}$, remember that $X_{\mathcal{E}_1} \preceq X_{\mathcal{E}_2}$, and define $\widetilde{X}(t) = (X_{\mathcal{E}_1} + P) + t\Big((X_{\mathcal{E}_2} + P) - (X_{\mathcal{E}_1} + P)\Big) = (X_{\mathcal{E}_1} + P) + t(X_{\mathcal{E}_2} - X_{\mathcal{E}_1})$ for $t \in [0,1]$ and

$$\hat{f}_e(X(t)) = \log\,\det(\widetilde{X}(t) + X_e) - \log\,\det\widetilde{X}(t).$$

Since we assumed that there exists a set of edges that provides rigidity and P is constant, $X(t)$ has full rank for all t, and it follows that

$$\begin{aligned} \frac{d}{dt}\hat{f}_e(X(t)) &= \frac{d}{dt}\Big[\log\,\det(\widetilde{X}(t) + X_e) - \log\,\det\widetilde{X}(t)\Big] \\ &= \mathrm{trace}\Big[(\widetilde{X}(t) + X_e)^{-1}(X_{\mathcal{E}_2} - X_{\mathcal{E}_1})\Big] \\ &\quad - \mathrm{trace}\Big[\widetilde{X}(t)^{-1}(X_{\mathcal{E}_2} - X_{\mathcal{E}_1})\Big] \\ &= \mathrm{trace}\Big[\Big((\widetilde{X}(t) + X_e)^{-1} - \widetilde{X}(t)^{-1}\Big)(X_{\mathcal{E}_2} - X_{\mathcal{E}_1})\Big] \leq 0. \end{aligned}$$

We used the matrix derivative formula $\frac{d}{dt}\log\,\det\widetilde{X}(t) = \mathrm{trace}[\widetilde{X}(t)^{-1}\frac{d}{dt}\widetilde{X}(t)]$. The remainder of the proof follows trace of the pseudoinverse. □

We conclude this section by noting that the cost functions in the two stages of constructing the network need not be the same, i.e., the cost function of the optimization problem (12) maybe different from the one in (13). To emphasise this, let $f_{S1}(\mathcal{E})$ and $f_{S2}(\mathcal{E})$ denote the cost functions of (12) and (13), respectively. For example, one might first construct a minimally rigid via applying Algorithm 3 for the case where $f_{S1}(\mathcal{E}) = \mathrm{trace}(X_{\mathcal{E}})$ and then add extra edges to this minimally rigid network though the application of Algorithm 4 where $f_{S2}(\mathcal{E}) = \prod_{i=1}^{\mathrm{rank}(X_{\mathcal{E}})} \lambda_i(X_{\mathcal{E}})$, where $X_{\mathcal{E}}$ is the rigidity Gramian associated with $\mathcal{E} \subset \mathcal{E}_c$ as defined earlier.

4 Optimal Anchor Selection

We now consider a separate but related network design problem involving a choice of subsets of vertices rather than subsets of edges. As stated earlier, a globally rigid network is uniquely realisable. However, even given the graph and distance set of a globally rigid network, there is not enough information to position the network absolutely in $\mathbb{R}^2$. In fact, the network can only be positioned to within a translation, rotation or reflection. To eliminate this non-uniqueness requires further knowledge, typically the absolute position of at least three vertices. In a physical sensor network, this information is either derived from global position sensing (GPS) measurements or other form of independent measurements. The vertices in question must not be non-collinear; for if they were, there would be ambiguity up to a reflection in the position of all other vertices. The vertices with known positions are called *anchors*, and we denote the set of all anchors with $\mathcal{A}$. Just as the edge sets discussed in the previous section affected the network rigidity properties, so do sets of chosen anchors.

In this vertex selection setting, a rigidity Gramian again quantifies network rigidity properties for anchor selections, and this translates to better performance in localization tasks. Links between the Gramian and localization properties are described in detail in [19] and [20]. A noise-free network localization problem comprises finding sensor positions p_i, $i \in \mathcal{V} \setminus \mathcal{A}$ such that

$$\|p_i - p_j\|^2 = d_{ij}^2, \quad \{i,j\} \in \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}, \tag{17}$$

given a set of distance measurements d_{ij}, $\{i,j\} \in \mathcal{E}$ between certain pairs of sensors and at least three non-collinear anchor positions p_j, $j \in \mathcal{A}$. This problem has a unique solution if $\mathcal{G}$ is generically globally rigid. When the distance measurements are noisy, i.e., d_{ij}^2 are replaced by $d_{ij}^2 + \delta_{ij}$ (17) with δ_{ij} being an error in the squared distance (rather than in the distance itself), there is in general no exact solution, but one can obtain an approximate solution by solving

$$\min_{p_i, i \in \mathcal{V}\setminus\mathcal{A}} \sum_{\{i,j\}} \Big[||p_i - p_j||^2 - \big(d_{ij}^2 + \delta_{ij}\big) \Big]^2. \tag{18}$$

Let $\bar{p}_i$ be the true position of node i and $p_i^\star$ is obtained from solving (18), and define $\tilde{p}_i = p_i^\star - \bar{p}_i$. From [19, Theorem 2] one can observe that $\tilde{p}_i$ is the minimizer of the following optimization problem

$$\min_{\tilde{\mathbf{p}}} \left[\left\| R_r(\mathcal{A}) \tilde{\mathbf{p}} - \frac{\tilde{\delta}}{2} \right\|^2 + \frac{1}{2} \left\| \tilde{H} \tilde{\mathbf{p}} \right\|^2 \right]. \tag{19}$$

Here $R_r(\mathcal{A})$, the *reduced rigidity matrix*, is constructed by deletion of columns of $R_{(\mathcal{G},\mathbf{p})}$ corresponding to anchor positions and rows of $R_{(\mathcal{G},\mathbf{p})}$ corresponding to edges between two anchors, respectively. Furthermore, $\tilde{\mathbf{p}}$ is the perturbation in non-anchor positions and $\tilde{\delta}$ is the error vector in the square of the length of edges connecting those edges with at least one non-anchor end. Additionally, $\tilde{H} = H \otimes I_2$ where H is obtained by removing the columns and rows of the incidence matrix of $\mathcal{G}$ corresponding to the anchor positions and the edges connecting the anchors, respectively; $I_2 \in \mathbb{R}^{2\times 2}$ is the identity matrix. The optimization problem described by (19) has the following unique solution:

$$\tilde{\mathbf{p}} = \tilde{R}_r \tilde{\delta}$$

where

$$\tilde{R}_r = \frac{1}{2} \left(R_r(\mathcal{A})^\top R_r(\mathcal{A}) + \frac{1}{4} \tilde{H}^\top \tilde{H} \right)^{-1} R_r(\mathcal{A})^\top. \tag{20}$$

Note that $R_r(\mathcal{A})$ depends on the positions of the nodes while $\tilde{H}$ does not. As a result by an appropriate choice of system of coordinates $\tilde{R}_r$ can be approximated arbirarily accurately as

$$\tilde{R}_r \approx \frac{1}{2} \Big(R_r(\mathcal{A})^\top R_r(\mathcal{A}) \Big)^{-1} R_r(\mathcal{A})^\top = \frac{1}{2} R_r(\mathcal{A})^\dagger,$$

where $R_r(\mathcal{A})^\dagger$ is the Moore-Penrose pseudoinverse of $R_r(\mathcal{A})$ (For its existence see [19, Theorem 3]). Since $\tilde{\delta}$ is unknown accurately calculating the error in the localization is impossible. However, the error vector $\tilde{\delta}$ can be modeled by random variables, with specific covariance and mean value. Let $\mathrm{cov}(\tilde{\delta})$ denote the covariance of $\tilde{\delta}$. Thus,

$$\mathrm{cov}(\tilde{\mathbf{p}}) = \frac{1}{4} R_r(\mathcal{A})^\dagger \mathrm{cov}(\tilde{\delta}) R_r(\mathcal{A})^{\dagger^\top}. \tag{21}$$

Often in practice $\mathrm{cov}(\tilde{\delta})$ can be assumed to be diagonal with nonidentical diagonal entries. Let σ^2 be the minimum of these diagonal entries, hence,

$$\mathrm{cov}(\tilde{\mathbf{p}}) = \frac{1}{4} R_r(\mathcal{A})^\dagger \mathrm{cov}(\tilde{\delta}) R_r(\mathcal{A})^{\dagger^\top} \geq \frac{\sigma^2}{4} R_r(\mathcal{A})^\dagger R_r(\mathcal{A})^{\dagger^\top}. \tag{22}$$

Therefore, the magnitude of the eigenvalues of the *reduced rigidity Gramian* induced by anchor set $\mathcal{A}$, denoted $X_{\mathcal{A}} = R_r(\mathcal{A})^\top R_r(\mathcal{A})$ are directly related with the lower bound on the covariance of error in position estimates, which relates to Fisher information. Before proceeding further we comment on the eigenvalues of the reduced rigidity Gramian in a deterministic setting. From [20] it is known that there exists a suitably small positive Δ and an associated positive constant c such that if the measurement errors in the squares of the distances obey $\| \sum_{\{i,j\}} \delta_{ij} \| < \Delta$, the solution of (18) is unique and $\| \sum_{\{i,j\}} p_i^* - \bar{p}_j \| \leq c \| \sum_{\{i,j\}} \delta_{ij} \|$. Moreover, the magnitude of c is inversely proportional to the magnitudes of the eigenvalues of the reduced rigidity Gramian induced by anchor set $\mathcal{A}$. To summarize, networks with large reduced rigidity Gramians for a given anchor selection allow for better performance in localization tasks.

The problem of anchor selection in a network in its most general form can be written as:

$$\underset{\mathcal{A}\subseteq\mathcal{V}, |\mathcal{A}|=m}{\text{maximize}} \quad f(\mathcal{A}), \tag{23}$$

where $m \geq 3$ is the number of anchors to be chosen and $f : 2^{\mathcal{V}} \to \mathbb{R}$ is a set function. This problem is again a difficult combinatorial optimization problem. For certain functions of the reduced rigidity Gramian, there are convex relaxation techniques that can be employed to obtain approximations efficiently; e.g., see [19]. However, such techniques often suffer from two drawbacks: (i) they generally return a suboptimal solution with no performance guarantees, and (ii) the semidefinite programs that emerge out of such relaxations, though computationally efficient in theory, are numerically cumbersome and are virtually unsolvable with modern general purpose solvers for networks with size larger than 100 nodes.[2]

Alternatively, one can consider purely combinatorial algorithms and study properties of associated set functions. We will show that a certain function of the reduced rigidity Gramian has a desirable modularity property that allows a

[2] Developing more efficient relaxation techniques for solving mixed-integer semidefinite optimization problems is an active area of research and methods with better peformance might be developed in the future.

globally optimal anchor selection to be obtained from the application of Algorithm 1 to (23). Before stating our main result in this section, we make a technical assumption that ensures that the associated localization problem after any anchor selection is well-posed.

Assumption 2. *Consider a network $\mathcal{N} = (G, \mathbf{p})$, with the underlying sensing graph $G = (\mathcal{V}, \mathcal{E})$, where no agent is an anchor. For any subset $\mathcal{A} \subset \{1, 2, \dots, n\}$ with $|\mathcal{A}| = m \geq 3$, let $\mathcal{N}(\mathcal{A}) = (\mathcal{G}(\mathcal{A}), \mathbf{p})$ with underlying sensing graph $\mathcal{G}(\mathcal{A}) = (\mathcal{V}, \mathcal{E}(\mathcal{A}))$ denote the network obtained from $\mathcal{N}$ by assigning all the agents $i \in \mathcal{A}$ as anchors. Let $\mathcal{E}(\mathcal{A})$ denote the set of edges connecting these anchor agents. It is assumed that $\mathcal{N}(\mathcal{A})$ is globally rigid for any $\mathcal{A} \subset \{1, 2, \dots, n\}$ with $|\mathcal{A}| = m \geq 3$.*

We have the following result for the trace of the reduced rigidity Gramian.

Theorem 6. *For any selection of anchors $\mathcal{A} \subset \mathcal{V}$ such that $|\mathcal{A}| \geq 3$, the set function $f(\mathcal{A}) = \mathrm{trace}(X_{\mathcal{A}})$, where $X_{\mathcal{A}} = R_r(\mathcal{A})^\top R_r(\mathcal{A})$ is the reduced rigidity Gramian and $R_r(\mathcal{A})$ is the reduced rigidity matrix, is modular. As a consequence, when Algorithm 1 is used to optimize this function, a globally optimal anchor selection is obtained.*

Proof. To prove the statement note that $\mathrm{trace}[(R_r(\mathcal{A})^\top R_r(\mathcal{A})] = \mathrm{trace}[(R_r(\mathcal{A})R_r(\mathcal{A})^\top]$. Moreover,

$$\mathrm{trace}[(R_r(\mathcal{A})R_r(\mathcal{A})^\top] = \mathrm{trace}\left[\sum_{i \in \overline{\mathcal{A}}} r_i^\top r_i\right] = \sum_{i \in \overline{\mathcal{A}}} \mathrm{trace}\left[r_i^\top r_i\right],$$

where $\overline{\mathcal{A}} = \mathcal{V} \setminus \mathcal{A}$ and $r_i \in \mathbb{R}^{|\mathcal{E}| \times 2}$ is a matrix with the $2i-1$th and $2i$th columns of the rigidity matrix $R((\mathcal{G}, \mathcal{E}), \mathbf{p})$. From Definition 4, the function is modular. □

One can consider other spectral functions of the reduced rigidity Gramian. However, as in the first-stage problem in the previous section, submodularity properties cannot be established with the approach of the proof of Theorem 5 because the Gramian changes rank (indeed, here it also changes dimension) as anchors are added. If submodularity could be established for other functions, the approximation guarantee of Theorem 2 would apply when Algorithm 1 is applied. In any case, even though we cannot currently establish submodularity properties for other functions, this does not preclude the use of the greedy algorithm. Though the solutions do not currently come with approximation guarantees, the algorithm may still yield good solutions and scales to large problems.

Selecting the anchors via the method described above requires forming the reduced rigidity matrix, which in turn requires knowledge of certain relative positions between the nodes in the network. One can envisage two scenarios in which the method could be used. First, suppose that multiple nodes in a sensor network have GPS receivers, and to conserve energy it is desired that only a few receivers should be turned on for localization purposes. Initially, a random subset of them may activate their GPS receivers so that a coarse localization can be done. Using the information obtained from this initial optimization step, a rigidity matrix can be constructed, which consequently makes selecting which nodes to turn on their GPS receivers, i.e., be selected as an anchor, in order to obtain a better localization accuracy possible. Second, suppose that relative position information in the absence of a global coordinate frame is available, however, finding the positions in a global coordinate frame requires fixing the position of a subset of nodes, i.e., selecting some of them as anchors. This is particularly important when the nodes are moving while maintaining their relative positions from each other.

5 Numerical Examples

In this section, initially, we study the performance of the algorithms introduced in Section 3.

In the first scenario, we compare the performance of Algorithms 3 and 4 with a semidefinite programming (SDP) relaxation of (11) given below:

$$\begin{aligned}
&\underset{t_i}{\text{maximize}} && f\left(\sum_{i=1}^{|\mathcal{V}|(2|\mathcal{V}|-1)/2} t_i s_i^\top s_i\right) \\
&\text{subject to} && t_i \in [0,1], \quad \sum_{i=1}^{|\mathcal{V}|(2|\mathcal{V}|-1)/2} t_i = \kappa, \\
& && i = 1, \dots, |\mathcal{V}|(2|\mathcal{V}|-1)/2, \\
& && B^\top \left(\sum_{i=1}^{|\mathcal{V}|(2|\mathcal{V}|-1)/2} t_i s_i^\top s_i\right) B \succeq \gamma I.
\end{aligned} \tag{24}$$

where s_i its ith row of $((\mathcal{V}, \mathcal{E}_c), \mathbf{p})$ — the rigidity matrix associated with network , — B is a $2|\mathcal{V}| \times 2|\mathcal{V}| - 3$ matrix whose columns are orthogonal to the vectors in the null space of the rigidity Gramian of a rigid network— i.e., they are orthogonal to $[1, 0, 1, 0 \dots]^\top$, $[0, 1, 0, 1, \dots]^\top$, and $[y_1, -x_1, y_2, -x_2, \dots]^\top$, — and $\gamma \ll 1$ is a small positive constant. Consequently, the resulting network from solving (24) is obtained from selecting edges $\eta(j)$ ($\eta : \{1, \dots, |\mathcal{V}|(|\mathcal{V}| - 1)/2\} \to \mathcal{E}_c$ relates each index j to its corresponding edge) where the corresponding $t_j^\star$ obtained from (24) is among the κ-largest values of $t_i^\star$, $i = 1, \dots, |\mathcal{V}|(2|\mathcal{V}| - 1)/2$. However, the network obtained from solving (24) is not guaranteed to be rigid.

To illustrate this we constructed networks via solving (24) for a network of $n = 50$ nodes and different κ. We drew sensor positions from a uniform distribution and compared the average of the trace of the rigidity Gramian of the networks compared from solving (24) with those obtained from applying the two-stage method outlined in Section 3 after 100 runs in Fig. 4. For smaller values of κ it can be seen that solving (24) results in networks with larger Gramian teace compared to Algorithms 3 and 4. However, it is important to note that in none of these situations the resulting network is guaranteed to be actually rigid. The percentage of the cases where the resulting network is not rigid for different edge numbers κ is shown in Fig. 5.

In the second scenario, we consider the problem of constructing minimally rigid networks. To this aim, we compared the magnitude of the trace of the rigidity Gramian obtained from applying the Henneberg sequence and Algorithm 3. The numerical experiments are repeated 20 times

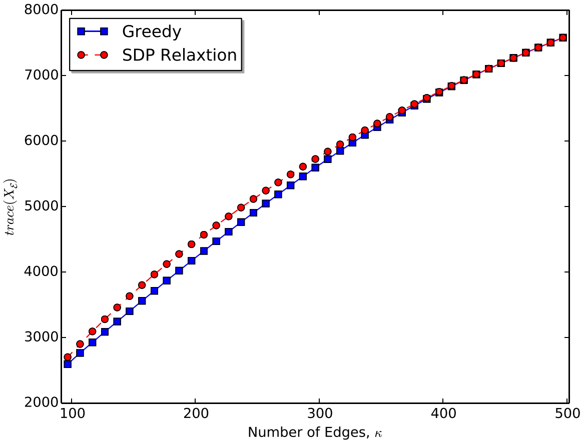

Fig. 4. The trace of the rigidity Gramian of networks generated via applying Algorithm 3 and 4 versus solving (24) for $n = 50$, different values of $\kappa \in \{97, 107, 117, \ldots, 497\}$, and 100 instances of nodes positions, p_i, $i = 1, \ldots, n$.

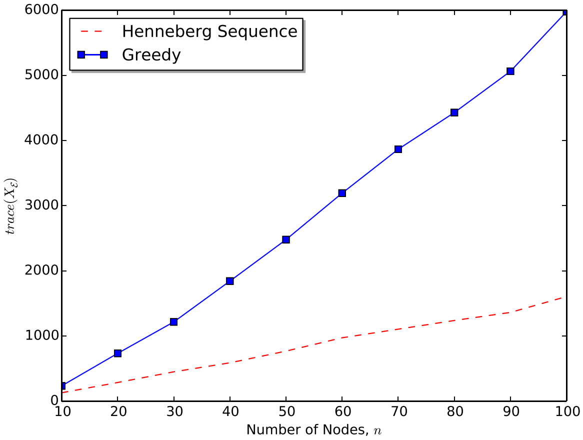

Fig. 6. The trace of the rigidity Gramian of networks generated via applying Algorithm 3 and a Henneberg sequence for constructing minimally rigid networks for different number of nodes, n.

and the average of the trace across all experiments for different number of nodes are constructed. The positions of the nodes are selected randomly from a uniform distribution for each experiment. The result is depicted in Fig. 6. It can be seen that the greedy algorithm exhibits a far better performance than randomly applying Henneberg sequences. Note that while applying random Henneberg sequence might not seem to be the most efficient strategy to construct rigid networks, it is the only method in the literature for constructing rigid graphs. There is no other method to this date that uses algebraic rigidity properties to construct rigid networks.

Next we study the problem of anchor selection in a network of 1,000 nodes where 20 anchors are to be selected. The result using the trace metric is shown in Fig. 7. Qualitatively, those nodes are selected as anchors that the sum of the lengths of all the edges incident at them are the largest. Particularly, it can be observed that the majority of the selected anchors lie in an area with equal distances to the center and the boundary of the network. The anchor selection problem using `python` took about 49 seconds on a 2.8 GHz laptop; the convex relaxation algorithm proposed in [19] fails to run on a problem this size using standard SDP solvers.

In all the numerical examples we encountered the greedy algorithm not only guarantees solutions that satisfy the rank constraint but also scales to far larger problems than the SDP relaxations. For the greedy, limiting factor is function evaluation, which only involves carrying out an singular value decomposition (SVD) which can easily scale to tens of thousands of variables using standard algorithms on typical laptops. Solving the SDP relaxation, on the other hand, becomes virtually impossible using typical solvers on standard laptops for problems with around 100 variables.

To conform with the guidelines of reproducible research the `python` scripts used to generate the results in this paper are available at [42].

6 Conclusion

In this paper, we proposed algorithms to construct networks with desirable algebraic rigidity properties. We identified several scalar metrics of the rigidity Gramian that are modular or submodular set functions, which allows global optimality or approximation guarantees to be obtained using simple

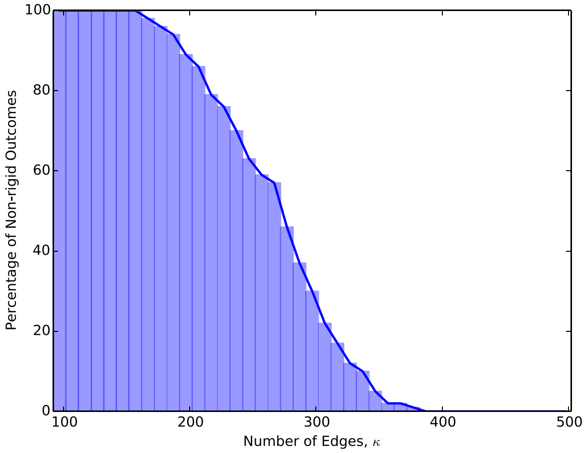

Fig. 5. The percentage of the cases where the network obtained from solving (24) is not rigid for $n = 50$, different values of $\kappa \in \{97, 107, 117, \ldots, 497\}$, and 100 instances of nodes positions, p_i, $i = 1, \ldots, n$.

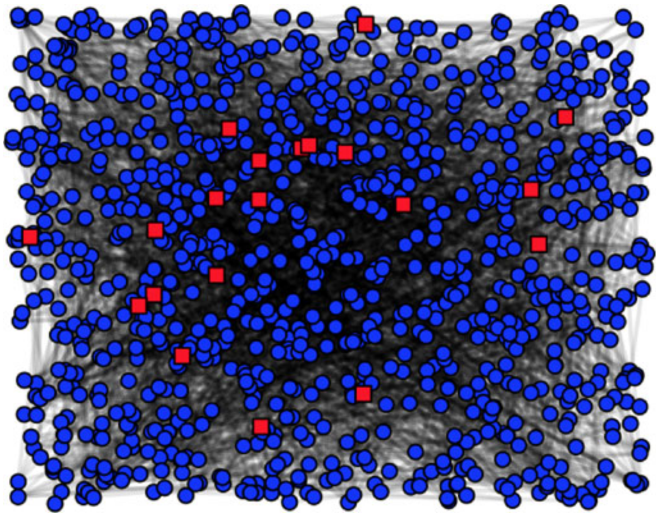

Fig. 7. The selection of 20 anchors in a network of 1,000 nodes. The selected anchors are depicted as red rectangles.

greedy algorithms. A matroid constrained formulation was used to enforce a rank constraint that ensures that solutions also satisfy binary rigidity properties. We also considered the related problem of optimal anchor selection for sensor networks localization. The problem was again formulated as a set function optimization problem, and we identified a modular metric that can be globally optimized with a greedy algorithm. Almost all methods in the literature for constructing rigid networks are purely graph theoretic and do not consider algebraic rigidity properties; our methods can produce networks with vastly superior performance in localization and formation control tasks. Moreover, the greedy algorithms scale to problems far beyond the current capabilities of semidefinite programming based relaxation techniques.

There are several interesting topics for future work. One would be to incorporate the objective functions considered in [15], [16], [17] and study the corresponding performance tradeoffs and (sub)modularity properties. Others would be to study other objective functions associated with the anchor selection problem and to pursue theoretical extensions and empirically evaluate the greedy algorithm.

Appendix

Lemma 1. *Consider* $X, Y \in \mathbb{R}^{n\times n}$ *be positive semidefinite matrices such that* $\mathrm{rank}(Y) \le \mathrm{rank}(X) \le n$ *and* $\mathrm{rank}(X+Y) = \mathrm{rank}(X)$. *First, the null spaces of* X *and* $X+Y$ *are the same. Second, there exists a positive semidefinite* P *such that*

$$\lambda_i(X+P) = \begin{cases} 1 & , \lambda_i(X) = 0 \\ \lambda_i(X) & , \lambda_i(X) \neq 0. \end{cases} \tag{25}$$

Third, for the same P *as in (25),* $\prod_{i=1}^{\mathrm{rank}(X+Y)} \lambda_i(X + Y) = \det(X+Y+P)$, *where* $\lambda_i(\cdot)$ *is the* i*th eigenvalue of its argument and* $\lambda_n(\cdot) \le \ldots \le \lambda_i(\cdot)$.

Proof. Let $\mathrm{rank}(X) = \rho$ and u_i be the unit length eigenvectors of X associated with $\lambda_i(X)$ such that $u_i^\top u_j = 0$ for $i \neq j$. For any $j \in \{n-\rho+1, \ldots, n\}$, It follows that $u_j^\top(X+Y)u_j = u^\top Yu$. To obtain a contradiction assume there is a $j \in \{n-\rho+1, \ldots, n\}$ such that $u_j^\top Yu_j \neq 0$. Thus the dimension of the null space of $X+Y$ is at most $n-\rho-1$, which is equivalent to $\mathrm{rank}(X+Y) = \rho+1$ which is a contradiction. Hence, the spaces spanned by the eigenvalues of X and $X+Y$ are the same.

Second, because X is symmetric it can be diagonalised. Let $X = U\Lambda(X)U^\top$, where $U^\top U = I$, $\Lambda(X) = \mathrm{diag}(\lambda_1(X), \ldots, \lambda_n(X))$, and $U = [\,u_1 \quad \ldots \quad u_n\,]$. Define $P = U\Lambda_P U^\top$ where Λ_P has zero entries except the iith diagonal entry of 1 if $\lambda_i(X) = 0$. Such a P satisfies (25) and $P = \sum_{i=n-\rho+1}^{n} u_i u_i^\top$.

Third, $X+Y = \overline{U}\Lambda(X+Y)\overline{U}^\top$, where $\overline{U}^\top\overline{U} = I$, $\Lambda(X+Y) = \mathrm{diag}(\lambda_1(X+Y), \ldots, \lambda_n(X+Y))$, and $\overline{U} = [\,_1 \quad \ldots \quad \bar{u}_n\,]$ with $\bar{u}_i$ being the ith eigenvector of $X+Y$ associated with $\lambda_i(X+Y)$. Similar to above there exists a $\overline{P} = \sum_{i=n-\rho+1}^{n} \bar{u}_i\bar{u}_i^\top$ such that

$$\lambda_i(X+Y+\overline{P}) = \begin{cases} 1 & , \lambda_i(X+Y) = 0 \\ \lambda_i(X+Y) & , \lambda_i(X+Y) \neq 0. \end{cases} \tag{26}$$

Hence, $X+Y+\overline{P}$ is nonsingular and $\prod_{i=1}^{\mathrm{rank}(X+Y)} \lambda_i(X+Y) = \det(X+Y+\overline{P})$. Moreover, in light of the first statement, $\bar{u}_i = u_i$, $i = n-\rho+1, \ldots, n$. Thus $P = \overline{P}$ and $\prod_{i=1}^{\mathrm{rank}(X+Y)} \lambda_i(X+Y) = \det(X+Y+P)$. □

Acknowledgments

Iman Shames is supported by a McKenzie fellowship. Tyler Summers is partially supported by an ETH Fellowship.

References

[1] R. Olfati-Saber and R. M. Murray, "Graph rigidity and distributed formation stabilization of multi-vehicle systems," in *Proc. 41st IEEE Conf. Decision Control*, 2002, vol. 3, pp. 2965–2971.

[2] B. D. Anderson, C. Yu, B. Fidan, and J. M. Hendrickx, "Rigid graph control architectures for autonomous formations," *IEEE Control Syst. Mag.*, vol. 28, no. 6, pp. 48–63, Dec. 2008.

[3] C. Yu, J. M. Hendrickx, B. Fidan, B. Anderson, and V. D. Blondel, "Three and higher dimensional autonomous formations: Rigidity, persistence and structural persistence," *Automatica*, vol. 43, no. 3, pp. 387–402, 2007.

[4] L. Krick, M. E. Broucke, and B. A. Francis, "Stabilisation of infinitesimally rigid formations of multi-robot networks," *Int. J. Control*, vol. 82, no. 3, pp. 423–439, 2009.

[5] C. Yu, B. D. Anderson, S. Dasgupta, and B. Fidan, "Control of minimally persistent formations in the plane," *SIAM J. Control Optim.*, vol. 48, no. 1, pp. 206–233, 2009.

[6] T. H. Summers, C. Yu, S. Dasgupta, and B. D. Anderson, "Control of minimally persistent leader-remote-follower and coleader formations in the plane," *IEEE Trans. Autom. Control*, vol. 56, no. 12, pp. 2778–2792, Dec. 2011.

[7] T. Eren, O. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. Anderson, and P. N. Belhumeur, "Rigidity, computation, and randomization in network localization," in *Proc. 23rd Annu. Joint Conf. IEEE Comput. Commun. Soc.*, 2004, vol. 4, pp. 2673–2684.

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

[9] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proc. 2nd Int. Conf. Embedded Netw. Sens. Syst.*, 2004, pp. 50–61.

[10] G. Mao, B. Fidan, and B. Anderson, "Wireless sensor network localization techniques," *Comput. Netw.*, vol. 51, no. 10, pp. 2529–2553, 2007.

[11] A. M.-C. So and Y. Ye, "Theory of semidefinite programming for sensor network localization," *Math. Programm.*, vol. 109, no. 2-3, pp. 367–384, 2007.

[12] B. Hendrickson, "Conditions for unique graph realizations," *SIAM J. Comput.*, vol. 21, no. 1, pp. 65–84, 1992.

[13] R. Connelly, "Generic global rigidity," *Discrete Comput. Geom.*, vol. 33, no. 4, pp. 549–563, 2005.

[14] B. Jackson and T. Jordán, "Connected rigidity matroids and unique realizations of graphs," *J. Combinatorial Theory, Series B*, vol. 94, no. 1, pp. 1–29, 2005.

[15] R. Ren, Y.-Y. Zhang, X.-Y. Luo, and S.-B. Li, "Automatic generation of optimally rigid formations using decentralized methods," *Int. J. Autom. Comput.*, vol. 7, no. 4, pp. 557–564, 2010.

[16] D. Zelazo and F. Allgower, "Growing optimally rigid formations," in *Proc. Am. Control Conf.*, 2012, pp. 3901–3906.

[17] A. Priolo, R. Williams, A. Gasparri, and G. Sukhatme, "Decentralized algorithms for optimally rigid network constructions," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 5010–5015.

[18] A. Gasparri, R. K. Williams, A. Priolo, and G. S. Sukhatme, "Decentralized and parallel constructions for optimally rigid graphs in $\mathbf{R}^2$," *IEEE Trans. Mobile Comput.*, vol. 14, no. 11, pp. 2216–2228, Nov. 2015.

[19] I. Shames, B. Fidan, and B. Anderson, "Minimization of the effect of noisy measurements on localization of multi-agent autonomous formations," *Automatica*, vol. 45, no. 4, pp. 1058–1065, 2009.

[20] B. D. Anderson, I. Shames, G. Mao, and B. Fidan, "Formal theory of noisy sensor network localization," *SIAM J. Discrete Math.*, vol. 24, no. 2, pp. 684–698, 2010.

[21] D. Zelazo, A. Franchi, H. H. Bülthoff, and P. R. Giordano, "Decentralized rigidity maintenance control with range-only measurements for multi-robot systems," *Int. J. Robot. Res.*, vol. 34, no. 1, pp. 105–128, 2015.

[22] L. Henneberg, *Die graphische Statik der Starren Systeme*, BG Teubner, Leipzig, 1911, vol. 31.

[23] T.-S. Tay and W. Whiteley, "Generating isostatic frameworks," *Struct. Topol.*, no. 11, pp. 21–69,1985.

[24] P. Biswas, T.-C. Liang, K.-C. Toh, Y. Ye, and T.-C. Wang, "Semidefinite programming approaches for sensor network localization with noisy distance measurements," *IEEE Trans. Autom. Sci. Eng.*, vol. 3, no. 4, pp. 360–371, Oct. 2006.

[25] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: A quantitative comparison," *Comput. Netw.*, vol. 43, no. 4, pp. 499–518, 2003.

[26] S. P. Chepuri, G. Leus et al., "Sparsity-exploiting anchor placement for localization in sensor networks," in *Proc. 21st Eur. Signal Process. Conf.*, 2013, pp. 1–5.

[27] J. N. Ash and R. L. Moses, "On optimal anchor node placement in sensor localization by optimization of subspace principal angles," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2008, pp. 2289–2292.

[28] N. Salman, H. K. Maheshwari, A. H. Kemp, and M. Ghogho, "Effects of anchor placement on mean-CRB for localization," in *Proc. 10th IFIP Annu. Mediterranean Ad Hoc Netw. Workshop*, 2011, pp. 115–118.

[29] T. H. Summers and J. Lygeros, "Optimal sensor and actuator placement in complex dynamical networks," in *IFAC World Congress*, Cape Town, South Africa, vol. 19, pp. 3784–3789, Jul. 2014.

[30] F. L. Cortesi, T. H. Summers, and J. Lygeros, "Submodularity of energy related controllability metrics," *arXiv preprint arXiv:1403.6351*, 2014.

[31] T. H. Summers, F. L. Cortesi, and J. Lygeros, "On submodularity and controllability in complex dynamical networks," *arXiv preprint arXiv:1404.7665*, 2014.

[32] T. Summers, I. Shames, J. Lygeros, and F. Dörfler, "Topology design for optimal network coherence," in *Proc. Eur. Control Conf.*, 2015.

[33] A. Clark, L. Bushnell, and R. Poovendran, "A supermodular optimization framework for leader selection under link noise in linear multi-agent systems," *IEEE Trans. Autom. Control*, vol. 59, no. 2, pp. 283–296, Feb. 2014.

[34] G. Laman, "On graphs and rigidity of plane skeletal structures," *J. Eng. Math.*, vol. 4, no. 4, pp. 331–340, 1970.

[35] L. Asimow and B. Roth, "The rigidity of graphs, ii," *J. Math. Anal. Appl.*, vol. 68, no. 1, pp. 171–190, 1979.

[36] L. Lovász, "Submodular functions and convexity," in *Mathematical Programming The State of the Art*. New York, NY, USA: Springer, 1983, pp. 235–257.

[37] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions-I," *Math. Programm.*, vol. 14, no. 1, pp. 265–294, 1978.

[38] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, "An analysis of approximations for maximizing submodular set functions-II," *Polyhedral Combinatorics*, vol. 8, pp. 73–87, 1978.

[39] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák, "Maximizing a monotone submodular function subject to a matroid constraint," *SIAM J. Comput.*, vol. 40, no. 6, pp. 1740–1766, 2011.

[40] J. G. Oxley, *Matroid Theory*. London, U.K.: Oxford Univ. Press, 2006, vol. 3.

[41] J. Graver, "Rigidity matroids," *SIAM J. Discrete Math.*, vol. 4, no. 3, pp. 355–368, 1991.

[42] I. Shames. (2014). Simulations scripts of Rigid Network Design Via Submodular Set Function Optimization [Online]. Available: http://eemensch.tumblr.com/post/91484828876/rigidnetworkdesign

Iman Shames received the BSc degree in electrical engineering from Shiraz University, Iran in 2006, and the PhD degree in engineering and computer science from the Australian National University, Canberra, Australia in 2011. He is a senior lecturer and a McKenzie fellow at the Department of Electrical and Electronic Engineering, the University of Melbourne. Previously, he was an ACCESS postdoctoral researcher at the ACCESS Linnaeus Centre, the KTH Royal Institute of Technology, Stockholm, Sweden. He has been a visiting researcher at ETH in 2012, Chalmers Technical University in 2011, the Delft Technical University in 2011, TU-Munchen in 2011, the KTH Royal Institute of Technology in 2010, the University of Tokyo in 2008, and at the University of Newcastle in 2010 and 2005. His current research interests include large scale optimisation, secure and reliable complex systems, ensuring privacy in networked systems, and localization and mapping in multi-robot systems.

Tyler H. Summers received the BS degree in mechanical engineering from Texas Christian University in 2004, and the MS and PhD degrees in aerospace engineering with emphasis on feedback control theory at the University of Texas at Austin in 2007 and 2010, respectively. He is an assistant professor of mechanical engineering at the University of Texas at Dallas. Prior to joining UT Dallas, he was an ETH postdoctoral fellow at the Automatic Control Laboratory at ETH Zurich from 2011 to 2015. He was a fulbright postgraduate scholar at the Australian National University in Canberra, Australia in 2007-2008. His research interests are in feedback control and optimization in complex dynamical networks, emphasizing theoretical tools and computational methods and driven by applications to electric power networks and distributed robotics.