Maximum Rigid Components as Means for Direction-based Localization in Sensor Networks*

Bastian Katz   Marco Gaertler   Dorothea Wagner

Universität Karlsruhe

Abstract. Many applications or algorithms in sensor networks require positional information of the sensors. Most approaches for this problem rely either on distances between communicating node pairs or on local angular information. Although distance-based methods are widespread, we here present a technique for direction-based localization in general networks. Whereas Bruck et al. proved that the corresponding realization problem can be solved by Linear Programming but becomes $\mathcal{NP}$-hard [2] for unit-disk-graphs, we focus on rigid components which allow both efficient identification and fast, unique realizations. We propose a technique to group small rigid components that can be found by standard techniques to maximum such components using a reduction to maximum flow problems. The method is analyzed for the two-dimensional case, but can easily be extended to higher dimensions. By evaluating our approach on (quasi-)unit-disk graphs, we observe that the required density in order to localize a large percentage of the network is comparably small to previous methods.

1 Introduction

A common field of application for sensor networks is monitoring, surveillance, and general data-gathering. As an example, we refer to the scenario where parts of the environment is observed. Such tasks include the monitoring of natural developments like glacier movements or harmful events like forest fires [15]. Positional information is a key requirement for these applications as well as for other network services such as geographic routing. For most scenarios, the use of GPS receivers to gain this information is not an option: They are expensive and clumsy compared to state-of-the-art sensor nodes. While earlier works in sensor network localization assume at least a fraction of nodes (so-called *anchors*) to know their position [13,11], research focused on anchor-free localization in recent years [4,2,8]. Without any absolute position information, these techniques can at most retrieve relative positions.

In general, localization approaches can be categorized into two groups, the first is based on distances between communicating nodes and the other is founded on relative directions. Distance information can be estimated using the Received Signal Strength Indicator (RSSI), which appears to be unreliable in practice [3], or by Time (Difference) of Arrival techniques (ToA, TDoA) requiring additional hardware. Unfortunately, finding valid embeddings for distance constraints for general and for unit-disk graphs is $\mathcal{NP}$-hard [1]. Nevertheless, most methods for localization depend on distances [4,8].

Local directions can, for example, be measured using multiple ultrasound receivers [12]. The only work considering the direction-based case of anchor-free localization we are aware of is the work of Bruck et al. [2] that presents an $\mathcal{NP}$-hardness result for (quasi-)unit-disk graphs [7] and an LP heuristic. Although the quasi-unit-disk graph communication model reflects some properties for wireless networks quite well and led to remarkable results [9], certain real-life scenarios conflict with this model [14]. Thus, we focus on localization in general networks. Here, the realization problem can be reduced to an LP and rigidity theory provides a characterization of subgraphs that have a unique realization. Uniqueness of a graph's realizability coincides with the notion of

* This work was partially supported by the German Research Foundation (DFG) within the Research Training Group GRK 1194 ”Self-organizing Sensor-Actuator Networks”, and under grant WA 654/14-3 and by EU under grant DELIS (contract no. 001907).

rigid components. For these rigid components of a network, localization with given communication directions loses most of its hardness: The localization problem reduces to to a system of linear equations for these subgraphs. There are some easy techniques to find small rigid structures in a network that work well especially in geometric graphs. It is more challenging to find a partition of a network into *maximum* such components. They could either be seen as the best possible exact localization or as the preprocessing to reduce a maximal number of variables of an LP which probably encodes some more constraints (like quasi-unit-disk graph contstraints in [2]).

We know of no algorithm to exploit the fact that small rigid substructures (or *bodies*) are easy to compute. Moukarzel [10] proposed an algorithm for identification of rigid structures in so-called *body-bar* frameworks where rigid bodies are connected by (multiple) edges. This approach, like ours, is based on an earlier work of Hendrickson [5], who developed efficient algorithms for rigidity testing in the plane, later known as the *pebble game* [6]. While the original work from Hendrickson cannot take any advantages from a rigid subgraphs that are known or easy to get, Moukarzel's approach focuses on a very special case requiring the graph to have some structural properties. We will provide an algorithm that works on general graphs and takes full advantage of known rigid substructures.

This paper is organized as follows: Section 2 summarizes preliminaries and some background work from rigidity theory. In Section 3 we will present the algorithm to find uniquely determined subgraphs modelling the problem as a flow maximization problem. We also give some results from our evaluation here. The paper is concluded in Section 4.

2 Preliminaries

Throughout this paper, we model the network topology as an undirected graph $G = (V, E)$ with a given embedding $\mathbf{p} : V \to \mathbb{R}^2$. We sometimes refer to an edge $\{v, u\}$ as (v, u) assigning an arbitrary orientation. As mentioned above, we do not assume this graph to have any specific properties except for a bounded degree and, of course, connectivity. To recover the true embedding $\mathbf{p}$, we suppose we are given the *directions* of all edges as

$$\forall (u, v) \in E\colon\ \alpha_{\mathbf{p}}(u, v) := \frac{\mathbf{p}(v) - \mathbf{p}(u)}{|\mathbf{p}(v) - \mathbf{p}(u)|}\ .$$

Alternative definitions use local angle information, i. e., for every node the angles between the incident edges, counterclockwise ordered, are known. Although this probably expresses the local availability of angular information much better, global directions can be retrieved (up to rotation) by picking an edge, defining its direction to $(1, 0)$ and propagating this decision along a spanning tree.

Even with fixed edge directions, a graph can still have several embeddings that respect these constraints. Embeddings that yield the same edge directions are called *parallel embeddings*. No set of constraints can determine a graph's embedding unambiguously in the sense that there are no parallel embeddings at all: For every $c > 0$ and $\mathbf{x} \in \mathbb{R}^2$ the embedding $\mathbf{p}'$ given by $\mathbf{p}'(v) := c\mathbf{p}(u) + \mathbf{x}$ yields the same direction constraints as $\mathbf{p}$ regardless of the edges involved. We call these embeddings *similar* that differ only by translation and scaling and say that a graph's embedding is uniquely determined, if all embeddings that have the same edge directions are similar.

The problem to find an embedding that respects the given constraints can be seen as the problem to find proper edge lengths $\ell : E \to \mathbb{R}^+$, as edge lengths together with edge directions determine an embedding up to translation. More precisely, we see that if the graph was a tree, every length assignment was valid, whereas any edge beyond that introduces additional constraints.

Lemma 1. *Let $G = (V, E)$ be a graph, $T \subseteq E$ a spanning tree. For any $e = (u, v) \in E \setminus T$ let C_e denote the unique cycle in $T \cup \{e\}$ with the same orientation as e and C_e^+ (C_e^-) the forward (backward) edges in C_e. A length assignment ℓ is compatible with edge directions α, if and only if*

$$\forall e \in E \setminus T: \quad \sum_{e \in C_e^+} \ell(e)\alpha(e) - \sum_{e \in C_e^-} \ell(e)\alpha(e) = 0\ . \tag{1}$$

Proof. Obviously, (1) holds, if ℓ is compatible with α, i. e., there is an embedding respecting α where edges have lengths $\ell(e)$. If on the other hand equation (1) holds, we can first embed the nodes V along the spanning tree, using only edge lengths and directions for edges $e \in T$. We denote the resulting embedding with $\mathbf{q}$. Now for any edge in $(u,v) \in E \setminus T$ we have $\mathbf{q}(v) - \mathbf{q}(u) + \sum_{e \in C_e^+ \setminus (u,v)} \ell(e)\alpha(e) - \sum_{e \in C_e^-} \ell(e)\alpha(e) = 0$, since this is a cycle in $\mathbb{R}^2$ and thus $\mathbf{q}(v) - \mathbf{q}(u) = \ell((u,v))\alpha((u,v))$.

From the above lemma, we can formulate the problem as a linear program with the feasible solutions being valid length assignments:

$$\begin{aligned} \text{maximize } & 1 \\ \text{subject to } & A(T,\alpha) \cdot \ell = 0 \\ & \ell \geq 0 \end{aligned} \tag{2}$$

This not only implies that valid edge lengths can be found by solving a linear program, but also that if the embedding is uniquely determined, the feasible solutions $\ker A(T,\alpha) \cap \mathbb{R}_+^{|E|}$ is $\{c\ell_{\mathbf{p}} \mid c \in \mathbb{R}_+\}^1$. That in turn means that $\ker A(T,\alpha)$ is has dimension one and thus, solving the respective homogeneous linear equation system suffices.

There are some quite obvious means to reduce the number of variables, which are also proposed similarly in [2]. They are all based on the construction of *rigid subgraphs*, node-induced subgraphs where all edge length ratios are known and which we also call *bodies*. Then, every such subgraph can be represented by one variable instead of each edge having a variable on its own in the LP formulation. Obviously, every single edge meets this definition and there are simple extensions: The first is based on the observation that edges have fixed length ratios if they form a triangle. Moreover, if for a subgraph $G(V_S)$ all edge length ratios are known and a node $u \notin V_S$ is connected to at least two nodes in V_S, all edge length ratios in $G(V_S \cup \{u\})$ can easily be calculated. We call those graphs *triangulation graphs*. If two triangulation graphs share two nodes, the union is also rigid. This also applies if they share an edge, which is a special case of node-overlapping (see Figure 1).

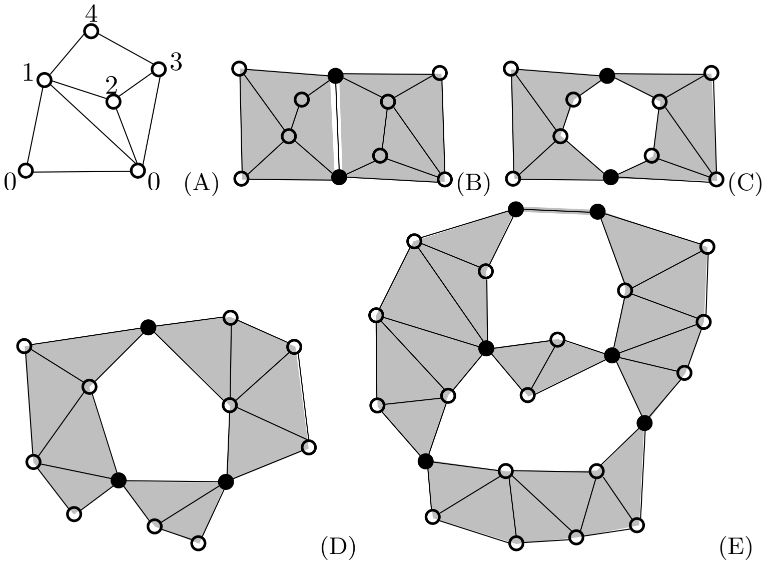

Fig. 1: *Rigid subgraphs can be found via triangulation (A). If two rigid subgraphs (grey) share an edge, the union is also rigid (B). The same holds, if two rigid subgraphs share two nodes (C). These approaches still leave more complex configurations where three (D) or more bodies form a larger rigid body (E).*

[1] Kernel $\ker A := \{x \in \mathbb{R}^m \mid Ax = 0\}$ for $A \in \mathbb{R}^{n \times m}$

To pinpoint the edge length ratios for those kinds of subgraphs is comparably inexpensive: Identifying these ratios for all maximal subgraphs constructible by triangulation and edge-overlapping for example can be done in $\mathcal{O}(n \log n)$ steps when the graph has a bounded node degree: During triangulation, positions are assigned to covered nodes, and whenever two subgraphs are merged, new positions only have to be assigned to the nodes of the smaller subgraph. For this task, the known positions of the two common nodes in both subgraphs allow to translate positions between the two subgraphs. Node-overlapping adds an effort of $\mathcal{O}(k^2)$ if triangulation and edge-overlapping leaves k subgraphs to identify overlappings.

Depending on the graph's topology, these simplifications can decrease the number of variables dramatically, but not exhaustively. Bruck et al. mention the case were rigid subgraphs that are found by edge- and node-overlappings form a larger rigid subgraph. Examples are also given in Figure 1.

This can be used to further reduce the number of variables and constraints of the LP: For every cycle that travels through multiple rigid subgraphs, one can state an equation similar to the LP constraints. If there are enough such equations, the ratios among the sizes of some subgraphs can be uniquely determined. Despite the large improvements these techniques provide for solving the LP, efficient algorithms were not stated in [2]. In addition, a characterization of subgraphs that are uniquely determined by their edge directions could be preferred. As a side-effect a topology control, which constructs a sparse topology, could more easily prune redundant connections with respect to localization.

One possible and straight-forward approach to determine maximum rigid subgraphs is as follows: Given the matrix $A(T, \alpha)$ from the LP above – possibly with redundant variables and constraints already pruned – calculate the kernel $\ell_1, \ell_2, \ldots, \ell_k$. To find out, which edges have a fixed edge length ratio to an edge (u, v) for all feasible solutions we can pick one ℓ_l with $\ell_l(u, v) \neq 0$ and rewrite the kernel vectors such that $\ell_l(u, v) = 1$ and $\ell_r(u, v) = 0$ for all $r \neq l$. Then, all edges with $\ell_r(u', v') = 0$ for all $r \neq l$ have a fixed length ratio to (u, v), namely $\ell_l(u', v')/\ell_l(u, v)$. In order to obtain a LP with a minimum number of variables and constraints (with respect to rigid components), one has to solve the homogeneous linear system of equations to identify rigid structures. This approach has the following two drawbacks: First, the time complexity is $\mathcal{O}(nk^2)$ for k rigid components and $\mathcal{O}(n)$ edges. Furthermore, the complex calculation of edge length ratios cannot be localized, i. e., performed in an incremental way. An incremental way is more adapted to the case where small groups of bodies are expected to form larger rigid components. Second, this approach relies on real-valued arithmetics not only for the definition of length ratios, but also for the identification of rigid groups. In the case of finite precision, necessary steps can easily fail due to small errors.

The next section will pick up that the above notion of rigidity corresponds to the one known from statics and computational geometry. The results developed there will allow us to derive a much faster way to identify maximum rigid subgraphs and solve (much) smaller linear equation systems decreasing the overall effort rigorously.

2.1 Rigidity Theory

The rigidity theory traditionally addresses the question whether a so-called *bar-and-joint framework*, an embedded graph $G(\mathbf{p})$ with fixed edge lengths, can be continuously deformed. Most interestingly for us, this question can be answered independently from the respective embedding as long as the embedding is guaranteed to be *generic*, i. e., there are no algebraic dependencies between the nodes' coordinates. This applies to an open, dense subset of almost all embeddings and especially when the embedding is randomly chosen with a continuous distribution, it is generic with probability 1. We will give a short outline of the main results here [16]: A (bar-and-joint) framework that can be deformed continuously without stretching or crushing the bars is said to be *flexible*; otherwise it is *rigid*. For generic embeddings, such a framework is flexible if and only if there is a non-trivial *infinitesimal motion*, a vector of velocities $\mathbf{v} \in \mathbb{R}^{2|V|}$ such that

$$\forall (i, j) \in E\colon\ (\mathbf{p}(j) - \mathbf{p}(i)) \cdot (\mathbf{v}(j) - \mathbf{v}(i)) = 0\ , \tag{3}$$

where trivial means that all nodes either have the same motion ($\mathbf{v}(i) \equiv \mathbf{x}$ for all $i \in V$), rotate around the origin ($\mathbf{v}(i) = \mathbf{p}(i)^\perp$) or combine these two motions. More formally, the constraints from (3) form a *rigidity matrix* $\mathcal{R}(G, \mathbf{p}) \in \mathbb{R}^{|E|\times 2|V|}$ and the infinitesimal motions correspond to its kernel. We call a set of edges $E' \subset E$ *independent* if their respective rows in $\mathcal{R}(G, \mathbf{p})$ are independent. As there exists a three-dimensional space of trivial motions, namely two translations and one rotation, $\mathcal{R}(G, \mathbf{p})$ has at most rank $2|V| - 3$. If it has rank $2|V| - 3$, it allows only trivial motions and is therefore rigid. As mentioned above, for generic embeddings $\mathbf{p}$, the rank of the matrix $\mathcal{R}(G, \mathbf{p})$ only depends on the underlying graph: A set of edges E' is generically independent if and only if it does not include a subset E'' with $|E''| > 2|V(E'')| - 3$. This observation is better known as *Laman's theorem*, that gives an combinatorial characterization of generically rigid graphs:

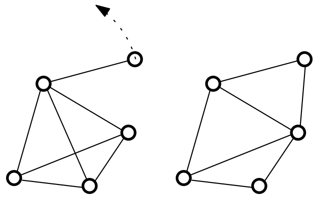

Fig. 2: *Both graphs have a sufficient number of edges to satisfy Laman's theorem, but only the right graph is rigid, since the left graph has a subgraph with an excessive edge.*

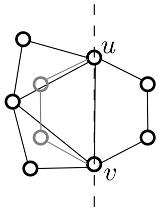

Fig. 3: *A rigid, but not 3-connected graph can have foldings when distances are constrained.*

Theorem 1 (Laman's theorem [16]). *A graph $G = (V, E)$ is generically rigid if and only if it contains a set of edges $E' \subseteq E$ with $|E'| = 2|V| - 3$ such that for all subsets $E'' \subset E'$*

$$|E''| \leq 2|V(E'')| - 3 \ .$$

Note this also implies that a generically flexible graph $G = (V, E)$ with a sufficient number of edges ($|E| \geq 2|V| - 3$) edges must either be rigid or have a subgraph $S = (V_S, E_S) \subset G$ with $|E_S| > 2|V_S| - 3$. An example is given in Figure 2.

Unfortunately, even a rigid bar-and-joint framework can have ambiguous edge lengths when discontinuous deformations are possible. If for example a graph is not 3-connected, it always allows foldings as shown in Figure 3. But there are more elaborate constructions and [5] presents some conditions necessary for generic uniqueness leaving the question open whether there is a complete characterization.

It is merely a marginal note in rigidity theory, how much easier things become when we have fixed directions instead of fixed edge lengths. These frameworks are named *parallel frameworks* (or *telescoping frameworks*). A *parallel drawing* of an embedded graph $G(\mathbf{p})$ is an embedding $\mathbf{u}$ where all edges are parallel (or anti-parallel) to the respective edge in the original embedding. This can be expressed as given in (4).

$$\forall (i,j) \in E\colon\ (\mathbf{p}(j) - \mathbf{p}(i))^\perp \cdot (\mathbf{u}(j) - \mathbf{u}(i)) = 0 \tag{4}$$

An example is given in Figure 4.

There is a one-to-one correspondence between the solutions of (3) and (4): An infinitesimal motion $\mathbf{v}$ fulfills (3) if and only if $\mathbf{u} : i \mapsto \mathbf{v}^\perp$ fulfills (4):

$$\begin{aligned} &(\mathbf{p}(j) - \mathbf{p}(i)) \cdot (\mathbf{v}(j) - \mathbf{v}(i)) \\ =\ &(\mathbf{p}(j) - \mathbf{p}(i))^\perp \cdot (\mathbf{v}(j) - \mathbf{v}(i))^\perp \\ =\ &(\mathbf{p}(j) - \mathbf{p}(i))^\perp \cdot \left(\mathbf{v}(j)^\perp - \mathbf{v}(i)^\perp\right) \end{aligned}$$

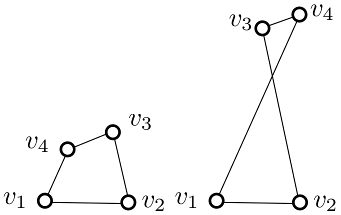

Fig. 4: *Parallel drawings have parallel or anti-parallel edges.*

Again, we have three dimensions of trivial solutions, spanned by the scaled versions of $\mathbf{p}$ and the shifted embeddings $\mathbf{p}_x : i \mapsto \mathbf{p}(i) + (1,0)$ and $\mathbf{p}_y : i \mapsto \mathbf{p}(i) + (0,1)$. These correspond to the trivial solutions of (3) as per description. Thus, a generic framework has a non-trivial parallel drawing if and only if it is flexible. But those parallel drawings have edge lengths that meet the constraint (2) of the LP except for the non-negativity (anti-parallel edges have negative lengths). More precisely, if and only if there is a non-trivial parallel drawing, the solution to $A(T,\alpha) \cdot \ell = 0$ is at least 2-dimensional, since parallel drawings whose edge lengths differ only by a constant scaling are all trivial and vice versa.

Thus, the following theorem holds:

Theorem 2. *A generic embedding $\mathbf{p} : V \to \mathbb{R}^2$ of a graph $G = (V,E)$ is uniquely determined up to translation and scaling by its edge directions $\alpha_{\mathbf{p}}$ if and only if G contains a set of edges $E' \subseteq E$ with $|E'| = 2|V| - 3$ such that for all subsets $E'' \subset E'$*

$$|E''| \leq 2|V(E'')| - 3 \ .$$

As rigidity coincides with uniqueness of the parallel drawings in the plane, we will stick to the term of rigid subgraphs as used in [2] and above. In three dimensions however, the advantages of edge directions over lengths for our purposes become even clearer: Here we still lack a combinatorial characterization of generic rigidity whereas theory for parallel drawings can easily be extended to any dimension: Theorem 2 holds analogously for embeddings $\mathbf{p} : V \to \mathbb{R}^d$ by replacing (**??**) by $(d-1)|E''| \leq d|V(E'')| - (d+1)$ [16]. As a consequence, the following approach works similarly for the three-dimensional case.

3 Maximum Rigid Components

We have seen that Laman's theorem for rigid graphs is also a complete characterization of graphs where edge directions determine the embedding uniquely. Although there are algorithms to identify rigid components of a graph, they cannot take advantage of known rigid substructures, which can be found by much less complex methods as mentioned in the previos section.

3.1 Fundamentals

In this section, we will present an algorithm to obtain maximum rigid components from a given graph G which is already partitioned into rigid components, for example using edge- and node-overlappings. By a partition, we here refer to a set of rigid subgraphs that cover all of G's edges disjointly but may have common nodes. We call this a *Laman partition* (see Figure 5):

Definition 1 (Laman partitions). *Let $G = (V,E)$ be a simple undirected graph and $\mathcal{S}$ be a set of* pairwise edge-disjoint*, generically rigid subgraphs. Then*

1. *The* partition graph $G(\mathcal{S}) := (V(\mathcal{S}), E(\mathcal{S}))$ *is defined as the union of the rigid components graphs, i. e.,*

$$V(\mathcal{S}) := \bigcup_{(V,E)\in\mathcal{S}} V \qquad E(\mathcal{S}) := \bigcup_{(V,E)\in\mathcal{S}} E \ .$$

The set $\mathcal{S}$ is also called a Laman partition *(of $G(\mathcal{S})$). It is* rigid*, if $G(\mathcal{S})$ is rigid and it is* independent*, if there is no $\mathcal{S}' \subset \mathcal{S}$ which is rigid*[2]*.*

2. *The redundancy of a node $v \in V$ is defined as $\texttt{rd}_{\mathcal{S}}(v) := |\{(V,E) \in \mathcal{S} \mid v \in V\}| - 1$. The notion is extended to rigid partitions by $\texttt{rd}(\mathcal{S}) := \sum_{v \in V(\mathcal{S})} \texttt{rd}_{\mathcal{S}}(v)$. We denote the redundantly used nodes as $R(\mathcal{S}) := \{v \in V(\mathcal{S}) \mid \texttt{rd}_{\mathcal{S}}(v) > 0\}$.*
3. *The surplus of edges in a graph $H = (V', E')$ with respect to Laman's theorem is denoted by $\texttt{sp}(H) := |E'| - 2|V'| + 3$. We will also write $\texttt{sp}(\mathcal{S})$ for $\texttt{sp}(H(\mathcal{S}))$. Note that a graph H has at most $E - \texttt{sp}(H)$ independent edges.*

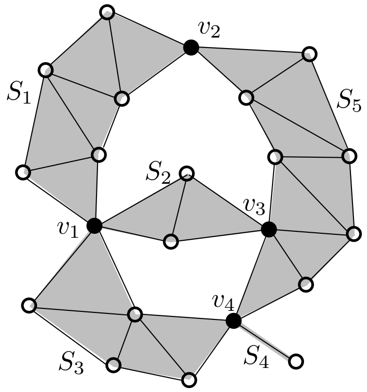

Fig. 5: *A Laman partition with five bodies. Nodes have redundancies $\texttt{rd}_{\mathcal{S}}(v_1) = \texttt{rd}_{\mathcal{S}}(v_4) = 2$, $\texttt{rd}_{\mathcal{S}}(v_2) = \texttt{rd}_{\mathcal{S}}(v_3) = 1$, but, for example, $\texttt{rd}_{\mathcal{S}'}(v_4) = 1$ for $\mathcal{S}' = \{S_1, S_2, S_3, S_5\}$. Theorem 3: The surplus of $\mathcal{S}'$ is $\texttt{sp}(\mathcal{S}') = 2 \cdot \texttt{rd}(\mathcal{S}') - 3(|\mathcal{S}'| - 1) = 2 \cdot 5 - 3(4-1) = 1$. $\{S_1, S_2, S_3\}$ is independent and no subset $\mathcal{S}'' \subseteq \mathcal{S}'$ with $S_5 \in \mathcal{S}''$ has more surplus.*

The simplest Laman partition of a graph is a partition into $|E|$ graphs which all consist of exactly one edge. Although this will work as well, we assume that in most scenarios, we have significantly less rigid bodies. Without loss of generality, we assume furthermore that in a Laman partition every graph $S = (V, E)$ has exactly $|E| = 2 \cdot |V| - 3$ edges (i.e., $\texttt{sp}(S) = 0$), which are all independent. Since these graphs are rigid, they contain such a set of edges and probably some more which we simply ignore.

The approaches from Hendrickson and Moukarzel have in common that they manage a growing set of independent edges. Due to the matroidal character of the problem, an edge can greedily be chosen to join this set if there is no dependency to present edges. Rigid areas of the network can be identified *en passant*. When talking about rigid bodies, we lose some of this ease, since a subgraph can have both, edges that are important for rigidity as well as excessive ones. But the greedy approach still works: If we go through the bodies of a Laman partition and merge bodies as soon as there are bodies that form a larger rigid structure, we end up with a partition into maximum rigid components. Unfortunately, it is not sufficient to look for bodies, that together have a sufficient number of edges. This would for example apply to the set of all subgraphs in Figure 5 ($\texttt{sp}(\{S_1, \ldots, S_5\}) = 0$), but whereas the bodies S_1, S_2, S_3, S_5 have one edge more than needed, the size ratio of S_4 to the other subgraphs is not fixed.

We start with the observation that Laman partition with sufficiently overlapping bodies must have enough edges to fulfill Laman's theorem:

Lemma 2. *Let $\mathcal{S}$ be a rigid partition. Then $\texttt{sp}(\mathcal{S}) = 2 \cdot \texttt{rd}(\mathcal{S}) - 3 \cdot (|\mathcal{S}| - 1)$.*

Proof. As the graphs in a rigid partition have disjoint edge sets, the edges of $G(\mathcal{S})$ just sum up as $|E(\mathcal{S})| = \sum_{(V,E) \in \mathcal{S}} |E| = \sum_{(V,E) \in \mathcal{S}} (2 \cdot |V| - 3)$, whereas the nodes were counted $\texttt{rd}_{\mathcal{S}}(v) + 1$

[2] Note, that independency here differs slightly from the notion of independent edges

times. Thus, $|V(\mathcal{S})| = \sum_{(V,E)\in\mathcal{S}} |V| - \mathtt{rd}\,(\mathcal{S})$ holds which results in the following equation:

$$\begin{aligned}\mathtt{sp}\,(\mathcal{S}) &= |E(\mathcal{S})| - 2\cdot|V(\mathcal{S})| + 3\\ &= \sum_{(V,E)\in\mathcal{S}} (2\cdot|V|-3) - 2\cdot\left(\sum_{(V,E)\in\mathcal{S}} |V| - \mathtt{rd}\,(\mathcal{S})\right) + 3\\ &= 2\cdot\mathtt{rd}\,(\mathcal{S}) - 3\cdot(|\mathcal{S}|-1)\ .\end{aligned}$$

From the remark to Laman's theorem follows that a Laman partition $\mathcal{S}$ with $\mathtt{sp}\,(\mathcal{S}) \geq 0$ at least contains a rigid subset. Adapting the iterative scheme, we will use the following theorem to maintain an independent rigid partition merging bodies whenever a rigid subset appears:

Theorem 3. *Let $\mathcal{S}$ be a rigid partition and $S^\star \in \mathcal{S}$ such that $\mathcal{S} - S^\star$ is independent. Then $\mathcal{S}' \subseteq \mathcal{S}$ is rigid if and only if for all non-empty $\mathcal{S}'' \subseteq \mathcal{S}'$ that contain $S^\star$ the inequality $\mathtt{sp}\,(\mathcal{S}') \geq \mathtt{sp}\,(\mathcal{S}'')$ holds.*

Proof. First assume that $\mathcal{S}'$ is rigid. If there was any subset $\mathcal{S}''$ of $\mathcal{S}'$ with $\mathtt{sp}\,(\mathcal{S}') < \mathtt{sp}\,(\mathcal{S}'')$, one could not choose $|E(\mathcal{S}')| - \mathtt{sp}\,(\mathcal{S}')$ edges from $E(\mathcal{S}')$ without choosing more than $|E(\mathcal{S}'')| - \mathtt{sp}\,(\mathcal{S}'')$ from $E(\mathcal{S}'')$. Therefore, any $2|V(\mathcal{S})| - 3$ edges from $G(\mathcal{S})$ cannot be independent.

If on the other hand for all $\mathcal{S}'' \subset \mathcal{S}'$ with $S^\star \in \mathcal{S}''$ the inequality $\mathtt{sp}\,(\mathcal{S}') \geq \mathtt{sp}\,(\mathcal{S}'')$ holds, then we know that $\mathtt{sp}\,(\mathcal{S}') \geq 0$, as it holds for all graphs, i. e., $\mathtt{sp}\,(H) = 0$ for all $H \in \mathcal{S}$. Suppose that $\mathcal{S}$ was not rigid. According to Laman's theorem, there must be a rigid subgraph $G' = (V', E') \subsetneq G(\mathcal{S}')$ with $|E'| > 2|V(E')| - 3$. This graph G' spans over at least 2 graphs in $\mathcal{S}$ which also form a rigid graph with at least one dependent edge. All those non-trivial rigid subsets $\mathcal{S}''$ include $S^\star$; thus their union $\mathcal{S}^{\max}$ forms the unique maximal rigid subgraph $G(\mathcal{S}^{\max})$. But we're able to choose $|E(\mathcal{S}')| - \mathtt{sp}\,(\mathcal{S}')$ edges from $E(\mathcal{S}')$ even if we restrict ourselves to take only a set of independent edges from $E(\mathcal{S}^{\max})$ where we only have to leave out $\mathtt{sp}\,(\mathcal{S}^{\max}) \leq \mathtt{sp}\,(\mathcal{S}')$. These $2|V(\mathcal{S}')| - 3$ edges are either all independent, so that $\mathcal{S}'$ must be rigid, or there still is a subgraph with $G' = (V', E') \subsetneq G(\mathcal{S}')$ with $|E'| > 2|V(E')| - 3$ which is not covered by $\mathcal{S}^{\max}$. Both cases are inconsistent with either the assumptions or the definition of $\mathcal{S}^{\max}$.

In the above example (Figure 5), the surplus of $\mathcal{S}' = \{S_1, S_2, S_3, S_5\}$ is $\mathtt{sp}\,(\mathcal{S}') = 2\cdot\mathtt{rd}\,(\mathcal{S}') - 3(|\mathcal{S}'| - 1) = 2\cdot 5 - 3(4-1) = 1$. $\{S_1, S_2, S_3\}$ is independent and no subset $\mathcal{S}'' \subseteq \mathcal{S}'$ with $S_5 \in \mathcal{S}''$ has more surplus. The detection of subsets with this property is not trivial. We present an efficient algorithm to solve this task by formulating it as a maximum-flow problem.

Definition 2. *For a rigid partition $\mathcal{S}$ and a particular graph $S^\star \in \mathcal{S}$ such that $\mathcal{S} - S^\star$ is an independent rigid partition, the bipartite* intersection network *$B(\mathcal{S}, S^\star) = (R(\mathcal{S}), \mathcal{S}, A, \kappa, b)$ is given by*

$$A = \{(v, G) \in R(\mathcal{S}) \times \mathcal{S} \mid v \in G\} \qquad \kappa \equiv 2$$
$$b(v) = 2\cdot \mathtt{rd}_{\mathcal{S}}\,(v) \qquad b(G) = \begin{cases} 3 : G \neq G^\star \\ 0 : G = G^\star \end{cases}$$

A flow then is a function $f : A \to \mathbb{N}$ with $f(a) \leq \kappa(a)$ and

$$b_f(v) := b(v) - \sum_{(v,G)\in A} f(v, G) \geq 0$$
$$b_f(G) := b(G) - \sum_{(v,G)\in A} f(v, G) \geq 0\ .$$

Definition 3. *Let $\mathcal{S}$ be a Laman partition, $S^\star \in \mathcal{S}$ such that $\mathcal{S} - S$ is an independent Laman partition and f a maximal flow in $B(\mathcal{S}, S^\star)$. Then a subset $\mathcal{S}'$ is called* saturated *if and only if $\forall S \in \mathcal{S}' : \sum_{(v,S)} f(v, S) = b(S)$ and* closed *if and only if*

$$\forall (v, S), (v, S') \in A\colon\ S \in \mathcal{S}' \wedge f(v, S) > 0 \wedge f(v, S') < 2 \implies S' \in \mathcal{S}'\ ,$$

i.e., there is no path from a contained graph to one that isn't by traversing edges in the residual network. For any set of graphs $\mathcal{S}'$, the (minimal) closure is denoted by $\overline{\mathcal{S}'}$. Analogously, the closure of a set of nodes $R \subseteq R(\mathcal{S})$ is defined as $\overline{R} := \overline{\{G \in \mathcal{S} \mid f(v, G) < 2\}}$.

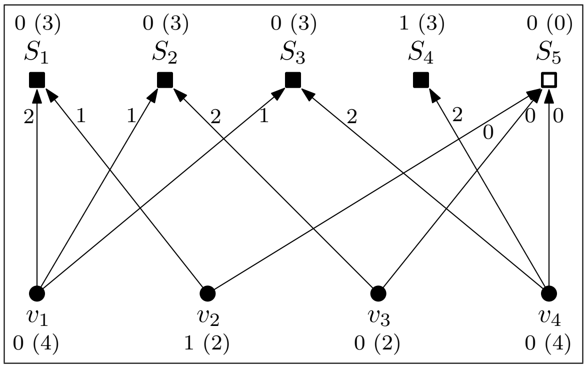

Fig. 6: *The intersection network of the graph from Figure 5 (with $S^\star = S_5$) with a maximum flow f. Nodes and graphs are annotated with $b_f(\cdot)$ $(b(\cdot))$, edges with $f(\cdot)$.*

For example, in Figure 6, the intersection network of the example 5, $\{S_1, S_2, S_3, S_5\}$ is a maximum closed and saturated subset. $\{S_2, S_3, S_5\}$ is a smaller closed and saturated set whereas $\{S_1, S_3, S_5\}$ is not, as S_2 can be reached from S_1 by traversing (v, S_1) and (v, S_2).

The following two lemmas (Lemma 3 and 4) ensure that for a maximum flow firstly any saturated and closed set is rigid, and secondly as long as a rigid set is contained, there has to be a saturated and closed set:

Lemma 3. *If for any valid flow f in a rigidity network $B(\mathcal{S}, S^\star)$ there is a saturated, closed set of graphs $\mathcal{S}'$ then the following properties hold:*

1. *The flow to $\mathcal{S}'$, $\sum\limits_{(v,S)\in A, S\in\mathcal{S}'} f(v,S')$, is $2\boldsymbol{\mathtt{rd}}(\mathcal{S}') - b_f(R(\mathcal{S}))$.*
2. *The graph $S^\star$ is contained in $\mathcal{S}'$, i. e., $S^\star \in \mathcal{S}'$.*
3. *The set $\mathcal{S}'$ is rigid.*

Proof. We prove these properties one at a time:

1. As $f(v, S) = 2$ holds for all $v \in \mathcal{S}'$ and $S \notin \mathcal{S}'$, we obtain the following equalities:

$$\begin{aligned}\sum_{(v,S)\in A, S\in\mathcal{S}'} f(v,S) &= \sum_{v\in R(\mathcal{S}')} \left(\sum_{(v,S)\in A} f(v,S) - \sum_{(v,S)\in A, S\notin\mathcal{S}'} 2 \right)\\ &= \sum_{v\in R(\mathcal{S}')} \left(2\mathtt{rd}_{\mathcal{S}}(v) - b_f(v) - \sum_{v,S\in A, S\notin\mathcal{S}'} 2 \right)\\ &= \sum_{v\in R(\mathcal{S}')} \left(2\mathtt{rd}_{\mathcal{S}'}(v) - b_f(v) \right)\\ &= 2\mathtt{rd}(\mathcal{S}') - b_f(R(\mathcal{S}'))\ .\end{aligned}$$

2. As the flow saturates the graphs in $\mathcal{S}'$, $\sum_{(v,S)\in A, S\in\mathcal{S}'} f(v,S') \geq 3(|\mathcal{S}'|-1)$. Thus, $2\mathtt{rd}(\mathcal{S}') \geq 2\mathtt{rd}(\mathcal{S}') - b_f(R(\mathcal{S}') = 3(|\mathcal{S}|-1)$. Therefore, $\mathcal{S}'$ at least contains a rigid subset, which then must contain $S^\star$.
3. With theorem 3 it is sufficient to show that for all subsets $\mathcal{S}'' \subseteq \mathcal{S}'$ that include $S^\star$ $\mathtt{sp}(\mathcal{S}') \geq \mathtt{sp}(\mathcal{S}'')$. For a closed, saturated subset $\mathcal{S}'' \subseteq \mathcal{S}'$ is $\mathtt{sp}(\mathcal{S}'') = b_f(R(\mathcal{S}''))$ since (see considerations above)

$$\begin{aligned} 3(|\mathcal{S}''|-1) &= \sum_{(v,S)\in A, S\in\mathcal{S}''} f(v,S) = 2\mathtt{rd}(\mathcal{S}'') - b_f(R(\mathcal{S}''))\\ \iff \underbrace{3(|\mathcal{S}''|-1) - 2\mathtt{rd}(\mathcal{S}'')}_{=-\mathtt{sp}(\mathcal{S}'')} &= -b_f(R(\mathcal{S}''))\ .\end{aligned}$$

For a saturated, but not necessarily closed set $\mathcal{S}'' \ni S^\star$ this becomes $\mathtt{sp}\,(\mathcal{S}'') \leq b_f(R(\mathcal{S}''))$. Therefore, when $\mathcal{S}'$ is a saturated, closed subset with respect to f, and $\mathcal{S}'' \subset \mathcal{S}'$ such that $S^\star \in \mathcal{S}''$ the following inequality holds:

$$\mathtt{sp}\,(\mathcal{S}'') \leq b_f(R(\mathcal{S}'')) \leq b_f(R(\mathcal{S}')) = \mathtt{sp}\,(\mathcal{S}') \quad .$$

Lemma 4. *Let $\mathcal{S}$ be a rigid partition, $S^\star \in \mathcal{S}$ such that $\mathcal{S} - S^\star$ is independent. If $\mathcal{S}$ contains a non-trivial rigid subset and $\mathcal{S}'$ is an inclusion-maximal rigid subset, then for any maximum flow in $B(\mathcal{S}, S^\star)$, $\mathcal{S}'$ is saturated and closed.*

Proof. Let $\mathcal{S}'$ be a non-trivial, inclusion-maximal rigid subset of $\mathcal{S}$. As all rigid subsets overlap in $S^\star$, $\mathcal{S}'$ is well-defined as the union of all rigid subsets of $\mathcal{S}$. Suppose, $\mathcal{S}'$ was not closed or saturated with respect to a maximum flow f. Then $b_f(R(\mathcal{S}')) > \mathtt{sp}\,(\mathcal{S}')$ and therefore $R_f = \{v \in R(\mathcal{S}') \mid b_f(v) > 0\}$ must be non-empty. But the closure $\mathcal{S}'' := \overline{R_f}$ is saturated. As $\mathcal{S}''$ is rigid, $\mathcal{S}'' \subseteq \mathcal{S}'$. Furthermore, by this choice we assure that $b_f(\mathcal{S}'') = b_f(\mathcal{S}')$. However, this contradicts with $b_f(\mathcal{S}'') = \mathtt{sp}\,(\mathcal{S}'') \leq \mathtt{sp}\,(\mathcal{S}') < b_f(R(\mathcal{S}'))$.

3.2 Implementation

Together, the Lemma 3 and 4 are the foundation for our algorithm that finds maximum rigid components starting with a Laman partition $\mathcal{S}$. It is given in pseudo-code in Algorithm 1.

Algorithm 1: MergeRigidComponents($\mathcal{S}$)

```
    $\mathcal{S}^I \leftarrow \emptyset$;
    while $\mathcal{S} \neq \emptyset$ do
        choose $S^\star$ from $\mathcal{S}$;
        $\mathcal{S} \leftarrow \mathcal{S} - S^\star$;
A       while $\exists S \in \mathcal{S}^I$: $|V(S^\star) \cap V(S)| > 1$ do
            $\mathcal{S}^I \leftarrow \mathcal{S}^I - S$;
            $S^\star \leftarrow G(\{S, S^\star\})$;
B       $f \leftarrow$ maximum flow in $B(\mathcal{S}^I \cup \{S^\star\}, S^\star)$;
        $\mathcal{S}' \leftarrow$ maximum closed and saturated set with respect to $f$;
        if $|\mathcal{S}'| > 1$ then
            $S^\star \leftarrow G(\mathcal{S}')$;
            $\mathcal{S}^I \leftarrow \mathcal{S}^I \setminus \mathcal{S}'$;
        $\mathcal{S}^I \leftarrow \mathcal{S}^I \cup \{S^\star\}$;
```

First, this algorithm clearly ensures $\mathcal{S}^I$ to be the unique partition into maximum rigid subgraphs. As an invariant, $\mathcal{S}^I$ is independent: Before we add a graph $S^\star$ to $\mathcal{S}^I$, we find the maximum rigid subset of $\mathcal{S}^I \cup \{S^\star\}$, remove the involved graphs from $\mathcal{S}^I$ and add the graph formed by the subset to $\mathcal{S}^I$. For this to hold, we don't need the steps marked with A, which will only play an important role for the analysis. Second, this algorithm runs in $\mathcal{O}(n + l \log l + k^2)$ for $k := |\mathcal{S}|$ and $l := |R(\mathcal{S})|$. We first iterate over all graphs in $\mathcal{S}$ and all contained nodes to find the nodes from $R(\mathcal{S})$ and to annotate the graphs with their respective intersection nodes. With a bounded node degree of Δ, this can be done in $\mathcal{O}(n)$ as no node can be part of more than Δ edge-disjoint graphs. This annotation can be kept up-to-date during merging operations by processing only the annotations of the smaller graph (in terms of intersection nodes). This can be done with an overall effort of $\mathcal{O}(l \log l)$ steps. Now we have k iterations of the outer 'while'-loop. For every $S^\star \in \mathcal{S}$, we first test, whether there is a graph in $\mathcal{S}^I$ which has two nodes with $S^\star$ in common. This check can at most be performed $2k - 1$ times over all, k times failing (once for every $S^\star$) and at most $k - 1$ times succeeding and combining two graphs, i. e., reducing the overall number of graphs. For such a check, at most k intersection nodes must be considered. The k-th intersection node at the latest is the second common node with one of the other graphs. The analysis of the second ('B'-) part of the algorithm is more cumbersome. We therefore first analyze the structure of $B(\mathcal{S}^I \cup \{S^\star\}, S^\star)$. Every node in $R(\mathcal{S}^I \cup \{S^\star\})$ has at least two incident edges in A. Less than $3/2|\mathcal{S}^I| - 3/2$ of them can have more than two edges to graphs in $\mathcal{S}^I$, as every such node has $\mathtt{rd}_{\mathcal{S}^I}(v) > 0$ and $\mathcal{S}^I$ is independent. On the other hand, nodes with only one edge to graphs in $\mathcal{S}^I$ must have an edge to $S^\star$. This can only apply to at most $|\mathcal{S}^I|$ nodes. Thus, we have less than $5/2|\mathcal{S}^I| - 3/2 \in \mathcal{O}(k)$ intersection nodes. Similarly, as only $|\mathcal{S}^I|$ edges can be incident to $S^\star$, and for every node all but one incident edge corresponds to a redundant use, we have $\mathtt{rd}\left(\mathcal{S}^I\right) > |A| - |\mathcal{S}^I| - 5/2|\mathcal{S}^I| - 3/2$. Since $\mathtt{rd}\left(\mathcal{S}^I\right) < 3/2|\mathcal{S}^I| - 3/2$ holds, we also get $|A| < 2 \cdot |\mathcal{S}^I| \in \mathcal{O}(k)$. Furthermore we know, that a maximum flow can have at most a value of $3|\mathcal{S}^I|$. Naïvely implemented, this still could lead to a complexity of $\Theta(k^2)$ per solved maximum flow problem. Fortunately, there is an easy way to re-use solutions from the preceding iteration. If f_i is a valid flow in $B(\mathcal{S}^I_i, S^\star_i)$, the intersection network of the ith iteration, then a valid flow for the network $B(\mathcal{S}^I_{i+1}, S^\star_{i+1})$ can be constructed in $\mathcal{O}(k)$ by

$$f_{i+1}(a) = \begin{cases} f_i(a) & : \text{if } a \in A_i \\ 0 & : \text{else} \end{cases} .$$

This flow f_{i+1} cannot violate any of the conditions as

- no edge a has $f_{i+1}(a) > 2$ if this held for f_i,
- no graph $S \in \mathcal{S}^I_{i+1}$ gets more flow than 3 if no graph $S \in \mathcal{S}^I_i$ did, and
- every node $v \in R(\mathcal{S}^I_{i+1} \cup \{S^\star_{i+1}\})$ is either included in the same set of graphs as in the i-th iteration and has therefore the same value for $b(v)$ which is not violated by f_{i+1}, or it must be in $S^\star_{i+1}$. In this case, there is an edge $a = (v, S^\star_{i+1}) \in A_{i+1}$ with $f_{i+1}(a) = 0$. It then must have $b_f(v) \geq 0$ by construction.

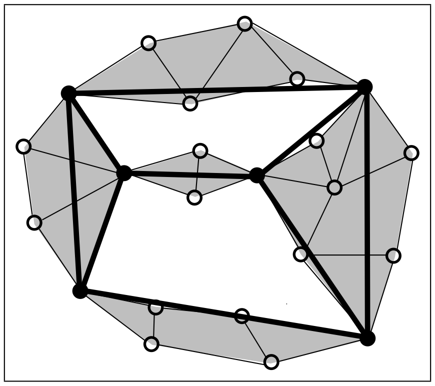

Fig. 7: *Virtual edges used to determine edge length ratios.*

Although the changes of $\mathcal{S}^I$ look quite complex, we only have k additions of a new graph, and by the re-use of flows, the flow accepted by any graph is non-decreasing. Therefore, we have at most $3k$ successful augmenting steps and k failing tests in flow maximization over all, which can then be done in $\mathcal{O}(k^2)$.

While the above algorithm finds maximum rigid components of a graph, it does not maintain unique valid embeddings for these components. If valid embeddings are known for the graphs in $\mathcal{S}$, it is sufficient to calculate embeddings whenever $S^\star$ is replaced by either the union of two graphs in the case of two overlapping nodes (A) or the union of a rigid set of graphs $\mathcal{S}'$ found in section B. In both cases, we have to calculate length ratios of the involved graphs first. This is quite easy for the union of two graphs $S^\star, S'$ with overlapping nodes u and v. Here, the ratio is simply $|\mathbf{p}_{S^\star}(u) - \mathbf{p}_{S^\star}(v)|/|\mathbf{p}_{S'}(u) - \mathbf{p}_{S'}(v)|$.

When more graphs ($k' := |\mathcal{S}'|$) are involved, we first observe that we can only have $\mathcal{O}(k')$ intersection nodes. This follows analogously to the considerations above: $\mathcal{S}'$ always consists of a set of independent graphs $\mathcal{S}' \cap \mathcal{S}^I$, which can only have $\mathcal{O}(k')$ intersection nodes among them, and a fresh graph $S^\star$, which can have at most one node in common with every graph in $\mathcal{S}'$.

To determine the correct length ratios, it is sufficient to consider only a set of "virtual" edges for every graph: If a graph S contains l intersection nodes $u_1, \ldots, u_l$, we can treat it like the complete graph on these nodes and restrict ourselves to a maximum independent subset of the edges, for example the edges $\{u_1, u_2\} \cup \{\{u_i, u_j\} \mid i \leq 2, j > 2\}\}$ (see Figure 7). This means that for a body with l intersection nodes we introduce less than $2l$ edges. As every intersection node can only be part of a constant number of bodies we have an overall of $\mathcal{O}(k')$ edges to consider. The following steps, choosing a spanning tree T, solving the homogeneous linear equation system from (2) and assigning consistent coordinates to the intersection nodes then take $\mathcal{O}({k'}^3)$. As in every merging step with k' graphs the number of graphs decreases by $k' - 1$, this sums up to at most $\mathcal{O}(k^3)$ since

$$\sum_i k_i^3 \leq \left(\sum_i (k_i - 1) + 1 \right)^3 \quad \text{for } k_i \geq 2 \ . \tag{5}$$

Here, the left-hand side is an upper bound for the complexity of solving i subproblems, whereas the right-hand side corresponds to the complexity of solving the whole linear equation system (up to constant factors). This is of course in a worst-case scenario (where $k = m$ and the graph $G(\mathcal{S})$ is minimally rigid) as bad as determining the kernel of $A(G(\mathcal{S}), T, \alpha)$ as mentioned in the preliminaries, but in most scenarios, the number of bodies left by triangulation, edge- and node-overlapping is much smaller than the number of edges, and usually the iterative approach identifies smaller rigid sets of graphs reducing the number of graphs stepwise. Note that for decreasing size (and increasing number) of subproblems, the gap in (5) increases dramatically.

3.3 Evaluation on Quasi-Unit-Disk Graphs

The presented algorithm has been evaluated on random quasi unit-disk graphs with 10000 nodes and different node densities. Here, the density refers to the average number of nodes per unit square. Node positions were chosen uniformly distributed from the square area $[0, \sqrt{n/d}]^2$ for n nodes with a density d. Edges have been introduced for all node pairs with a distance of less

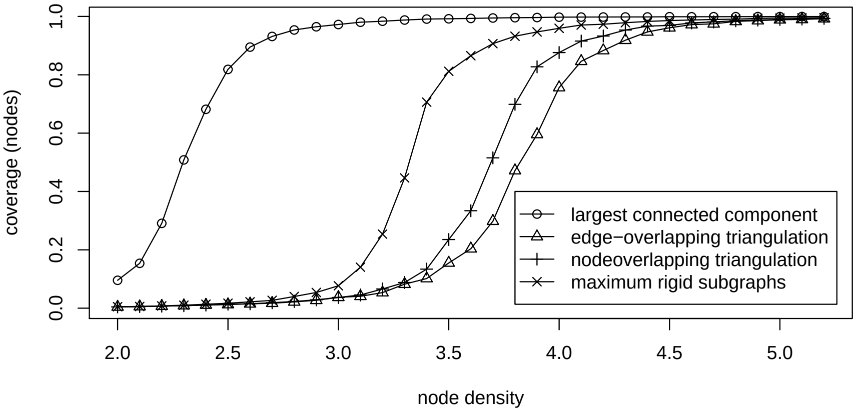

Fig. 8: *The coverage of the largest component to be localized by edge- and node-overlapping triangulation and the identification of maximum rigid components compared with the largest connected component.*

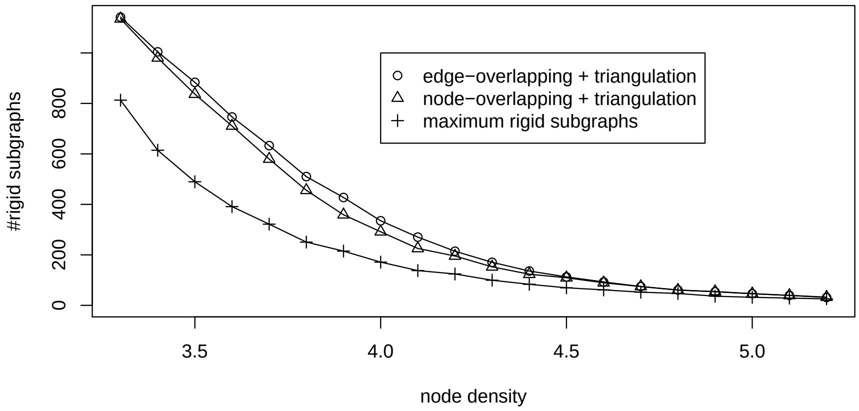

Fig. 9: *Number of rigid subraphs left after standard techniques and identification of maximum rigid subgraphs.*

than 0.5 and for node pairs with a distance $0.5 < r < 1.0$ with probability $2 \cdot (1 - r)$. The results of these tests are shown in Figure 8. For quasi-unit-disk graphs, the critical density for direction-based localization is significantly smaller when maximum components are used instead of node- and edge-overlapping triangulation only; without using Linear Programming heuristics, a node density of 3.5 was sufficient to localize more than 80% of the nodes. Similarily, when the approach is used as a preprocessing to the technique of Bruck et al., the number of variables for the respective LP can be reduced by a factor of 2 with respect to node- and edge-overlappings for certain node densities. Our tests furthermore showed, that the iterative approach identified almost only very small groups of bodies that formed larger rigid areas. Hence the computational bottleneck from determining matrix kernels played no role in these settings.

4 Conclusion

In this paper we presented an approach for direction-based localization in sensor networks, which can be applied to general networks and systematically exploits rigidity theory. Unlike for distance-based localization, this theory provides a full characterization of rigid network structures that are sufficient for this task and can be extended to the $\mathbb{R}^3$. Our approach is an advancement and generalization of the technique [2] for direction-based localization. Our algorithm not only considers node- and edge-overlapping components but also identifies maximum rigid components. Thus, the size of the associated LP can be drastically reduced up to a factor of 2 with respect to node- and edge-overlappings. Furthermore, if the network exceeds a certain density, a large percentage of the network can be localized without considering the LP at all. The iterative approach in almost all scenarios reduces the complexity by applying the costly operations only to necessary and in most cases very small subproblems.

References

1. J. Aspnes, D. Goldenberg, and Y. Yang. On the computational complexity of sensor network localization. In *Proceedings of the First International Workshop on Algorithmic Aspects of Wireless Sensor Networks*, 2004.
2. J. Bruck, J. Gao, and A. Jiang. Localization and routing in sensor networks by local angle information. pages 181–192, New York, NY, USA, May 2005. ACM Press.

3. Quentin Lindsay Dimitrios Lymberopoulos and Andreas Savvides. An Empirical Characterization of Radio Strength Variability in 3-D IEEE 802.15.4 Networks Using Monopole Antennas. In *Third European Workshop on Wireless Sensor Networks*, February 2006.
4. C. Gotsman and Y. Koren. Distributed graph layout for sensor networks. In *Proceedings of the 12th International Symposium on Graph Drawing (GD'04)*, volume 3383 of *Lecture Notes in Computer Science*, pages 273–284, 2005.
5. B. Hendrickson. Conditions for unique graph realizations. *SIAM J. Comput.*, 21(1):65–84, 1992.
6. D. Jacobs and B. Hendrickson. An algorithm for two dimensional rigidity percolation: The pebble game, 1997.
7. F. Kuhn, R. Wattenhofer, and A. Zollinger. Ad-hoc networks beyond unit disk graphs. In *DIALM-POMC '03: Proceedings of the 2003 Joint Workshop on Foundations of Mobile Computing*, pages 69–78, New York, NY, USA, 2003. ACM Press.
8. D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 50–61, New York, NY, USA, November 2004. ACM Press.
9. T. Moscibroda, R. O'Dell, M. Wattenhofer, and R. Wattenhofer. Virtual coordinates for ad hoc and sensor networks. In *DIALM-POMC '04: Proceedings of the 2004 joint workshop on Foundations of mobile computing*, pages 8–16, New York, NY, USA, 2004. ACM Press.
10. C. Moukarzel. An efficient algorithm for testing the generic rigidity of graphs in the plane. In *J. Phys. A: Math. Gen.*, volume 29, page 8079, 1996.
11. D. Niculescu and B. Nath. Dv based positioning in ad hoc networks. In *Telecommunication Systems, Volume 22, Issue 1 - 4, Pages 267 - 280*, 2003.
12. N. B. Priyantha, A. K. L. Miu, H. Balakrishnan, and S. Teller. The cricket compass for context-aware mobile applications. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 1–14, New York, NY, USA, 2001. ACM Press.
13. A. Savvides, C.-C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179, New York, NY, USA, 2001. ACM Press.
14. Stefan Schmid and Roger Wattenhofer. Algorithmic Models for Sensor Networks. In *14th International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS), Island of Rhodes, Greece*, April 2006.
15. M. Tubaishat and Sanjay Madria. Sensor networks: An overview. In *IEEE Potentials, 22(2)*, pages 20–23, 2003.
16. W. Whiteley. Matroids from discrete applied geometry. In *Matroid Theory, AMS Contemporary Mathematics*, pages 171–311, 1996.