DIMACS Series in Discrete Mathematics
and Theoretical Computer Science

Combinatorial Roadmaps in Configuration Spaces of Simple Planar Polygons

Ileana Streinu

ABSTRACT. One-degree-of-freedom mechanisms induced by minimum pseudo-triangulations with one convex hull edge removed have been recently introduced by the author to solve a family of non-colliding motion planning problems for planar robot arms (open or closed polygonal chains). They induce canonical roadmaps in configuration spaces of simple planar polygons with fixed edge lengths.

While the combinatorial part is well understood, the search for efficient solutions to the algebraic components of the algorithm is posing a number of interesting questions, some of which are addressed in this paper. A list of open problems and further research topics on pointed pseudo-triangulations and related structures motivated by this work is appended.

1. Introduction

In this paper we address aspects of a computational algebraic nature arising in the Pseudo-Triangulation Roadmap Algorithm introduced by the author in [**20**] to solve the Carpenter's Rule Problem. We formulate the problems and discuss several possible solutions, together with the theoretical and computational challenges they induce and related open questions and conjectures.

The **Carpenter's Rule Problem.** Consider a simple planar polygonal chain (linkage) with fixed edge lengths (*robot arm*). Orient it so that the interior lies to the left when walking along the polygon in the positive direction. The edges (*bars*) are allowed to move freely around the vertices (*joints*). We want to avoid collisions between the bars while moving the linkage continuously from an initial to a final configuration with the same orientation. It suffices to show that we can move from any position to a convex polygon position. Then, to move between any two configurations, take one path in reverse (it is easy to move between two distinct convex positions). That this is always possible was shown by Connelly, Demaine

1991 *Mathematics Subject Classification.* Primary 68U05, 65D18; Secondary 52C25, 52C40, 52C30, 52C45, 05C10, 13P10.

Key words and phrases. Computational Geometry, Rigidity Theory, Computational Algebraic Geometry.

Research supported by NSF RUI grant CCR-0105507 and NSF-DARPA CARGO CCR-0138374.

and Rote [**8**], based on Rote's ground-breaking idea of using *expansive infinitesimal motions* to avoid collisions.

For finding algorithmically a path in configuration space between any two compatible positions, the author has proposed in [**20**] a combinatorial approach: the path consists of a finite number of arcs, each being the unique trajectory of a one-degree-of-freedom mechanism induced by a *pointed* (or *minimum*) *pseudo-triangulation*.

Expansive Motions and Pseudo-Triangulation Mechanisms. A one-degree-of-freedom (1DOF) mechanism is a bar-and-joint framework whose configuration space is a one-dimensional curve. The mechanism is said to be *(infinitesimally) expansive* at some point in its configuration space, if all the pairwise interdistances between its vertices simultaneously increase or decrease (including the possibility of some staying the same), when the mechanism is moved infinitesimally along its unique trajectory in configuration space. Expansive motions guarantee that no collisions will occur. It is shown in [**20**] that pseudo-triangulations with one convex hull edge removed (defined in section 2) are (infinitesimally) expansive one-degree-of-freedom mechanisms.

The infinitesimally expansive motions form a cone ([**8**] and [**19**]), and linear programming can be used to find a set of infinitesimal velocities of the moving points. Pseudo-triangulations with a convex hull edge removed correspond to canonical basic feasible solutions found by such a linear program. They can be computed very efficiently geometrically, without using linear programming. Our algorithmic approach ([**20**]) is to start with a mechanism induced by a pseudo-triangulation and move it until a special event occurs (alignment of two edges). This invalidates the expansive property, should the motion continue. Therefore, at that point, the mechanism is locally reconfigured to get a new pseudo-triangulation and the motion continues. After a finite number of such steps, the convex position is attained.

Results. In this paper we investigate the algebraic components of this algorithm. Two general problems which await efficient and accurate numerical solutions are discussed: the **Simulation of Motion** for a pseudo-triangulation mechanism, and the **Detection of the Next Event.** We exhibit some special cases which are satisfactorily solved. These depend on a certain inductive construction for pseudo-triangulations introduced in [**20**] and modelled after the Henneberg constructions of [**14**] for generically minimally rigid graphs (see [**12**], [**23**], [**21**]).

Organization. The paper is organized as follows. To make the paper self-contained, in section 2 we introduce the necessary concepts and give an overview of the pseudo-triangulation roadmap algorithm of [**20**]. In section 3 we state the two problems addressed in this paper and formulate them algebraically. In section 4 we discuss some of the drawbacks of using standard techniques from computational algebraic geometry in solving them. In section 5 we include proofs and heuristics based on rigid-components and in section 6 we describe heuristics based on Henneberg constructions. Open problems, conjectures and research topics (both combinatorial and algebraic) related to this approach are included in almost every section, and additional ones are gathered in section 7.

2. Definitions and Preliminaries

This largely self-contained section introduces the basic terminology, definitions, problems and relevant previous results.

References. For rigidity theory terminology and classical results, we refer the reader to [**18**], [**23**], [**24**] and [**12**]. In particular, rigidity, first-order and generic rigidity, as well as classical results on 2-dimensional rigidity such as Laman's theorem and the Henneberg constructions are to be found there. For computational algebraic geometry terminology and results, see [**9**] and [**10**]: Gröbner bases and the homotopy method are described there.

Basic concepts, notations and abbreviations. Our setting is the Euclidian plane. We denote by $\mathcal{P} = \{p_1, \cdots, p_n\} \subset R^2$ a finite set of planar points. When it is a function of a time parameter t we may denote it as $P(t) = \{p_1(t), \cdots, p_n(t)\}$. We abbreviate *counter-clockwise* as *ccw* and *one-degree-of-freedom mechanism* as *1DOF mechanism*.

Basic Definitions. A *pseudo-triangle* is a simple planar polygon with exactly three convex vertices (its *corners*). These are joined by three inner convex polygonal chains called its *side chains*. See Figure 1.

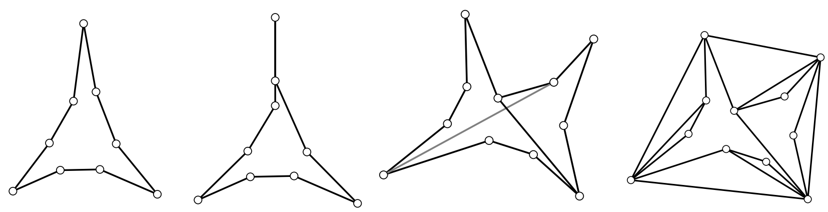

FIGURE 1. A pseudo-triangle, a semi-simple pseudo-triangle, a flip in a pseudo-quadrilateral and a pointed (minimum) pseudo-triangulation.

A set of vectors $v_1, \cdots, v_k \in R^2$ is *pointed* if there is no linear combination with not-all-zero positive coefficients summing them up to zero. Such vectors lie in a half plane. Equivalently, among the k angles spanned by pairs of cyclically consecutive vectors, one is *strictly* larger than π (reflex). [1]

A *framework* (G, L) is a graph $G = (V, E)$, $V = \{1, \cdots, n\}$, together with a given set of edge lengths $L = \{l_{ij} | ij \in E\}$. A *configuration* or *realization* of (G, L) is a set of points $P = \{p_1, \cdots, p_n\}$ such that if the vertices of G are mapped to P, $i \in V \mapsto p_i \in P$, then the length of the line segment $\overline{p_i p_j}$ equals l_{ij} if $ij \in E$. The notation $G(P)$, referring to both G and P, is used to denote the realization of G on the set of points P. Questions of realizability of a particular set of edge lengths will not be of interest to us here, since we already start with a set of realizable lengths via an initial embedding $G(P)$ of the graph on a planar point set.

The *configuration space* of (G, L) is the set of all its possible realizations. If non empty, this space can be factored to the reorientation of the plane, as each realization comes with a mirror image which is itself realizable. It also contains a 3-dimensional subspace corresponding to rigid plane transformations (translations and rotations). We will later make specific choices that will induce this factorization and then we will be concerned only with questions of realizability in the quotient space. For the particular case of planar polygons, if we factor out the rigid motions, it is well known that their configuration spaces are $(n-3)$-dimensional manifolds,

[1]The terminology used in [**20**] for the same concept was *acyclic* for reasons related to the oriented matroids motivation.

with the sole exception when there exists a configuration where all the vertices lie on a line, see [**15**].

An embedding is *planar* or *non-crossing* if any pair of non-incident edges (ij and kl, $i, j \not\in \{k, l\}$) are disjoint as line segments in the embedding ($\overline{p_i p_j} \cap \overline{p_k p_l} = \emptyset$). In the particular case when G is a path or a cycle the embedding is a planar polygonal path or a polygon. Instead of "non-crossing", a polygon will be called *simple*. A pseudo-triangle is a special case of a simple polygon. A *semi-simple pseudo-triangle* may have two overlapping edges adjacent to one of the corners, i.e. one or more of the internal convex angles of the corners may be zero. See Fig. 1. [2]

Pointed Pseudo-Triangulations. A *pseudo-triangulation* is a tilling of the convex hull of a planar point set with pseudo-triangles. In other words it is an embedded planar graph whose outer face is the complement of the convex hull and each interior face is a pseudo-triangle. A *minimum pseudo-triangulation* uses the minimum number of edges. A *pointed pseudo-triangulation* has an angle larger than π adjacent to every vertex, i.e. the edge vectors around each vertex form a pointed set. See Fig. 1. Minimum and pointed pseudo-triangulations turn out to be the same object. They have remarkable combinatorial and rigidity theoretic properties, see [**20**]. For the sake of completeness, we reproduce here the relevant ones, which will be used implicitly throughout the paper.

THEOREM 2.1. [**20**] *Let $G(P)$ be a graph embedded on a set of n points, satisfying any one of the following* 6 *properties. Then it satisfies all the others.*

(1) *pointed pseudo-triangulation*
(2) *minimum pseudo-triangulation*
(3) *maximally non-crossing and pointed*
(4) *pseudo-triangulation with* $2n - 3$ *edges*
(5) *non-crossing, pointed with* $2n - 3$ *edges*
(6) *admits a non-crossing and pointed Henneberg construction (described below)*

From now on, we will work only with pointed pseudo-triangulations and refer to them, for conciseness, simply as *pseudo-triangulations*. A pseudo-triangulation has $2n-3$ edges and since both planarity and pointedness are hereditary properties, each subset of k vertices spans at most $2k - 3$ edges. This hereditary $(2n - 3)$-property characterizes *generically minimally rigid graphs* and is known as the *Laman property* ([**16**], see [**18**]). Graphs satisfying Laman's property are called *Laman graphs*. Minimality means that removing any edge causes the (generic) framework to move, hence it becomes a mechanism. Laman graphs are infinitesimally rigid in *almost all* possible embeddings, called *generic embeddings*.

In general, removing $k \geq 1$ edges from a minimally infinitesimally rigid graph creates an infinitesimal mechanism with k degrees of freedom, i.e. k is the dimension of its space of infinitesimal motions. The configuration space of a 1DOF mechanism is a curve obtained by integrating the infinitesimal motions and is *in general* a manifold. It may have several connected components. On each component, one may choose a parametrization and move in exactly two directions on the closed curve. The distance between at least one pair of points $p_i p_j$ changes: it either increases or decreases strictly. A mechanism is *(locally or infinitesimally) expansive*

[2] Semi-simple pseudo-triangles appear in the Pseudo-Triangulation Roadmap Algorithm 2.3 at the beginning of each step except the first.

for some position in its configuration space if all the interdistances between pairs of points simultaneously increase (or stay the same), or simultaneously decrease (or stay the same) (infinitesimally). If this property holds at a generic position, then it holds on an open neighborhood: the mechanism is *expansive* as it moves along that portion of its configuration space.

THEOREM 2.2. [**20**] *Pseudo-triangulations are always minimally infinitesimally rigid (and hence rigid). A pseudo-triangulation with a convex hull edge removed is an (infinitesimally) expansive 1DOF mechanism. As it moves on its component in configuration space, it remains expansive as long as the underlying combinatorial pseudo-triangulation structure doesn't change.*

The definition of a combinatorial pseudo-triangulation will be given a bit later in this section.

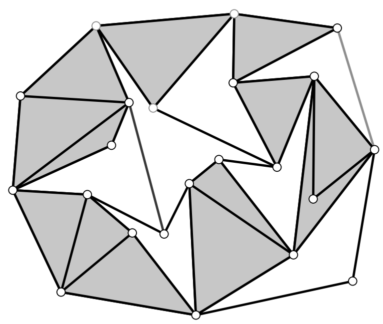

FIGURE 2. The rigid components of a pseudo-triangulation mechanism.

We will refer to a pseudo-triangulation with a convex hull edge removed as a *pseudo-triangulation mechanism*. Like any Laman graph with an edge removed, a pseudo-triangulation mechanism decomposes into *maximally rigid components*, or *r-components*: maximal subsets of some k vertices spanning exactly $2k-3$ edges. See [**12**]. This *r-decomposition* has a nice structure, [**20**]: each component is a pseudo-triangulation of a convex subset of points (i.e. a subset whose convex hull does not include points outside the set). The interesting components have at least 3 vertices, although we can view the remaining edges also as r-components. These are bitangents between various subcomponents, in the sense used in [**17**] for pseudo-triangulations of the free space among convex obstacles. Figure 2 gives an example of such an r-decomposition.

Theorem 2.1 (3) implies that every set of pointed edges can be extended to a pointed pseudo-triangulation. In particular, the edges of a simple planar polygon form a pointed set of edges. Any extension of it to a pseudo-triangulation, which will be referred to from now on as a *pseudo-triangulation of the polygon*, has the property that the underlying graph contains a Hamiltonian cycle (the polygon). In general, define a *Hamiltonian pseudo-triangulation* to be a pseudo-triangulation whose underlying graph is Hamiltonian. This is to be distinguished from what is usually referred to in the literature as a (pointed) *polygon pseudo-triangulation*, which is a (pointed) decomposition of the *interior* of a polygon into pseudo-triangles.

To *flip* an interior edge in a pseudo-triangulation ([**19**]) means to remove it, thus merging two faces into one, and then to add the *unique* edge different from the removed one that would pseudo-triangulate again this merged face. See Fig. 1.

Henneberg constructions for pseudo-triangulations. As a generalization of the known Henneberg constructions for Laman graphs ([**14**], [**21**], see also [**12**]), pseudo-triangulations can be constructed inductively as follows (cf. [**20**]).

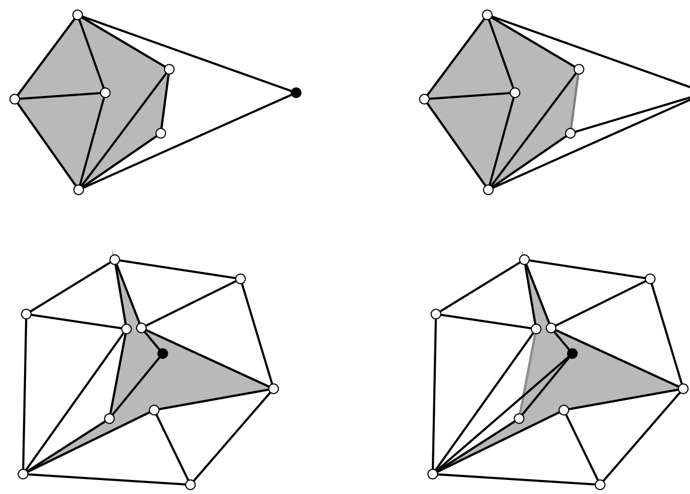

FIGURE 3. Henneberg construction for pseudo-triangulations. Left: type I, right: type II. On the right, the flipped edge is still slightly visible for reference, but it does not belong to the pseudo-triangulation. The new vertex is added on the outside face (top) or on an inner face (bottom). The two extreme edges adjacent to the new vertex are tangent to the convex hull (top, shaded) or to one or two of the inner chains of a pseudo-triangular face (bottom, shaded).

Start with the base case of a triangle. At each step, add a vertex v in one of the existing faces F. There are two types of steps (see Figure 3):

- **Henneberg I**: add two tangents from v to side chains of the face. This splits the face into two faces.
- **Henneberg II**: perform a Henneberg I step and choose the newly created face where the new vertex is a corner. Then pick up one edge on the side chain of this face which is visible from the corner (there always exists one), and flip it to obtain a third edge incident to v.

A Henneberg I step creates a new vertex of degree 2 and a Henneberg II step creates a vertex of degree 3. A pseudo-triangulation which can be obtained by applying only type I steps is called a *Henneberg I pseudo-triangulation.* [3] It must have at least one vertex of degree 2. When the edge lengths are known, they can be constructed inductively using only ruler and compass.

Henneberg steps can be generalized to the expansive 1DOF mechanisms obtained from pseudo-triangulations using the same types of steps. A 1DOF mechanism which can be obtained by applying only type I steps is called a *Henneberg I mechanism*, and can be constructed inductively with ruler and compass only.

[3]In the Rigidity Theory literature, Henneberg I graphs (general, not pseudo-triangulation) are called 2-simple.

Combinatorial Pseudo-Triangulations. Pseudo-triangulations are plane graphs (i.e. embedded planar graphs) with additional (partial) oriented matroid information for pointedness. Any plane embedding of a planar graph induces a *topological embedding* of the underlying graph G. This is independent of point coordinates, and captures only the combinatorial information about the faces of the embedding and their adjacencies. Equivalently, this information is captured by the *rotations* at each vertex: the ccw circular order of the adjacent edges at each vertex in the embedding. A *topological embedding* of a planar graph is such a system of rotations (or equivalently, the faces and their adjacencies) which can be realized in the plane with non-crossing curves (pseudo-segments). A *topological embedding with a marked face* contains, in addition, information about which face of the embedding is unbounded (the *outer* or *exterior* face).

An assignment of a label $\{C, R\}$ (standing for *convex* and *reflex*) to each angle of a face F of a topologically embedded planar graph is called a *combinatorial (pointed) pseudo-triangulation* if:

(1) For each vertex, exactly one of its adjacent angles is labelled R.
(2) There exists one face (the *outer face*) whose angles are all labelled R.
(3) All other faces have exactly three angles labelled C.

Combinatorial pseudo-triangulations are relevant to our work because they provide a combinatorial way of distinguishing among solutions to algebraic equations describing embeddings, with given edge lengths, of Laman graphs. In particular, the desired solutions for both the Local and the Global Problems described in Section 3 have a known combinatorial type.

The Pseudo-Triangulation Roadmap Algorithm. The properties of pseudo-triangulations induce the following simple approach for constructing roadmaps in configuration spaces of simple planar polygons.

Algorithm 2.3. (The pseudo-triangulation roadmap algorithm)

(1) *Construct a pseudo-triangulation of the polygon.*
(2) *Choose a convex hull edge which is not one of the original edges of the polygon and remove it to obtain an expansive mechanism.*
(3) *Move the mechanism along its unique trajectory which increases the distance between the endpoints of the removed edge.*
(4) *Stop when the framework is no longer pointed: at some vertex, the two extreme adjacent edges align. Call this an* alignment event*, or simply an* event.
(5) *Perform a local flip to restore the pseudo-triangulation (at least one pseudo-triangle will be semi-simple at this moment) and continue the motion using a new expansive mechanism induced by it.*

How to perform the flips to restore the pseudo-triangulation and how to control it so that the total number of such readjustments is kept small is shown in [**20**]. In particular, for a scheme based on the shortest-path tree from a vertex, the algorithm will end after at most $\mathcal{O}(n^3)$ such events.

A *shortest-path pseudo-triangulation* of a polygon is obtained as follows. Take the convex hull of the polygon. This induces a number of *pockets*: the complements with respect to the convex hull of the interior of the polygon. Each pocket is bounded by a simple polygon. Inside each polygon (the original and the pockets) add the edges of the shortest path tree from an inner convex vertex to all the other inner convex vertices of the polygon. It is shown in [**20**] that this produces a pointed pseudo-triangulation.

In the sequel we are interested only in how to perform steps 3 (Simulation of Motion) and 4 (Detection of next Event).

3. The Algebraic Components of the Pseudo-triangulation Roadmap Algorithm

In addition to the combinatorial part described above, a complete implementation of the algorithm involves the choice of a time parametrization of the motion and algebraic computations in steps 3 and 4, as follows.

- **Parametrization.** Parametrize the motion of one pseudo-triangulation mechanism between two consecutive events, so that, if t_0 and t_f are the start and finish times, δt is the time step, then the motion of the mechanism is simulated as a sequence of frames, one for each time $t = t_0 + k\delta t$, $t_0 \leq t \leq t_f$.
- **Simulation of the Motion of a Pseudo-Triangulation Mechanism.** Given a time step t, $t_0 \leq t \leq t_f$ and the embedding $P(t)$ at time t, find the coordinates $P(t + \delta t)$ of the embedded mechanism $G(P(t + \delta t))$ at time $t + \delta t$.
- **Detection of the Next Event.** For each mechanism corresponding to one step of the Algorithm, compute the final value $t = t_f$ of the time parameter (the time of the next alignment event) and the index k of the vertex at which the next alignment event happens.

In this section we formulate these problems in algebraic terms.

Parametrization. Originally our systems of equations will be thought of as having $2n$ variables $x_1, \cdots, x_n, y_1, \cdots, y_n,$ standing for the unknown positions of the points $p_i = (x_i, y_i)$ at some moment in time t. For the Local Problem, t is a constant and for the Global Problem it is a variable.

To factor out the trivial plane motions assume without loss of generality (i.e after a suitable relabelling) that the convex hull edge missing from the pseudo-triangulation is the edge between vertices 1 and 2. After a rigid transformation of the system of coordinates, assume that the origin is at vertex 1, the edge itself lies on the x-axis in the positive direction and the second endpoint is constrained to move along the axis from the initial position $t = t_0$ to increasing values. This parametrization adds to each system the extra equations $x_1 = 0, y_1 = 0, y_2 = 0,$ and uses x_2 is the *time parameter*. Substituting in all equations x_1, y_1 and y_2 by 0 and using x_2 as time parameter t we are reduced to $2n - 3$ variables.

A reminder that all happens during the lifetime of only one mechanism. The parametrization may change after an event, because it is possible that another convex hull edge will be chosen to be dropped when the pseudo-triangulation is locally reconfigured. In addition, sometimes it may happen that the expanding (missing) convex hull edge is “crossed” by some other part of the pseudo-triangulation during the motion of the mechanism: while this does not invalidate the expansive property, it may be necessary to readjust the parametrization to record that a different edge is now the missing convex hull edge, so that we maintain the format of the equations and their desired invariant properties.

Simulation of the Motion of a Pseudo-Triangulation Mechanism and the **Local Problem.** The values t_0 and t_f are computed before the simulation starts: t_0 is simply the x-coordinate of the second point p_2 at the beginning of the simulation, and t_f is computed as part of the *Detection of the Next Event*. At each step, our goal is to compute the new coordinates of the vertices, from which (say) a drawing is produced by superimposing the edges. The $2n-4$ edge lengths of the pseudo-triangulation mechanism are fixed. Since we have chosen a parametrization where the length of the removed convex hull edge at time t is t, it follows that at each time step we have a full set of $2n-3$ edge lengths for a minimally rigid framework (Laman graph), and we want to compute the *realization* of this framework which is the *closest* (in a sense that must be defined satisfactorily) to the previous instance. In particular, it must be a pseudo-triangulation with the same combinatorial structure as the one at the previous step.

Writing down the edge length constraints of these edges we get a system of $2n-3$ quadratic equations of two types:

- $2n-4$ quadratic equations, stating that the $2n-4$ edges ij of the mechanism have their prescribed lengths:

$$(x_i - x_j)^2 + (y_i - y_j)^2 = l_{ij}^2 \tag{3.1}$$

- The additional equation:

$$x_2 = t \tag{3.2}$$

This last equation corresponds to the length condition for the convex hull edge. We can use this simplified, linear version, which quadratically would be $(x_1-x_2)^2+(y_1-y_2)^2 = t^2$ (but we eliminated $x_1 = y_1 = y_2 = 0$). In this case, the value t is a constant, more precisely it is equal to $t_0 + k\delta t$ at the $k+1$st step of the simulation.

We now formulate the **Local Problem:** *Find the solution of the algebraic system (3.1)+(3.2) which corresponds to the correct position of the pseudo-triangulation mechanism at time t.*

The algebraic system (3.1)+(3.2) will be called a *Laman graph embedding system*, because the underlying graph of n vertices and $2n-3$ edges is a Laman graph: equation (3.2) amounts to adding back to the pseudo-triangulation mechanism the convex hull edge that was removed.

A Laman graph system has $2n-3$ quadratic equations in $2n-3$ unknowns. It may have exponentially many real solutions (see [**5**]): we are interested in exactly one of them, corresponding to the next position of the mechanism. This is the unique point in the configuration space of the mechanism which lies on the expansive part of the trajectory at time step t. Since a good starting point for an iterative solution is always available, as we already know the solution $G(P(t))$ to an algebraic system that is *close* to the one we want to solve at time $t+\delta t$, the natural approach is to use the Homotopy method ([**10**], [**22**]). This will compute *only one solution* (the one closer to the previous instance) via numerical iterative methods (e.g. Newton's method). The idea is simple and implementable and we have experimented with it for our algorithm. We found that it works very well unless one gets close to the alignment event, when sometimes (and not rarely) it fails to converge in a reasonable, pre-defined (as a global constant) number of iterations or it gives a wrong answer (such as a solution *on the other side of the alignment event*). We have noticed that sometimes this happens when complex solutions would be obtained, should the motion extend past the alignment event. But these phenomena await a more systematic investigation.

Open Problem 1. *Investigate the nature of those alignment events which cause Newton's method to fail as one gets close to them, and try to find good combinatorial or algebraic predictors for such a performance.*

On the theoretical side and based on our experimentation in Mathematica with this method, we expect the following to be true.

Conjecture 2. *For a generic instance of the Simulation of Motion problem (where* generic *will have to be defined appropriately, see section 4), there exist time moments $t_Z \geq t_0$ and $t_F \leq t_f$ sufficiently close to the event moments t_0 and t_f, such that if $t_Z \leq t \leq t_F$, Newton's method produces reliable solutions. I.e. in a* polynomially bounded (in n) *number of iterations, it computes a solution (point coordinates) lying on the trajectory of the pseudo-triangulation mechanism.*

The following stronger conjecture may facilitate this proof. It is also related to an alternate approach to the Local Problem based on Elimination: compute all the solutions and select *the* particular solution having the desired combinatorial pseudo-triangulation structure. For this to work, the following must be true.

Conjecture 3. *Among the many possible embeddings of a planar Laman graph with given edge lengths, no two are pseudo-triangulations with the same underlying combinatorial pseudo-triangulation structure.*

If this is true, then the problem could be formulated in semi-algebraic terms by adding inequality constraints to distinguish the solution that corresponds to the desired pseudo-triangulation. These can be expressed easily semi-algebraically as convex/reflex constraints on the angles between consecutive edges in the topological embedding of the planar graph. They are similar to *order type* ([**11**]) or *oriented matroid* (see [**2**]) constraints. Indeed, the combinatorial pseudo-triangulation concept is just *partial* oriented matroid information. For instance, for two edges ik and jk around a vertex k, the condition for convexity of the ccw oriented angle $\angle ikj$ is captured by a **semi-algebraic** constraint of the form:

$$\det \begin{pmatrix} x_i & y_i & 1 \\ x_k & y_k & 1 \\ x_j & y_j & 1 \end{pmatrix} < 0 \tag{3.3}$$

Assuming that Conjecture 3 holds, this formulation of the Local Problem becomes in principle solvable using techniques from Semi-Algebraic Geometry [**4**], perhaps by making use of combinatorial properties specific to our problem. We have not investigated this possibility yet. Instead we will discuss some possible heuristic approaches in section 5.

A weaker version of Conjecture 3 is:

Conjecture 4. *Among the many possible embeddings of a planar Laman graph with given edge lengths, no two are pseudo-triangulations with the same underlying oriented matroid.*

It is conceivable that two embeddings may have distinct oriented matroids, but that they would differ in the orientation of triplets of points that would not affect a common underlying combinatorial pseudo-triangulation. Thus while Conjecture 3 implies Conjecture 4, the opposite may not be true.

A *side remark*: to the best of our knowledge, there are very few such results of a *global rigidity* nature. One of the most conspicuous ones is the famous Cauchy Rigidity proof ([**7**], see [**1**]).

We end the paragraph by mentioning an example which shows that Conjecture 3 would fail if we dropped *combinatorial* and asked instead just for the uniqueness of a *pseudo-triangulation* embedding. Indeed, the two pseudo-triangulations in Figure 4 have identical edge lengths and differ *only* in the combinatorial part (assignment of convex/reflex angles), not in the topological embedding as planar graphs.

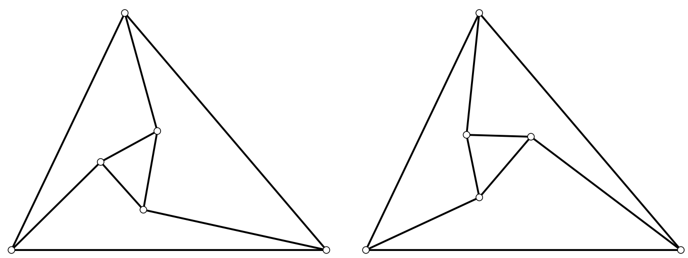

FIGURE 4. Example of a framework with two distinct pseudo-triangulation embeddings differing only in the combinatorial information of which vertices are corners.

Detection of the Next Event. We will refer to the following as the **Global Problem**: *For a pseudo-triangulation mechanism in expansive motion, compute the final value $t = t_f$ for the time parameter (the time of the next event) and the index k of the vertex at which the next alignment event happens.*

We first formulate the problem algebraically to show that it contains several instances of the Local Problem. We write down n systems of equations capturing the potential alignment of each vertex. They will be called *alignment systems* and have a common part, which expresses the fact that the $2n-4$ edge lengths of the mechanism are fixed. They differ in one extra equation expressing the alignment of the extreme edges at the vertex.

- The $2n-4$ quadratic *edge length* equations are the same as (3.1).
- The *alignment event equation* at vertex k expresses the condition that the (constant) sum of the lengths of the two extreme edges ik and jk of k equals the (unknown) distance between the other endpoints i and j:

$$(x_i - x_j)^2 + (y_i - y_j)^2 = (l_{ik} + l_{jk})^2 \tag{3.4}$$

Thus the algebraic part of each alignment system is similar to the Local Problem, except that we replaced an equation corresponding to a convex hull edge with an equation corresponding to an edge between the endpoint vertices i and j of the extreme edges ik and jk of vertex k. We will show in section 5 that, if i and j are not in the same rigid component, the underlying graph is still a Laman graph (but not a pseudo-triangulation, since it misses a convex hull edge and has one non-pointed (aligned) vertex). We will also show that we may not need to solve all n systems, since for some of them we can decide a priori that they have no solution.

The challenge of the *Global Problem* is to distinguish between the solutions to the alignment events systems produced by solving each system in turn: which corresponds to the next event?

The nature of the problem suggests the use of the Homotopy method. We can consider using it in two ways:

(1) As an iterative method for solving each alignment system, given as starting point of the iteration the position of the mechanism at time t_0. The drawback is that, in this case, the solution we seek may *not* be the closest to the one we start with. In practice, we have observed that an implementation of this approach found sometimes solutions that indeed align the vertices, but going in the opposite of the expansive direction! However, if we assume that somehow we get, for each system, the *closest* solution *in the direction of the motion* of the mechanism, then it is easy to detect the next event: it corresponds to the value k of the vertex index which produced a solution (alignment event) at the *smallest* value of the time parameter t. Indeed, after this time all the other alignments will contain k as a non-pointed vertex, corresponding to the fact that the mechanism moved past its aligning event.
(2) As an iterative method applied in conjunction with the Simulation of Motion. Instead of providing a final value for t_f, we compute it as we move along. At each step in the Simulation, we must check whether we have reached an Alignment event. This can be done by *verifying* whether equation 3.4 holds for the current values. This can only be done if we know the root values *exactly* (which is not a practical assumption except for some special situations presented in Section 5) and if we happen to use the right time step to fall exactly on the event. Most of the times, we will be either before or after the event.

Here are the main questions that must be researched for efficient solutions to the Homotopy method applied to the Global Problem.

Given an embedded Laman graph, verify whether it is a pseudo-triangulation. Moreover, verify that it has a given combinatorial structure. We call this the **Pseudo-Triangulation Verification Problem.**

Since this may have to be done at each step of the simulation, we want it to be very fast.

Open Problem 5. *What is the complexity of the Pseudo-triangulation Verification Problem? Can it be done in linear time?*

We might get better solutions under the assumption that we *know* something about which edges might cross or which vertices might become non-pointed, since we are using it within the Simulation of Motion phase for a specific pseudo-triangulation.

Open Problem 6. *Is it possible to detect in sublinear time the violation of pseudo-triangulation properties if we already know the Laman graph structure and (possibly) additional information (such as the infinitesimal velocities of the vertices at the previous time step t of the simulation)?*

Another problem with this approach is caused by the fact that the time step δt might be too big: we might accidentally pass several alignment events, and even if we detect this, we may not see which one we passed. When the final value t_f is known, the trivial choice is to simply divide the interval $t_f - t_0$ into a constant number of steps, but with this approach we must choose δt in another way: either some constant, or a value to be decided dynamically, as we move along the trajectory.

Open Problem 7. *Develop criteria for choosing the value of the time step to avoid that we accidentally pass over events and to guarantee that we make progress towards the next event.*

Alternately, we might want to solve the following problem instead.

Open Problem 8. *Develop criteria for recognizing that the Simulation passed over several events.*

If these problems would be solved, the Homotopy method might be the right approach in practice, as it will lead directly to the aligned vertex. We are still interested, though, in the possibility of a semi-algebraic formulation of the Global Problem, as well as in the use of non-homotopy based methods (e.g. Elimination). This leads to an extension of Conjecture 3, formulated for the *combinatorial type* of what would be the Laman graph behind each alignment event. We skip the details here, as this definition is easy to formulate and is a simple variation on the combinatorial pseudo-triangulation.

Conjecture 9. *Among the many possible embeddings of a planar Laman graph with given edge lengths, no two have the same underlying combinatorial structure, which is the "almost pseudo-triangulation" structure of the underlying graph of an alignment event.*

In addition, we need:

Conjecture 10. *Assuming that no two vertices align simultaneously, exactly one of the Alignment systems will have a solution compatible with the combinatorial structure of the pseudo-triangulation mechanism before the alignment event.*

We skip the details of how this compatibility is defined, as it is straightforward and the reader can reconstruct it easily. If this conjecture holds, the **semi-algebraic** formulation of the *Global Problem* is similar to what we had for the local problem, i.e. using orientation constraints of the type expressed by the equations (3.3).

On an even more ambitious level, one may ask whether the partial oriented matroid information of *any* Laman graph embedding suffices to guarantee the uniqueness of the embedding.

Open Problem 11. *Investigate whether the partial oriented matroid information of a Laman graph embedding suffices to guarantee uniqueness.*

4. Problems Encountered with Standard Techniques for Solving Algebraic Equations

In this section we identify three problems that may be encountered when using one or the other of the two approaches for solving algebraic systems of equations, Elimination or Homotopy. Our examples will show why using blindly some standard functions already available in computational algebraic systems (such as Mathematica) may not work or may be very inefficient (exponential). This will motivate the need for using heuristics, such as those presented in section 6, and will provide intuitions to the related sequence of open questions and problems requiring further investigation.

4.1. Problems with Elimination. Elimination (Gröbner bases) is a notoriously time intensive algorithmic approach, but in our case it would have the advantage of better controlled precision. We would like to understand when it would or would not work, and leave as a future goal to seek techniques for fine-tuning it to the specifics of the problem to possibly get efficiency and the desirable degree of accuracy.

We first prove that a quadratic system of equations modelling the algebraic part of the *Local Problem*, i.e. the realizability problem for a Laman graph, may not have a finite set of solutions, and when it has one, it may be exponentially large.

THEOREM 4.1. *There exists a pseudo-triangulation, and even a Hamiltonian pseudo-triangulation, whose configuration space has a component of arbitrarily large dimension.*

Proof: Based on the examples in Fig. 5. The underlying graph of the image on the left is Laman (a Henneberg I graph, in fact). The given embedding is generic (non-infinitesimally rigid Henneberg I graphs occur only when a vertex of degree 2 is aligned), but the edge lengths are very special. All the vertices (except the two at the bottom) lie on the perpendicular bisector of the segment joining the two bottom ones. Taking the mirror image of the right half of this embedding in the axis of symmetry containing the vertical vertices produces another realization, where the points and edges on the right overlap over those on the left hand side. It is now obvious that all the vertical points can be moved continuously and (except for one of the two top connected ones) independently, while keeping the edge lengths fixed, to obtain other embeddings. The dimension of the component of the realization space containing this embedding is very large ($n-3$).

An example of a Hamiltonian pseudo-triangulation with a similar symmetry inducing a one-dimensional component of the configuration space is given in the rightmost picture of Fig. 5. □

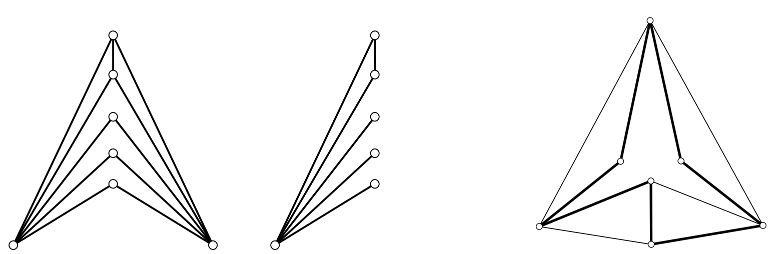

FIGURE 5. Example of a pseudo-triangulation framework (left) which has another embedding (middle) which is non-rigid, with a large number of degrees of freedom. Right: a Hamiltonian pseudo-triangulation with a similar vertical axis of symmetry inducing an embedding which is 1DOF.

A consequence of this example is that the Gröbner bases elimination algorithm may not eliminate all the variables. If such an example could be constructed for a Hamiltonian pseudo-triangulation, then elimination may fail to give a univariate equation in, say, the time t for one alignment system, when searching for the time of the next event t_f in the global problem.

The "bad" examples from Fig. 5 arise from graphs with an axis of symmetry (which in general may affect only some part of the graph), and which have a pseudo-triangular embedding conforming with this symmetry. Are all bad examples of this nature?

OPEN PROBLEM 12. *Characterize the Laman graphs with "bad" (non zero-dimensional) components in their configuration spaces. More precisely, characterize those which have a (Hamiltonian) pseudo-triangulation embedding* and *a bad component.*

In particular, this question contains as a sub-problem the characterization of those planar Laman graphs which have a pseudo-triangular embedding. This has been answered recently in [**13**]: *all* planar Laman graphs have pseudo-triangular embeddings. Hence for a more extensive collection of "bad" examples we may start with planar Laman graphs having a "combinatorial axis of symmetry" on some subgraph (*how to characterize these?*) and see if we can always realize it as a pseudo-triangulation conforming to the symmetry. We do not know whether these are the only situations to consider.

THEOREM 4.2. *There exists a pseudo-triangulation whose configuration space has an exponential number of zero-dimensional components. The statement holds also for Hamiltonian pseudo-triangulations.*

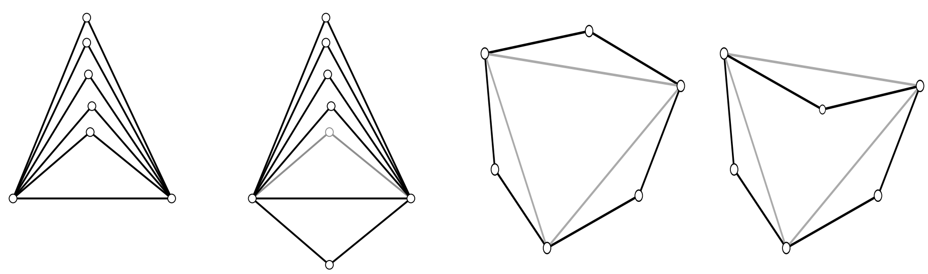

FIGURE 6. Two examples of frameworks with exponentially many embeddings (real solutions). *Flips over* are illustrated. Left: 2^{n-2} embeddings. Right: a Hamiltonian pseudo-triangulation with $n/2$ ears inducing $2^{n/2}$ embeddings, all pseudo-triangulations.

Proof: The proof relies on the examples in Figures 6. For the first example (left two pictures), the bottom two vertices are fixed, the others may be independently *flipped over* the bottom edge to obtain another pseudo-triangular embedding. This yields 2^{n-2} possible embeddings. Note that although each of them is a pseudo-triangulation, they have distinct combinatorial types. For the second example (right two pictures) we have a convex polygon with an even number n of vertices and $n/2$ ears which can be flipped over, yielding $2^{n/2}$ embeddings. □

Another annoying problem may occur when several vertices align simultaneously. In this case, the embedding of the Laman graph for the global problem may have a space of infinitesimal motions of higher dimension. Let's see first how bad this can be.

Theorem 4.3. *There exists a 1DOF expansive mechanism obtained from a pseudo-triangulation for which a linear number of vertices will align simultaneously.*

Proof: Based on the obvious generalization to $n = 3k$ vertices and $n/3$ alignments of the example in Fig. 7 (done for $k = 5$). The picture shows the mechanism at the moment when the bottom two edges align. Simultaneously, all the $k - 1 = n/3 - 1$ vertices above it also align. Of course, for this to happen the edge lengths must be very special. As an aside, notice that the underlying graph is a pseudo-triangulation with a Henneberg I construction (for which the heuristics of section 6 can be applied). But it is easy to turn this into a general Henneberg example by flipping edges in the triangles adjacent to the degree 2 vertices. □

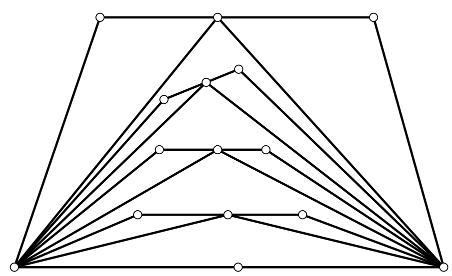

Figure 7. Example of a pseudo-triangulation mechanism with a linear number of vertices aligning simultaneously (shown at the moment when the alignments occur.)

A consequence of this example is that solving the alignment systems approximately may lead to numerical inaccuracies which in turn will lead to combinatorially unreliable decisions. If the times of alignment of several vertices coincide, but the numerical approximations return slightly different values, then we won't be able to detect that and might make incorrect decisions regarding which edges to flip to get the next pseudo-triangulation mechanism.

We now define our concept of **generic pseudo-triangulation mechanism**. It must satisfy two conditions:

(1) The underlying Laman graph, with edge constraints given by the $2n - 4$ mechanism bar lengths plus the convex hull edge length at time t, has a *zero-dimensional configuration space*, for any time t, $t_0 \leq t \leq t_f$.
(2) The mechanism has no multiple alignments.

In other words, a pseudo-triangulation mechanism is *generic* if the extreme examples of this section do not apply to it. The definition is not very satisfactory because it refers to a continuum of positions that have to be tested for the verification of the property. At each position the property can be in principle verified algorithmically (although not very efficiently) by computing the dimension of the configuration space.

Open Problem 13. *Characterize in finite terms the* genericity *of pseudo-triangulation mechanisms and devise good algorithms for testing the property.*

Non-generic pseudo-triangulations have very special edge lengths and very special symmetries. Could it be the case that we can avoid them altogether for the original motivating problem (Carpenter's Rule unfolding)?

Open Problem 14. *Given a simple planar polygon, does it always support a generic pseudo-triangulation mechanism? I.e., among the many ways of placing additional bars to obtain a pseudo-triangulation, and of removing convex hull edges, is there one that is guaranteed to yield a generic mechanism?*

More generally, the problem may be formulated for arbitrary mechanisms.

Open Problem 15. *Define and characterize genericity of arbitrary 1DOF mechanisms.*

Notice that Open Problem 12 appears as a subproblem.

4.2. Problems with Homotopy. In this approach we attempt to find only one root of the algebraic system, starting from a solution at the previous step that is, hopefully, close enough to guarantee that Newton's method will converge to the right solution in a reasonable amount of time.

We have experimented with this technique in Mathematica and found that there are severe drawbacks even on relatively small examples. For the Local Problem, Newton's method works well until we get close to the alignment event. In that case, either the method fails to converge within a number of iterations or (even worse) it ends up giving a solution *going beyond the alignment event*. Both behaviors are impediments to a full automation of the algorithm. In connection with Open Problem 1, we venture a candidate for a good predictor. The infinitesimal velocities of the vertices of the mechanism may be computed via linear programming and used to assist with this decision.

Conjecture 16. *The relative speed of motion of the vertices is a possible predictor for bad behavior of the Homotopy method on the Local Problem around the alignment events.*

For the Global problem Homotopy causes more severe problems, because the starting point for the iteration (given by the position of the mechanism at time t_0) may be quite far from the desired solution t_f. Our experiments have shown that:

- We do not have to compute roots of systems associated to all the vertices. In fact, for those vertices which yield no real roots because they belong to some rigid component, the Homotopy method gives very strange "solutions". In Section 5 we show how to systematically avoid looking for solutions when they do not exists, via pre-computations of rigid components.
- It is hard to distinguish between alignment events that are very close to each other. When several are close, Newton's method may not converge or may give wrong solutions.

Open Problem 17. *Look for predictors of Newton's method failure for the Global Problem.*

We conclude with the comment that it might be possible that, instead of looking for the final value t_f *at the beginning of the Simulation of Motion phase* (which may cause the described bad behavior with the Homotopy method applied to the Global Problem), we may try to apply it later on in the simulation. This would work if we guaranteed that we do not accidentally go beyond the event, and if we could predict that we get closer (e.g., if Newton's method would fail in a limited number of iterations on the Local problem). We have not yet experimented with this approach.

5. Elimination-based Approaches to the Local and Global Problems

The common thread in this section is the rigidity of some graphs and subgraphs occurring in the lifetime of our algorithm. We start this section by proving some properties related to the correct formulation of the Local and Global Problems, for generic pseudo-triangulations. Then we discuss a potential approach based on Elimination (as opposed to Homotopy) methods and present heuristics for simplifying the search for the alignment event.

Theorem 5.1. *Let G be a pseudo-triangulation with a convex hull edge e removed. Let k be an arbitrary vertex and let ik and jk be the extreme edges adjacent with this vertex. Then the graph G' obtained by adding the edge ij is a Laman graph if and only if i and j do not belong to the same r-component of the mechanism $G - \{e\}$.*

The theorem is a consequence of the following more general one.

Theorem 5.2. *Let G be a Laman graph with some edge e removed. Then the graph G' obtained by adding another edge ij is a Laman graph if and only if i and j do not belong to the same r-component of the mechanism $G - \{e\}$.*

Proof: r-components satisfy the Laman count, so adding an edge will violate the Laman property on the set of vertices of the component. Conversely, adding an edge whose endpoints lie on two different r-components does not increase the Laman count for any subset of edges which was already saturated to $2k - 3$. But it makes the global count $2n - 3$, hence we get a Laman graph. □

The resulting graph G', although Laman, may not be infinitesimally rigid, because of the alignment of adjacent edges and the addition of an overlapping edge. See Figure 8.

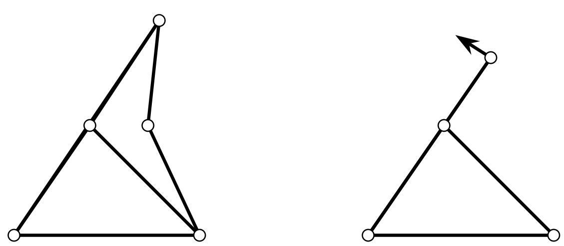

Figure 8. Examples of pseudo-triangulation mechanisms with one aligned vertex. The overlapping added bar between the endpoints of the two aligned vertices can't be seen. Left: infinitesimally rigid. Right: with a non-trivial infinitesimal velocity vector assignment.

However, the following may not be hard to prove:

Conjecture 18. *Let G be a pseudo-triangulation with a convex hull edge e removed. Let k be an arbitrary vertex and let ik and jk be the extreme edges adjacent with this vertex. Let G' be the graph embedding obtained from $G-\{e\}$ by adding the edge ij of length equal to the sum of the lengths of the edges ik and jk and whose combinatorial embedding is the same as it was for G except for the aligned edges ik*

and jk. If k is the only vertex that aligns with this modification, then G' is rigid (but may not be in general infinitesimally rigid).

A stronger conjecture is that the same may be true even when dropping the condition that no other vertices align.

We now discuss Elimination-based approaches to the Local and Global problems. Their correctness relies heavily on the fact that we deal with Laman graphs and on expected positive answers to Conjectures 3 and 9. Their practical applicability may be restricted if elimination would behave exponentially, and one would have to use the special structure of pseudo-triangulations to gain efficiency. We may think of pruning, in a semi-algebraic fashion, solutions to the Laman embedding problem that do not satisfy the semi-algebraic constraints of the desired pseudo-triangulation embedding. On the other hand, we expect better accuracy through such a method. The accuracy is not so much needed for the Simulation of Motion (which is very stable and recovers easily from errors at some previous step). It is essential, however, when deciding the next event.

For reducing the number of alignment systems to be solved when seeking the solution for the Global Problem via Elimination, we can use the following heuristic. We first detect the r-components of the 1DOF mechanism. An alignment will never occur at a vertex whose extreme edges lie in the same r-component (as they never move).

Although not explicitly designed for this purpose, algorithms for detecting r-components, based on matroidal properties of Laman graphs, can be traced in the literature. See [**12**] and the references given there. They are polynomial, but we would like better than this, at least in the case of pseudo-triangulations, where r-components have a nice structure, as they are themselves pseudo-triangulations. In particular, they contain their convex hull edges. We expect that efficient detection of rigid pseudo-triangular components is possible.

Open Problem 19. *Find an* efficient *algorithm to detect r-components of a pseudo-triangulation with an edge removed. Can this be done in linear time (like finding connected and biconnected components of an arbitrary graph)?*

6. Heuristics based on Henneberg Constructions

A consequence of Lemma 4.2 is that it may be inefficient to rely on root finding functions (such as `Solve` in Mathematica) which attempt to find all the solutions of the algebraic systems (3.1)+(3.2) or (3.1)+(3.4). To simplify, in some cases dramatically, the complexity of the system and the accuracy of finding its roots, we will use in this section the particular structure given by the *Henneberg constructions* for pseudo-triangulations. Instead of solving the associated quadratic system of equations and obtain all the solutions simultaneously, we can (sometimes) compute them sequentially. As we move on, we prune those which do not satisfy the order-type semi-algebraic constraints. This is possible when the pseudo-triangulation has a Henneberg I construction.

In this case, though, we will change the parametrization slightly: the variables to be fixed will correspond to the first edge of the Henneberg construction, which is not necessarily the missing edge of the pseudo-triangulation mechanism. But is this really necessary? Lemma 6.1 shows that it is not, for general Henneberg constructions. However, Henneberg I constructions may even be unique, see Figure

9. This example (which is also Hamiltonian) has a unique Henneberg I construction, because at each step there exists a unique vertex of degree 2. (It has many Henneberg II constructions, though.) Hence the starting edge cannot be fixed a priori in such a case.

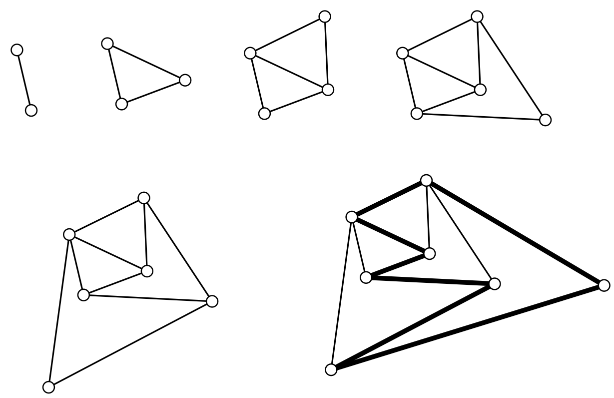

FIGURE 9. Example of a Hamiltonian pseudo-triangulation with a unique Henneberg I construction.

LEMMA 6.1. *For an arbitrary Laman graph, one can always choose any edge to be the first one in a Henneberg construction that never removes it afterwards.*

Proof: This simple argument is due to Walter Whiteley [**25**]. The Laman count guarantees that there are at least 3 vertices of degree at most 3. Therefore, we can fix an edge and, no matter what the degree of its two endpoints, never use it in a Henneberg step, because there will be other eligible vertices. □

To simplify the root finding, follow the Henneberg I construction. Start by fixing the coordinates of the first triangle in the construction and assume (for the sake of the analysis) that we have relabelled the vertices as $1, 2$ and 3. Then at each step i, the coordinates of the previously added vertices (labelled up to $i+2$) would have been computed. Adding the next vertex using the edge lengths of its adjacent edges amounts to intersecting two circles of known centers and given radii. We can solve this system exactly, using only square roots, and retain the solution which verifies the combinatorial pseudo-triangulation constraints. Indeed, it is trivial to show that for Henneberg I graphs, Conjecture 3 holds.

Notice that there is nothing special about pseudo-triangulations in this step. Laman graphs constructible with standard Henneberg I steps (see e.g. [**12**] or [**23**] for Henneberg constructions on Laman graphs) can be solved equally efficiently. We also notice that for given edge lengths, all their embeddings (which may be up to 2^{n-2}, when all the roots are real) have distinct oriented matroids (even partial oriented matroids induced by the rotations of lines through edges around each vertex).

The case when the pseudo-triangulation does not have a Henneberg I construction is more complex. When a type II step is used, an old edge is erased. But this means that in the construction so far we have used an edge length that did not exist. We could do this parametrically. Preprocess the Henneberg construction and mark all the edges that will be eventually removed by adding new variables (parameters) for their unknown lengths. Proceed with the Henneberg construction as long as only type I steps are used and compute parametrically the coordinates of the added points (so far, only square roots should be necessary).

Then when a type II step occurs, eliminate the parameter corresponding to the removed edge by solving a system of three quadratic equations (for the three circles that must cross at the added vertex).

The problem with this approach is that we may have to carry with us, all the way to the end, *all* the parametric solutions, as there is no way of knowing which one will match the combinatorial information of the desired embedding before all the parameters have been eliminated. This part needs further investigation.

Open Problem 20. *Investigate the feasibility of the previous approach based on parametric computation of roots for Henneberg II graphs.*

Finally, we state several open problems motivated by this heuristic. Depending on the answers, one might be able to improve and analyze formally its efficiency.

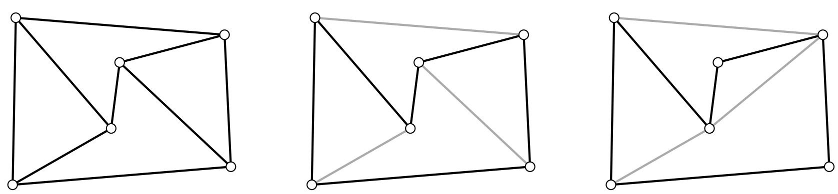

Figure 10. The smallest pseudo-triangulation which is not Henneberg I. The highlighted underlying polygon in the middle has a different, Henneberg I, pseudo-triangulation.

The smallest example of a pseudo-triangulation (of a point set) which is not Henneberg I is shown in Fig. 10 (left). Since it is also a Hamiltonian pseudo-triangulation (middle), this example gives the smallest pseudo-triangulation of a polygon with no Henneberg I construction. But this polygon has another pseudo-triangulation (right) which is Henneberg I. A natural question to ask is: *Does every polygon have a Henneberg I pseudo-triangulation?* That this is not true is shown by the example in Fig. 11 (left) and the following argument. [4] The proof relies on the fact that all the possible pseudo-triangulations have no vertex of degree 2. Indeed, every pseudo-triangulation of the polygon must include the three convex hull edge (middle): this already makes the vertices $1, 2, 4, 5, 7, 8$ be of degree 3. Moreover, not two of the three remaining edges adjacent to an interior vertex may not be simultaneously in the pseudo-triangulation, because of pointedness. Hence there must be an edge adjacent to each of them, making them all of degree at least 3.

This ruins the hope that there might be some simple general way (i.e. via Henneberg I heuristics) around the algebraic problems induced by the Local and the Global Problems for the Combinatorial Roadmap Algorithm.

A related problem is that a pseudo-triangulation may have several Henneberg constructions. Here's a general procedure to produce one. A simple degree count shows that every Laman graph must have either a vertex of degree 2 or of degree

[4]We are grateful to the anonimous referee for this example.

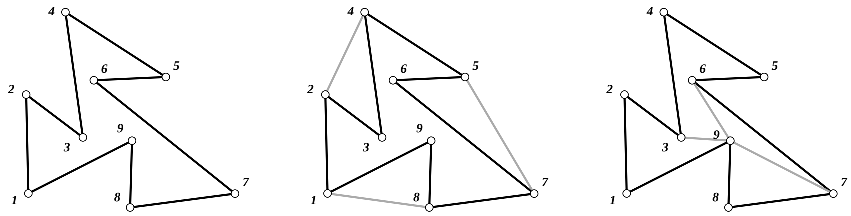

FIGURE 11. A polygon with no Henneberg I pseudo-triangulation.

3. Start working backwards, plucking off degree 2 and 3 vertices. The problem is that there might be several vertices which can be chosen at one step. Are there pseudo-triangulations with exponentially many Henneberg I constructions? Indeed, the example in Fig. 6 has this property: any of the degree 2 vertices can be chosen, yielding $(n-2)!$ orders in which they can be plucked off. But this is a happy situation, since any order yields a Henneberg I construction. Call it a *winning degree* 2 *construction*.

Are there pseudo-triangulatios with a winning degree 2 constuction, but having an option along the way so that when another degree 2 vertex is chosen, one can't continue without using a degree 3 vertex? The answer is NO, because vertices of degree 2 can be removed independently. The two edges adjacent to a degree 2 vertex are not shared with another degree 2 vertex, or else the graph won't be rigid if it has at least 4 vertices. Therefore, if there exists a Henneberg I construction, then any sequence of choices of degree 2 vertices produces one.

Another natural question, *Is there a Hamiltonian pseudo-triangulation with exponentially many Henneberg I sequences?* is answered via the same example from Figure 6 (right). Indeed, the $n/2$ ears can be plucked off independently, leading to $(n/2)!$ Henneberg I sequences.[5]

OPEN PROBLEM 21. *Characterize pseudo-triangulations which have Henneberg I constructions (other than "those for which the plucking-off-degree-2-vertices algorthms works").*

The recent result that planar Laman graphs have pseudo-triangular embeddings [**13**] reduces the problem to: *Characterize planar Laman graphs which have Henneberg I constructions.*

Assuming that we have a polygon and that we can find a Henneberg I pseudo-triangulation for it, we may want to use this one, not the shortest-path-tree approach from [**20**]. Indeed, a shortest-path pseudo-triangulation may not be Henneberg I. The simple example in Fig. 12 shows a hexagon, which has only two possible pseudo-triangulations, both of which are shortest-path pseudo-triangulations, but one is Henneberg I while the other is not.

OPEN PROBLEM 22. *Analyze the number of steps (events) of the pseudo-triangulation roadmap algorithm in other cases than the shortest path tree. In particular, one may want to see how the global number of steps (alignment events) is affected if (say) only Henneberg I pseudo-triangulations are used all the way (e.g. when this is possible) : could we guarantee that the algorithm ends after a finite number of events?*

[5]This example was found by the anonimous referee.

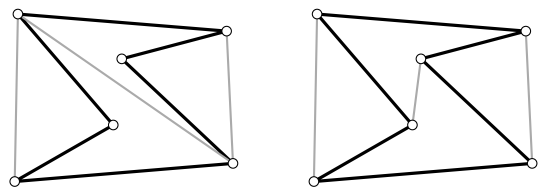

FIGURE 12. Polygon and its two possible pseudo-triangulations, both of which are shortest-path. One is Henneberg I (left), the other is not (right).

7. Further Issues and Open Problems

Our work was motivated by the practical goal of finding an efficient fully-automated implementation of the Pseudo-triangulation Roadmap Algorithm. We have experimented with existing algebraic systems such as Mathematica and found that relying on the built-in functions alone was not practical for the two problems discussed in this paper. But applying the Henneberg I heuristic was very efficient, and we could easily solve large systems. Some further issues related to this approach are included in this final section.

The parameter δt used in the simulation of *one* mechanism depends on the choice of a *constant* number of time steps. Ideally, all these choices should be synchronized and their calculation automated.

OPEN PROBLEM 23. *Develop criteria for computing the time steps δt of all the mechanisms occurring during the lifetime of an unfolding process, so that the motion looks locally smooth and at the same time maintains global esthetic qualities achieved, for instance, through a uniform global average speed.*

This calculation may happen, for instance, after all the alignment events have been computed and the *combinatorial* structure of the whole unfolding process has been determined. To achieve a global uniform speed of *one vertex*, one has to compute the lengths of all the trajectory pieces for that vertex. For a global effect of smoothness, one may have to average over all vertices.

One of the most basic questions induced by the Local Problem to which we briefly referred earlier is:

Characterize and recognize efficiently those graphs which have an embedding as a pseudo-triangulation.

An understanding of the specifics of these graphs is hoped to give insights into possible ways of solving the algebraic systems for the Local and Global Problems more efficiently. Since pseudo-triangulations are planar Laman graphs, a natural question is whether the reverse is also true. This has been answered recently, see [**13**]: *pseudo-triangulation graphs coincide with planar Laman graphs.* This still leaves open the question, naturally related to Open Problem 19.

OPEN PROBLEM 24. *Develop efficient algorithms for recognizing planar Laman graphs. Can this be done in linear time?*

A number of remaining open questions concern Hamiltonian pseudo-triangulations:

OPEN PROBLEM 25. *Characterize and recognize efficiently the Hamiltonian pseudo-triangulations.*

Open Problem 26. *Is there a Henneberg construction for Hamiltonian pseudo-triangulations (i.e. one that would maintain Hamiltonicity throughout the construction)?*

Further examples and results of our current and future experiments may be accessed through the author's web page, http://cs.smith.edu/ streinu/research.html.

Acknowledgements. The presentation benefited from the comments of a careful anonimous referee, who also answered some of the questions posed in the first version of the paper and suggested the terminology *Hamiltonian pseudo-triangulations*. I acknowledge useful discussions on computational algebraic geometry with Günter Rote, Bernd Sturmfels, Laureano Gonzalez-Vega and Frank Sottile. Eleanor Farrington implemented a number of Mathematica experiments during a Summer 2001 REU at Smith College, sponsored by the author's NSF RUI-grant CCR-0105507. Some ideas reported in this paper emerged during the Barbados Workshops on *Pseudo-triangulations*, January 2001 and on *Rigidity Theory and Scene Analysis* (organized by the author with partial NSF support under grants CCR-0104370 and CCR-0203224), the DIMACS Workshop on Algorithmic and Quantitative Aspects of Real Algebraic Geometry in Mathematics and Computer Science (organized by Richard Pollack, Marie-Françoise Roy and Micha Sharir) at DIMACS/Rutgers University in March 2001, and several Combinatorists of New England (CONE) one-day meetings held at Smith College (organized by Michael Albertson, Karen Collins and Ruth Haas). All of these enjoyed partial support from NSF.

References

[1] M. Aigner and G. Ziegler. *Proofs from THE BOOK*, Springer, 2nd edition, 2002.

[2] A. Björner, M. Las Vergnas, B. Sturmfels, N. White and G. Ziegler, *Oriented Matroids*, Cambridge University Press, 1993.

[3] S. Basu, R. Pollack and M.-F. Roy. *Computing Roadmaps of Semi-algebraic Sets on a Variety*, JAMS, 13, pp. 55-82, 2000. prelim. version in Proc. 28th ACM Symp. Theory of Computing, pp. 168-173, 1996.

[4] S. Basu, R. Pollack and M.-F. Roy. *Algorithms in Real Algebraic Geometry*, book, in preparation.

[5] C. Borcea and I. Streinu. *On the Number of Embeddings of Minimally Rigid Graphs*, Proc. ACM Symp. Comput. Geometry (SoCG), Barcelona, pp. xxx, 2002.

[6] J. Canny. *Complexity of Robot Motion Planning*, MIT Press, Cambridge, MA, 1988.

[7] A. Cauchy. *Sur les polygones et les polyédres, seconde mémoire*, J. École Polytechnique XVIe Cahier, Tome IX (1813), 87; Oevres Complétes, IIe Série, Vol. I, paris, 1905, 26-38.

[8] R. Connelly, E. Demaine and G. Rote. *Straightening Polygonal Arcs and Convexifying Polygonal Cycles*, Proc. 41st Symp. on Foundations of Computer Science (FOCS), 2000, pp. 432-442.

[9] D. Cox, J. Little and D. O'Shea. *Ideals, Varieties and Algorithms. An Introduction to Computational Algebraic Geometry and Commutative Algebra.* Springer Verlag, 1992.

[10] D. Cox, J. Little and D. O'Shea. *Using Algebraic Geometry.* Springer Verlag, 1998.

[11] J.E. Goodman and R. Pollack, *Semispaces of configurations, cell complexes of arrangements*, J. of Combinatorial Theory, Series A, 37 (1984) 257–293.

[12] J. Graver, B. Servatius and H. Servatius. *Combinatorial Rigidity*, Amer. Math. Soc., Graduate Studies in Mathematics vol.2, 1993.

[13] R. Haas, F. Santos, B. Servatius, D. Souvaine, I. Streinu and W. Whiteley. *Planar Minimally Rigid Graphs have Pseudo-Triangular Embeddings*, manuscript, 2002.

[14] L. Henneberg, *Die graphische Statik der starren Systeme*, Leipzig,1911.

[15] M. Kapovich and J. Millson. *On the moduli spaces of polygons in the Euclidian plane*, Journ. of Diff. Geometry, vol. 42, N 1, 1995, pp. 133-164.

[16] G. Laman, *On graphs and rigidity of plane skeletal structures.* Eng. Math., 4, 331-340, 1970.

[17] M. Pocchiola and G. Vegter *Topologically Sweeping Visibility Complexes via Pseudo-Triangulations*, Discrete Comput. Geometry 16(4), 419-453, 1996.

[18] A. Recski. *Matroid Theory and Its Applications*, Springer Verlag, 1989.

[19] G. Rote, F. Santos and I. Streinu, *Expansive Motions and the Polytope of Pointed Pseudo-Triangulations*, to appear in *J.E. Goodman and R. Pollack Festschrift* volume, (B. Aronov, J. Pach, eds.), 2002.

[20] I. Streinu. *A Combinatorial Approach to Planar Non-colliding Robot Arm Motion Planning*, Proc. 41st Symp. on Foundations of Computer Science (FOCS), 2000, pp. 443-453.

[21] T.S. Tay and W. Whiteley. *Generating Isostatic Frameworks*, Structural Topology 11, pp. 20-69, 1985.

[22] J. Verschelde, P. Verlinden and R. Cools. *Homotopies Exploiting Newton Polytopes for Solving Sparse Polynomial Systems*, SIAM J. Numer. Anal. 31, pp. 915-930, 1994.

[23] W. Whiteley. *Some Matroids from Discrete Applied Geometry*, Contemporary Mathematics, vol. 197, pp. 171-311, Amer. Math. Soc. 1996.

[24] W. Whiteley. *Rigidity and Scene Analysis*, in *Handbook of Discrete and Computational Geometry*, (J.E. Goodman, J. O'Rourke, eds.), 1997, pp. 893-916.

[25] W. Whiteley. *Personal communication, 2002.*

Department of Computer Science, Smith College, Northampton, MA 01063, USA.

Current address: Department of Computer Science, Smith College, Northampton, MA 01063, USA.

E-mail address: streinu@cs.smith.edu