Publ. Mat. **50** (2006), 279–299

FROM GRAPHS TO TENSEGRITY STRUCTURES: GEOMETRIC AND SYMBOLIC APPROACHES

Miguel de Guzmán† and David Orden*

Abstract

A form-finding problem for tensegrity structures is studied; given an abstract graph, we show an algorithm to provide a necessary condition for it to be the underlying graph of a tensegrity in $\mathbb{R}^d$ (typically $d = 2, 3$) with vertices in general position. Furthermore, for a certain class of graphs our algorithm allows to obtain necessary and sufficient conditions on the relative position of the vertices in order to underlie a tensegrity, for what we propose both a geometric and a symbolic approach.

1. Introduction

In this paper we study an instance of the so-called *form-finding problems* for tensegrity structures. These have brought special attention both among mathematicians and engineers since the seminal works of Kenneth Snelson around 1948 (see [**13**]). Roughly speaking, a form-finding problem for a tensegrity structure asks to determine a geometric configuration of points and straight edges in $\mathbb{R}^d$ (typically $d = 2, 3$) such that the whole structure is in a self-tensional equilibrium. The word *tensegrity* was coined from *tension* and *integrity* by Buckminster Fuller, deeply impressed by Snelson's work.

Apart from a purely mathematical interest [**3**], [**11**], understanding these structures has applications to architecture and structural engineering [**14**] and has led to interesting models for viruses and cellular structures [**1**], [**7**]. It is also considered a useful tool for the study of deployable structures [**9**], [**12**], [**15**]. Previous works have proposed a

2000 *Mathematics Subject Classification.* 05C85.
Key words. Form-finding problems, tensegrity, graphs, polynomial elimination.
†In memoriam.
*Research partially supported by grants MEC MTM2005-08618-C02-02 and CAM S-0505/DPI/000235.

number of different approaches to solve form-finding problems, which can be found in the recent review [**16**].

In particular, the present paper deals with the form-finding problem of building tensegrity structures with a given underlying graph G, in a given $\mathbb{R}^d$. The graph has to be understood as an *abstract graph*, i.e., a set of vertices and pairs of vertices (edges). We aim to solve the following two problems:

- First, to decide whether G can be the underlying graph of a tensegrity structure in $\mathbb{R}^d$.
- In case such a tensegrity with underlying graph G is possible, to characterize the relative position of its vertices.

In order to solve these problems, we first look for decompositions of tensegrities into basic instances, called *atoms*. This motivates a combinatorial method that allows to decompose a graph G into the smallest graphs that can underlie a tensegrity. In order to build up a tensegrity with graph G, we propose to reverse its decomposition: We show that this solves the above problems for a certain class of graphs and we present two different approaches. The first one looks at the geometric structure of the tensegrity; it is quite visual and provides intuition of the intrinsic properties of tensegrity structures. However, it becomes difficult to use for complicated structures. The second approach condenses in a matrix the information about the tensegrity; this allows to use tools from Symbolic Computation, despite being less intuitive.

The paper is organized as follows: The basic notions and results are introduced in Section 2. Then, Section 3 introduces a method to decompose a tensegrity into atoms, which motivates a decomposition of the abstract graph G, reversed then by geometric means. Finally, in Section 4 a *rigidity matrix* is used for a symbolic resolution.

2. Preliminaries

In this section we introduce the basic notions and results used in the paper. Despite it aims to be self-contained, an interested reader can look at [**18**] for further examples and a more detailed overview of the mathematical concepts. Let us introduce first the rigorous definition of "self-tensional equilibrium":

Definition 2.1. Let $G = (V, E)$ be an abstract graph:

- A *framework* $G(P)$ in $\mathbb{R}^d$ is an embedding of G on a finite point configuration $P := \{p_1, \ldots, p_n\}$ in $\mathbb{R}^d$, with straight edges. In the

sequel we will focus on *general position* point configurations (no $d+1$ points lie on the same hyperplane).

- A *stress* w on a framework is an assignment of scalars w_{ij} (called *tensions*) to its edges. Observe that $w_{ij} = w_{ji}$, since they refer to the same edge.
- Such a w is called a *self-stress* if, in addition, the following equilibrium condition is fulfilled at every vertex:

$$\forall\, i, \quad \sum_{ij \text{ edge}} w_{ij}(p_i - p_j) = 0. \tag{1}$$

That is, for each vertex p_i the scaled sum of incident vectors $\overrightarrow{p_i p_j}$ is zero.

Observe that the null stress is always a self-stress, of no interest for us. Note also that all scalar multiples of a self-stress (in particular its opposite) are self-stresses as well. We will see later that, indeed, the space of self-stresses on a given graph is a vector space.

Lemma 2.2. *Let $p \in P$ be a vertex of a d-dimensional framework $G(P)$ such that P is in general position. Given a non-null self-stress on $G(P)$, either at least $d+1$ of the edges incident to p receive non-null tension, or all of them have null tension.*

Proof: The result is true for any d, but the reader may consider $d = 2, 3$ here. Let k be the number of edges incident to p that have non-null tension. The equilibrium condition on p implies having k vectors in $\mathbb{R}^d$, with common tail, which add up to the zero vector. For $k < d+1$, this is only possible if their $k+1$ endpoints do not span a k-space. But either $k = 0$ or this contradicts the general position assumption. □

As a consequence, the next property makes particularly interesting the study of a certain family of general position frameworks, the so-called $(d+1)$-*regular* ones, for which a null tension on a single edge propagates to the rest of them:

Corollary 2.3. *Given a d-dimensional framework all of whose points are in general position and have exactly $d+1$ incident edges, a self-stress is non-null if, and only if, it is non-null on every edge.*

The following definition introduces our final object of study, which is a physical model of the mathematical objects defined above:

Definition 2.4. We define a *tensegrity structure* $T(P)$ to be a self-stressed framework in which:

- Edges ij such that $w_{ij} > 0$ have been replaced by inextensible *cables* (its endpoints constrained not to get further apart),
- Edges with $w_{ij} < 0$ have been replaced by unshrinkable *struts* (endpoints constrained not to get closer together), and
- Edges with $w_{ij} = 0$ have been removed.

If no confusion is possible, a tensegrity structure $T(P)$ will be denoted by just T. For another physical interpretation, one can think of cables and struts as springs endowed with a certain tension, respectively inwards and outwards. That is; cables and struts incident to point p_i have respectively tensions in the direction of $+\overrightarrow{p_ip_j}$ (inwards) and $-\overrightarrow{p_ip_j}$ (outwards), see Figure 1.

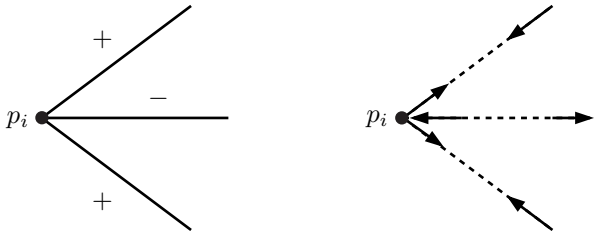

FIGURE 1. Left: two cables (+) and a strut (−) incident to p_i. Right: Their representation as springs with inwards and outwards tensions.

Observe that, given a tensegrity structure, it might be possible to replace the struts by bars which react to the surrounding tensions. For example, if we replace the strut in Figure 1 by a bar, this will receive an outwards tension at p_i, as a reaction to the sum of cable tensions. Such a replacement is usual when constructing tensegrity sculptures, like those in [**13**].

The most emblematic tensegrity structure is shown in Figure 2. Named *oblique triangular prism* with rotational symmetry, it is composed of nine cables, six of which form two copies of an equilateral triangle, the top one rotated 30 degrees, joined by three struts alternating the rest of cables. Thick edges denote the struts, which could be replaced by bars as before.

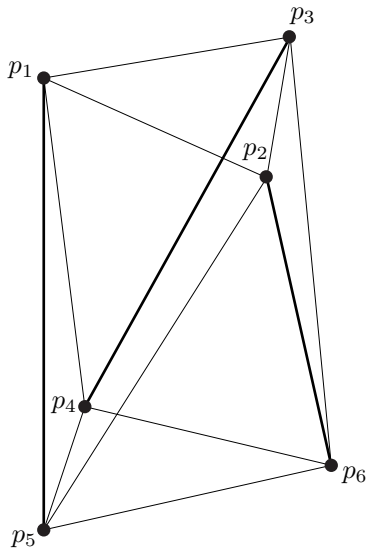

FIGURE 2. The oblique triangular prism.

The last definition in this section introduces the smallest tensegrities possible, which we will show in Section 3 to be the fundamental bricks for building up any tensegrity:

Definition 2.5. We define a *self-stressed atom* in $\mathbb{R}^d$ ($d = 2, 3$) to be a general position realization of the complete graph K_{d+2}, together with its unique (up to constant multiplication) non-null self-stress. The *tensegrity atoms* are then obtained replacing edges by cables and struts. When no confusion is possible, we will just refer to *atoms*.

Figure 3 shows half of the possible tensegrity atoms in $\mathbb{R}^d$ for $d = 2, 3$, where thick edges denote struts. The other half is obtained by interchanging cables and struts or, equivalently, by considering the opposite tensions. It is not difficult to check, using Lemma 2.2, that configurations with fewer points or edges do not admit self-stresses apart from the null one. The non-trivial fact that the above frameworks do admit a unique (up to constants) non-null self-stress appears in [**10**], where existence is proved by the following result.

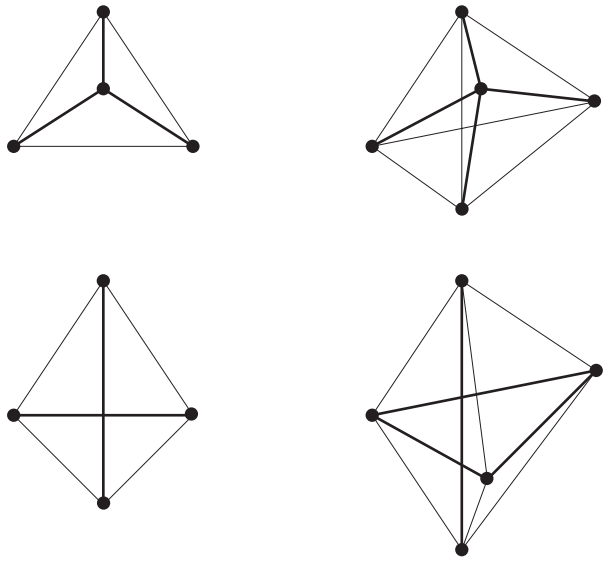

FIGURE 3. Types of tensegrity atoms in $\mathbb{R}^2$ (left) and $\mathbb{R}^3$ (right).

Proposition 2.6. *Let $\sum_{i=1}^n \lambda_i p_i = 0$, $\sum_{i=1}^n \lambda_i = 0$ be an affine dependence on a point set $P = \{p_1, \ldots, p_n\}$. Then, $w_{ij} := \lambda_i \lambda_j$ defines a self-stress on the complete graph $K(P)$.*

Proof: For any $p_i \in P$, we have:

$$\sum_{ij \in K} w_{ij}(p_i - p_j) = \sum_{j=1}^{n} \lambda_i \lambda_j (p_i - p_j) = \lambda_i p_i \sum_{j=1}^{n} \lambda_j - \lambda_i \sum_{j=1}^{n} \lambda_j p_j,$$

which equals zero. □

In order to prove the uniqueness up to constants, consider two different self-stresses, one not a scalar multiple of the other. Then some linear combination of them would cancel the tension at a particular edge but not at all of them, in contradiction with Corollary 2.3. Note that we are using the claimed fact that self-stresses form a vector space, as will be shown at the beginning of the next subsection.

3. Geometric approach

In this section we present a geometric algorithm to decompose a tensegrity into atoms. This decomposition motivates a combinatorial one, which opens the way towards the resolution of the two problems posed in Section 1.

3.1. Decomposing into atoms.

Let $G(P)$ and $G'(P')$ be two self-stressed frameworks such that $P \cup P'$ is a point configuration in general position. Let w and w' be their self-stresses. For the framework $G(P) \cup G'(P')$ obtained by union of vertices and edges, one can define the sum of self-stresses $w + w'$ in the natural way: Assign tension $w_{ij} + w'_{ij}$ to common edges ij and maintain the initial tension at the others.

It is easy to observe that equations (1) are fulfilled and hence $w+w'$ is indeed a self-stress. Furthermore, the space of self-stresses on a given graph $G = (V, E)$ together with this sum and the product by a scalar form a vector subspace of $\mathbb{R}^E$, when the latter is identified with the space of all self-stresses.

Abusing notation, we denote by $G + G'$ the self-stressed framework obtained. Observe that, after this addition is performed, one can appropriately replace edges by cables and struts in order to obtain a tensegrity structure $T+T'$. Hence, the sum of tensegrities yields another tensegrity.

Observation 3.1. *We will only consider this kind of addition when P and P' have at least d points in common; otherwise we obtain either two separate tensegrity structures or one of them hanging from the other.*

The main result in this section states that, reciprocally, under our conditions every tensegrity can be decomposed into a sum of tensegrity atoms:

Theorem 3.2 (Atomic decomposition of tensegrities)**.** *Every non-null tensegrity structure $T(P)$, P in general position, is a finite sum of tensegrity atoms. This decomposition is not unique in general.*

Proof: Let $G(P)$ and w be the framework and non-null self-stress associated to $T(P)$. We show how to obtain, by addition of atoms, a chain of non-null self-stresses w' on $G(P)$ in which the number of vertices with only null incident tensions (*null* vertices) is increased at each step. At the end we come up with a self-stressed framework with only null vertices, so that the original tensegrity T will be the sum of the opposites of those atoms that have appeared in the process.

Let us focus on the two-dimensional case, since the d-dimensional one is carried out analogously: At each step, an arbitrary non-null vertex $a \in P$ is chosen to be converted in a null one. By Lemma 2.2, only the following two cases are possible (see Figures 4 and 5):

- **Type 1:** If exactly three incident edges ab, ac, ad have non-null tension, we consider the atom K of vertices a, b, c, d. Since this atom has a non-null self-stress w^K which is unique up to constants, we can choose w^K_{ab} to be the opposite of the tension assigned to edge ab at the current stress w', i.e. $w^K_{ab} := -w'_{ab}$. Because of the equilibrium at a, it turns out that also $w^K_{ac} = -w'_{ac}$ and $w^K_{ad} = -w'_{ad}$. Therefore, adding w^K to the current self-stress makes vertex a have only null tensions at incident edges (i.e. makes it disappear from the induced tensegrity). See Figure 4, where dashed interior edges in the second picture are opposite to those in the first one. Note that at b, c, d the edges bc, bd and cd may have appeared with non-null tension, but these extra edges do not affect a. However, we will be concerned about them later.

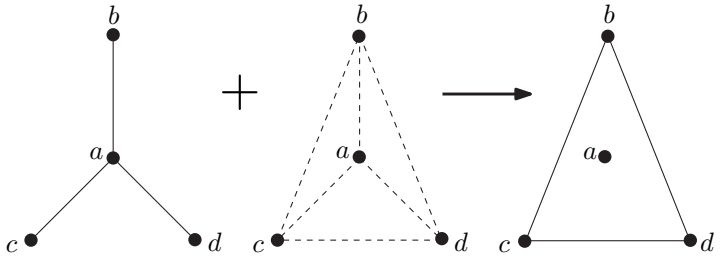

Figure 4. Type 1 step, exactly three incident edges with non-null tension.

- **Type 2:** If $a \in P$ has incidence degree greater than 3, let b, c, d be neighbors of a. Consider the atom $\bar{K}$ of vertices a, b, c, d (and all the possible edges between them) and choose it to have tension $w^{\bar{K}}_{ab} := -w'_{ab}$ at edge ab. Hence, obviously $w' + w^{\bar{K}}$ has null tension at edge ab. Again, other edges bc, bd, cd may appear with non-null tension, but not incident to a. Hence, repeating this process if needed, we obtain a self-stress on $G(P)$ in which a has only three incident edges with non-null tension (i.e. in the induced tensegrity, a has only three incident edges). Now we are in the previous case. See Figure 5, where now only dashed edge ab is guaranteed to be opposite to its filled counterpart.

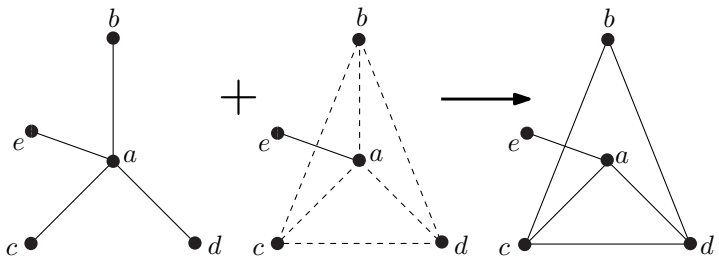

FIGURE 5. Type 2 step, more than three incident edges with non-null tension.

Since a sum of self-stresses is another self-stress, after a finite number of these steps we get a self-stress with at least one more null vertex, for which we can iterate the process until all vertices become null. Note that different choices of vertices to make null may lead to different decompositions. □

Remark 3.3. The reader should notice that the addition of an atom only changes the value of the self-stress on edges of the atom. Hence, adding an atom might cancel other tensions than the intended ones, but only at edges contained in the atom.

Motivated by the *geometric* process in Theorem 3.2, we define now the following *combinatorial* algorithm, that can be applied to any abstract graph G:

Algorithm 3.4 (Combinatorial decomposition)**.**

INPUT*:* *abstract graph $G = (V, E)$ and dimension d.*

OUTPUT*:* *list L of "atoms", where each atom is a subset of $(d+2)$ elements of V.*

1. *Initialize $L = \emptyset$.*
2. *While E is not empty, choose a vertex $a \in V$ with minimum degree and:*
   - 2.1 *If a has degree $\leq d$, remove its incident edges from E.*
   - 2.2 *If a has degree $d+1$, let $a_0, \ldots, a_d$ be its neighbors. Remove the edges aa_i from E. Add to E all the edges a_ia_j that were not in E. Insert the atom $\{a, a_0, \ldots, a_d\}$ to the list L.*

2.3 *If a has degree at least $d+2$ do the following until it has degree $d+1$, then go to* 2.2*: Choose $d+1$ neighbors $a_0, \ldots, a_d$ of a. Remove the edge aa_0 from E and insert to E all the edges $a_i a_j$ that were not in E. Insert the atom $\{a, a_0, \ldots, a_d\}$ to the list L.*

3. *Return L.*

See Figures 6 and 7 for examples. This combinatorial algorithm is the tool for the first result towards the resolution of the problems posed in Section 1:

Theorem 3.5. *Given an abstract graph G, in order for it to underlie a tensegrity $T(P)$ in $\mathbb{R}^d$ it is a necessary condition that, chosen a combinatorial decomposition, for every edge pq of G there is an atom containing both endpoints p and q. (In other words, that step 2.1 of the algorithm removes only edges that were contained in the complete graph defined by some atom of the combinatorial decomposition.)*

Proof: Choose a combinatorial decomposition as above. If such a tensegrity $T(P)$ exists, then the combinatorial decomposition induces a geometric one: On the one hand, the geometric counterpart of step 2.1 shows up naturally because of Lemma 2.2. On the other hand, steps 2.2 and 2.3 correspond to the geometric steps of types 1 and 2, respectively, in the proof of Theorem 3.2, with tensions determined by the edge(s) to be deleted. The necessary condition in the statement is then a consequence of Remark 3.3. □

Figure 6 shows a combinatorial decomposition that fulfills the condition of Theorem 3.5, while Figure 7 shows one for which edge pq shows that the condition is not fulfilled, hence the graph cannot underly a tensegrity.

Therefore, when asked about the existence of a tensegrity in $\mathbb{R}^d$ with underlying graph G, we first get a combinatorial decomposition and then check the condition of Theorem 3.5. If this is not fulfilled, then the answer is negative. However, in the next subsection we show that if the graph G admits a combinatorial decomposition of a certain type, then this decomposition can be geometrically reversed, leading in a straight way to a complete characterization of the point sets P on which $G(P)$ admits a tensegrity.

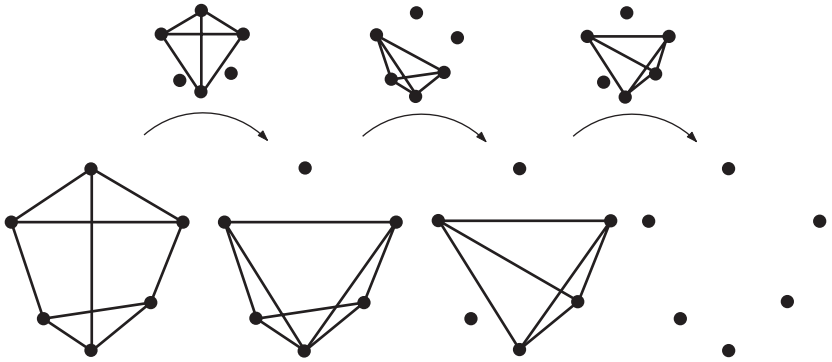

Figure 6. Combinatorial decomposition fulfilling the condition of Theorem 3.5.

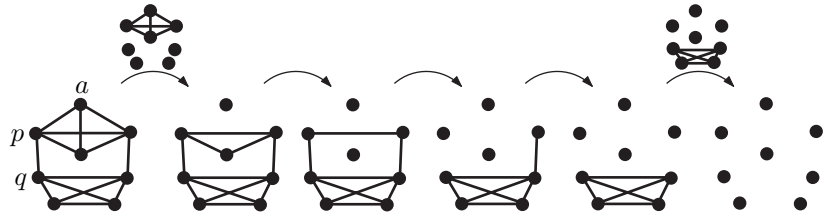

Figure 7. Combinatorial decomposition not fulfilling the condition of Theorem 3.5.

3.2. Geometrically solving form-finding problems.

Let us start defining a class of combinatorial decompositions, for which the reader can find an example in Figure 6.

Definition 3.6. We call *edge-inserting* combinatorial decompositions those in which all extracted atoms but the last one introduce at least one edge to the intermediate graph.

The following result states that, for this class of combinatorial decompositions, their geometric reversal provides a solution to the problems posed in Section 1. Note the request of a self-stress non-null on every edge, in order for G to underlie the tensegrity:

Theorem 3.7. *If an abstract graph G admits an edge-inserting combinatorial decomposition, then the reconstruction of the graph from the atomic decomposition produces a set of equations and negated equations characterizing the point sets P in general position that make $G(P)$ admit a self-stress non-null on every edge (and hence the ones that admit a tensegrity $T(P)$).*

Proof: In order to determine which choices of coordinates underlie tensegrities, we have to consider as variables the coordinates of the new points p_i added in the reconstruction (the first $d+2$ points can be arbitrarily chosen, since tensegrities are projectively invariant [**11**]). Thus, the tensions of the edges in each atom introduced are functions on these variables.

Furthermore, recall from Definition 2.5 that the self-stress of each atom has one degree of freedom. Hence, for the reconstruction of a general combinatorial decomposition we have to consider one extra variable α_j for each atom (except for the first one, whose stress can be considered a normalization constant).

However, these extra variables are not needed for edge-inserting combinatorial decompositions: For each step of the geometric reconstruction, at least one edge was inserted by the combinatorial decomposition algorithm and has to be removed in this precise reconstruction step. Then, the relative self-stress given to this atom (considered as a function on the positions) can be determined at the insertion step: It is exactly the one that cancels the tension(s) at the edge(s) to be removed.

Therefore, the reconstruction in this case leads to a system of equations $f(p_1,\ldots,p_n)=0$ and negated equations $f(p_1,\ldots,p_n)\neq 0$, one equation for each edge not in G that appears in an intermediate step of the process (expressing that this edge has tension zero and does not appear in the tensegrity) and one negated equation for each edge in G (expressing that it does appear in the tensegrity). □

Let us point up that two decompositions of G lead to equivalent collections of conditions, since the tensegrities constructed using one decomposition can always be decomposed and reconstructed using the other. Hence, they have to fulfill both sets of necessary and sufficient conditions, and therefore these have to be equivalent. In order to finish the section, we illustrate the decomposition and reconstruction process with the following example:

Example 3.8. The graph $G=(\{1,\ldots,6\},\{12,14,16,23,26,34,35,45,56\})$ underlies a tensegrity $T(P)$ in general position in $\mathbb{R}^2$ if, and only if,

the triangles $p_1p_2p_6$ and $p_3p_4p_5$ are in perspective position, i.e., the lines $\overline{p_2p_3}$, $\overline{p_5p_6}$ and $\overline{p_1p_4}$ are concurrent.

We start with the edge-inserting combinatorial decomposition of G depicted in Figure 6. In Figure 8 we show the reversal of this decomposition, where dashed edges are the inserted ones that have to be removed.

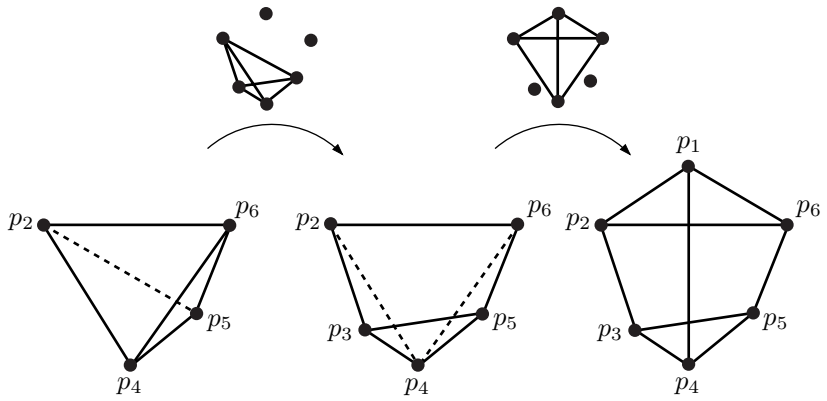

FIGURE 8. Reconstruction of an edge-inserting combinatorial decomposition.

As observed in the proof of Theorem 3.7, the initial points p_2, p_4, p_5, p_6 can be chosen arbitrarily. Then we add the atom p_2, p_3, p_4, p_5, whose self-stress is determined by the condition of canceling the tension at edge p_2p_5. The feasible positions of p_3 are characterized by the equations and negated equations corresponding respectively to the edges omitted and depicted in the middle picture.

Finally, the addition of the atom p_1, p_2, p_4, p_6 has to remove edges p_2p_4 and p_4p_6. This fact allows us to rephrase geometrically the set of equations and negated equations obtained for the graph G: The point p_1 has to be placed on the line spanned by the resultant of the "undesired" tensions w_{24} and w_{46} so that the tension on edge p_1p_4 replaces them. Moving p_1 along the feasible line towards the intersection of $\overline{p_2p_3}$ and $\overline{p_1p_4}$ forces tension w_{26} at point p_2 to be very small. But this equals tension w_{62} at point p_6; therefore, p_1 has to be close to the intersection of $\overline{p_1p_4}$ and $\overline{p_5p_6}$ as well.

Taking this argument to the limit, we conclude that p_1 has to be chosen in such a way that the lines $\overline{p_2p_3}$, $\overline{p_5p_6}$ and $\overline{p_1p_4}$ are concurrent.

4. Symbolic approach

This section is devoted to show that the form-finding problems considered can be reformulated in a matricial form, and how this can be used to obtain the equations and negated equations in Theorem 3.7 by means of symbolic computations. In particular, this provides an alternative to the geometric *ad hoc* reasonings like the one in Example 3.8, and allows to deal with more complicated and less visual examples.

4.1. The rigidity matrix.

Before starting this subsection, let us point up that in this paper we are not dealing with the rigidity of tensegrity structures; we are just concerned by their self-equilibrium. An interested reader can see [**18**] for more information on rigidity of frameworks. However, the following notion from rigidity analysis turns out to be useful for our purposes:

Definition 4.1. Let $G(P)$ be a framework with n vertices and e edges in $\mathbb{R}^d$. Its *rigidity matrix* $R(P)$ has e rows and nd columns, defined as follows:

- There is a row per edge ij of the framework, with $i < j$ and in lexicographic order.
- Each block of d columns is associated to a vertex p_i and contains either the d coordinates $p_i - p_j$, at those rows corresponding to edges ij incident to p_i, or zeros at the rest of rows.

For a *complete framework* on n vertices (with all the possible edges), the rigidity matrix has the following condensed form:

$$\begin{array}{l}\text{vertex } p_1\\ \quad\downarrow\\ \begin{pmatrix} p_1-p_2 & p_2-p_1 & 0 & 0 & \dots & 0 & 0\\ p_1-p_3 & 0 & p_3-p_1 & 0 & \dots & 0 & 0\\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & 0 & 0 & \dots & p_{n-1}-p_n & p_n-p_{n-1}\end{pmatrix}\begin{array}{l}\leftarrow \text{ edge } 12\\ \leftarrow \text{ edge } 13\\ \\ \\ \\ \end{array}.\end{array}$$

The key observation is that the equilibrium equations (1) can be restated in matricial form as

$$w \cdot R(P) = 0 \tag{2}$$

where w is a $1 \times e$ vector of entries w_{ij}, $i < j$ (recall that $w_{ij} = w_{ji}$) and the right-hand side is the $1 \times nd$ zero vector. That is to say, self-stresses w are row dependencies for the rigidity matrix, what leads to the following observation:

Observation 4.2. *For a framework $G(P)$, being the underlying graph of a tensegrity $T(P)$ is equivalent to the existence of a w with no null component in the left kernel of the rigidity matrix $R(P)$ of the framework.*

This observation already suggests a method to characterize the point sets P in general position that make $G(P)$ admit a tensegrity $T(P)$: Computing the kernel of the rigidity matrix $R(P)$ for the framework $G(P)$. For the sake of consistency, we refer the interested reader to [**6**] for an example of the use of this method, and we show here instead how to use the decomposition-reconstruction method together with the matricial expression of equilibrium.

4.2. Symbolically solving form-finding problems.

According to Observation 3.1, the reconstruction of a combinatorial decomposition inserts at most two new points at each step, with indeterminate coordinates $\overline{x}, \overline{y} \in \mathbb{R}^d$. For the framework to maintain equilibrium after their addition, the points have to belong to the projection of the solutions of (2) over the space of the variables x_i, y_j. Furthermore, in order to look for non-null tensions on every edge, the inequations $w_{ij} \neq 0$ for every i, j have to be considered. Equivalently, extra variables t_{ij} and equations $w_{ij}t_{ij} = 1$ can be introduced to the system.

It is well-known (see [**4**, Lemma 1, p. 120]) that the polynomial elimination of variables w_{ij}, t_{ij} from the above system of equations contains the above projection. Therefore, polynomial elimination provides necessary conditions for the position of the vertices, which can be later symbolically tested for sufficiency.

We illustrate this symbolic method with the following example, in which in order to perform polynomial elimination we have chosen to compute the intersection of $\mathbb{R}[\overline{x}, \overline{y}]$ with a *Groebner basis* of the equations, for an elimination order in which variables x_i, y_j are smaller than the w_{ij}, t_{ij}'s; see [**4**, Theorem 2, p. 113]. Although there are more efficient elimination techniques, the reason for this choice is to present the operations performed with the software `Maple` [**17**] which, apart from being broadly known, has a specific `linalg` package for linear algebra. For more efficient elimination software, the reader can consider `CoCoA` [**2**]. Let us also point up that the graph considered has eight triangles and, in fact, is the same as the graph of an octahedron.

Example 4.3. The graph $G = (\{1, \ldots, 6\}, \{12, 13, 14, 15, 23, 25, 26, 34, 36, 45, 46, 56\})$ of an oblique triangular prism underlies a tensegrity if, and only if, the six vertices lie on a ruled hyperboloid that contains the

edges of one of the three cycles of length 4 in G. Equivalently, if and only if the planes containing four alternating triangles intersect.

We start with the combinatorial decomposition of G in which the vertex 6 is chosen at step 2 of Algorithm 3.4. This gives the atoms $2,3,4,5,6$ and $1,2,3,4,5$ (see Figure 2). In the sequel we detail the steps of the reconstruction process using `Maple 7`, and omitting some outputs of no interest:

• According to the proof of Theorem 3.7, the first $d+2$ points $p_1,\ldots,p_5$ can be arbitrarily chosen:

```
> p1:=[0,0,0]:  p2:=[1,1,1]:  p3:=[0,1,0]:  p4:=[1,0,0]:
  p5:=[0,0,1]:
```

• Solving the corresponding equation (2) we get the tensions of the atom p_1,p_2,p_3,p_4,p_5. In order to generate the equations, we use the command `geneqns` of the `linalg` package, which needs the equivalent transpose form $R^t\cdot w^t=0$ of (2). Then we solve them and, since according to the proof of Theorem 3.7 the stress of the first atom can be considered a normalization constant, we take the value 1 for the parameter obtained.

```
> d:=3:  zeros:=[0,0,0]:  with(linalg):
> R:=matrix(10,5*d,[
op(p1-p2),op(p2-p1),op(zeros),op(zeros),op(zeros),
op(p1-p3),op(zeros),op(p3-p1),op(zeros),op(zeros),
op(p1-p4),op(zeros),op(zeros),op(p4-p1),op(zeros),
op(p1-p5),op(zeros),op(zeros),op(zeros),op(p5-p1),
op(zeros),op(p2-p3),op(p3-p2),op(zeros),op(zeros),
op(zeros),op(p2-p4),op(zeros),op(p4-p2),op(zeros),
op(zeros),op(p2-p5),op(zeros),op(zeros),op(p5-p2),
op(zeros),op(zeros),op(p3-p4),op(p4-p3),op(zeros),
op(zeros),op(zeros),op(p3-p5),op(zeros),op(p5-p3),
op(zeros),op(zeros),op(zeros),op(p4-p5),op(p5-p4) ]):
> Rt:=transpose(R):
> eqs:=geneqns(Rt,
[w12,w13,w14,w15,w23,w25,w26,w34,w36,w45,w46,w56],
vector(n*d,0)):
> tensions:=solve(eqs,
w12,w13,w14,w15,w23,w24,w25,w34,w35,w45);
```

$tensions := \{w15 = -2\,w45, w35 = w45, w12 = 2\,w45, w14 = -2\,w45, w23 = -w45, w34 = w45, w13 = -2\,w45, w45 = w45, w25 = -w45, w24 = -w45\}$.

```
> tensions:=subs(w45=1,tensions);
```

$tensions := \{1 = 1, w35 = 1, w12 = 2, w14 = -2, w23 = -1, w34 = 1, w13 = -2, w25 = -1, w24 = -1, w15 = -2\}$.

• Then we have to add the atom p_2, p_3, p_4, p_5, p_6, in which p_6 has unknown coordinates x, y, z and the stress of the atom is determined by the cancelation of tensions at edges $\overline{p_2p_4}$ and $\overline{p_3p_5}$. The same operations as above lead to a second system of equations *eqs2*, in which we substitute $w24 = 1, w35 = -1$ to get

$eqs2subs := \{w23+w25-w26\,x+w26 = 0, -w23-w34-x\,w36 = 0, w34+w45-w46\,x+w46 = 0, -w25-w45-x\,w56 = 0, -w26+w26\,x+x\,w36-w46+w46\,x+x\,w56 = 0, -w26+w26\,y-w36+w36\,y+y\,w46+y\,w56 = 0, -w26+w26\,z+z\,w36+z\,w46-w56+w56\,z = 0, 1+w25-w26\,y+w26 = 0, w23+1-w26\,z+w26 = 0, w34-1-w36\,y+w36 = 0, -w23+1-z\,w36 = 0, -1-w34-y\,w46 = 0, -1-w45-z\,w46 = 0, -w25+1-y\,w56 = 0, -1+w45-w56\,z+w56 = 0\}$.

• In order to look for non-null tension on every edge, Corollary 2.3 turns out to be crucial; if there is a non-null self-stress, then all its tensions are null. Therefore, the self-stress obtained when adding the two atoms is non-null since $w12 = 2 \neq 0$ and this tension is not affected by the second atom. In particular, it is not needed to consider extra variables t_{ij}: In order to obtain the polynomial elimination, we compute a Groebner basis of the polynomials in *eqs2subs*, which we call *polys2subs*, for an elimination order in which variables x, y, z are smaller than the w_{ij}'s. The command `gbasis` of the `Groebner` package is used:

```
with(Groebner):
G:=gbasis(polys2subs,
lexdeg([w23,w25,w26,w34,w36,w45,w46,w56],[x,y,z]));
```

$G := [x^2-x-z^2-y^2+y+z, \ldots$ and 40 polynomials more, involving w_{ij}'s.

We conclude that the hyperboloid $x^2-y^2-z^2-x+y+z = 0$ contains the set of points $p_6 := (x, y, z)$ such that the framework admits a self stress non-null on every edge. Therefore this is a necessary condition. In order to test its sufficiency, we have considered an extra edge in the framework and forced its tension to be null, obtaining the following stress

from the left-kernel of the corresponding rigidity matrix:

$$w := \begin{pmatrix} y+z-1-x \\ -y+x-z+1 \\ -y+x-z+1 \\ -y+x-z+1 \\ -y+x \\ x-z \\ 1 \\ -\dfrac{y(-y+x-z+1)}{y+z+x-1} \\ -\dfrac{-y+z+x-1}{y+z+x-1} \\ -\dfrac{z(-y+x-z+1)}{y+z+x-1} \\ \dfrac{-y+x-z+1}{y+z+x-1} \\ -\dfrac{y-z-1+x}{y+z+x-1} \end{pmatrix}.$$

We omit the (easy) computations checking that:

- Under the condition $x^2-y^2-z^2-x+y+z=0$ of the hyperboloid, this is indeed a self-stress.
- The denominator is not null: We observe that points p_3, p_4 and p_5 already lie on the plane $y+z+x-1=0$. Therefore, if point p_6 lied on that plane, the configuration would not be in general position.
- Similar arguments prove that all the components of w are non-null.

In conclusion, points $p_6=(x,y,z)$ for which a tensegrity exists are precisely those lying on $x^2-y^2-z^2-x+y+z=0$. We now observe that this hyperboloid contains the edges of the quadrilateral $p_2p_3p_4p_5$. Indeed, it is the only one passing through the initial five points and containing those four edges, since the space of all hyperboloids has dimension 9. In order to conclude the first condition in the statement we just have to observe that the combinatorial decompositions which at step 2 of Algorithm 3.4 choose vertices 1 or 6, lead to the 4-cycle 2345, those choosing 3 or 5 lead to 1264, and those choosing 2 or 4 to 1365 (see Figure 2).

For the equivalent condition in the statement, which appears in [**5**], easy computations check that the four planes defined by $p_1p_2p_3$, $p_1p_4p_5$, $p_3p_4p_6$ and $p_2p_5p_6$ intersect precisely for $x^2 - y^2 - z^2 - x + y + z = 0$, and the same condition is obtained for the other four alternate planes.

Let us finally note that an interested reader can find in [**8**] a different symbolic approach to the resolution of this problem, that uses comprehensive Groebner basis.

Remark 4.4. We have to call the attention of the reader to the fact that, in spite of dealing with more complicated problems than the geometric approach, the usefulness of the algebraic one is also limited by the size and type of the problem. For instance, if Corollary 2.3 cannot be used and all the inequations $w_{ij} \neq 0$ have to be considered in order to look for non-null tension on every edge. This would introduce more auxiliary variables $t_{12}, t_{13}, \ldots, t_{56}$ and a polynomial $(w_{12}t_{12} - 1)(w_{13}t_{13} - 1)\cdots(w_{56}t_{56} - 1)$, making the computations infeasible.

Finally, let us recall that the above symbolic computations follow the decomposition-reconstruction method from Section 3. We already noted that following Observation 4.2 it is also possible to compute symbolically the kernel of the 12×18 rigidity matrix $R(P)$ for the whole framework $G(P)$. Instead, with the decomposition-reconstruction approach we compute the kernel of the two 10×15 submatrices corresponding to the two atoms. The main advantage of the decomposition reconstruction method is related to this fact: Since $w24$ and $w35$ were no longer variables and the system had fewer equations, the computation of the Groebner basis was easier. In addition, we did not need to introduce an extra variable t_{ij}.

Acknowledgements

Both authors would like to thank the initial help and later encouraging comments and revisions by professors Juan Llovet and Tomás Recio, specially about the algebraic part. The second author also wants to thank an anonymous reviewer for several valuable comments that helped improving the results in the paper.

Final note

The present paper is dedicated to the memory of Miguel de Guzmán, who left us before it was finished. This work started when he gave a conference about tensegrities at the Departamento de Matemáticas of the Universidad de Alcalá in November 2003. Five months later he died

after a seminal version had been finished. Although part of the results presented here still had to be formulated and proved in their final form, their essence was already contained in that preliminary version.

References

[1] D. L. D. Caspar and A. Klug, Physical principles in the construction of regular viruses, in: *"Proceedings of Cold Spring Harbor Symposium on Quantitative Biology"* **27**, Cold Spring Harbor, N.Y., 1962, pp. 1–24.

[2] CoCoATeam, *"CoCoA: a system for doing Computations in Commutative Algebra"*, 1995, available at `http://cocoa.dima.unige.it`.

[3] R. Connelly and W. Whiteley, Second-order rigidity and prestress stability for tensegrity frameworks, *SIAM J. Discrete Math.* **9(3)** (1996), 453–491.

[4] D. Cox, J. Little and D. O'Shea, *"Ideals, varieties, and algorithms. An introduction to computational algebraic geometry and commutative algebra"*, Undergraduate Texts in Mathematics, Springer-Verlag, New York, 1992.

[5] H. Crapo, Structural rigidity, *Structural Topology* **1** (1979), 26–45, 73.

[6] M. de Guzmán and D. Orden, Finding tensegrity structures: Geometric and symbolic aproaches, in: *"Proceedings of EACA-2004"*, (L. González-Vega, T. Recio, eds.), Universidad de Cantabria, Santander, 2004, pp. 167–172.

[7] D. E. Ingber, Cellular tensegrity: defining new rules of biological design that govern the cytoskeleton, *Journal of Cell Science* **104** (1993), 613–627.

[8] M. Manubens and A. Montes, Improving DISPGB algorithm using the discriminant ideal, to appear in the special A3L issue of *J. Symbolic Comput.* (2006), available at `http://www.arxiv.org/abs/math.AC/0601763`.

[9] R. Motro, *"Tensegrity: Structural systems for the future"*, Kogan Page Science, London, 2003.

[10] G. Rote, F. Santos and I. Streinu, Expansive motions and the polytope of pointed pseudo-triangulations, in: *"Discrete and computational geometry"*, Algorithms Combin. **25**, Springer, Berlin, 2003, pp. 699–736.

[11] B. Roth and W. Whiteley, Tensegrity frameworks, *Trans. Amer. Math. Soc.* **265(2)** (1981), 419–446.

[12] R. E. SKELTON, Deployable tendon-controlled structure, *United States Patent* **5 642 590** (1997).
[13] K. SNELSON, Available at: `http://www.kennethsnelson.net`.
[14] J. SZABÓ AND L. KOLLÁR, *"Structural design of cable suspended roofs"*, Ellis Horwood, Chichester, 1984.
[15] A. G. TIBERT, Deployable tensegrity structures for space applications, Ph.D. Thesis, Royal Institute of Technology, Stokholm (2002).
[16] A. G. TIBERT AND S. PELLEGRINO, Review of form-finding methods for tensegrity structures, *International Journal of Space Structures* **18(4)** (2003). 209–223(15).
[17] WATERLOO MAPLE INC., *"Maple"*, Software, available at `http://www.maplesoft.com`.
[18] W. WHITELEY, Rigidity and scene analysis, in: *"Handbook of discrete and computational geometry"*, CRC Press Ser. Discrete Math. Appl., CRC, Boca Raton, FL, 1997, pp. 893–916.

Departamento de Matemáticas
Facultad de Ciencias
Universidad de Alcalá
E-28871 Alcalá de Henares (Madrid)
Spain
E-mail address: `david.orden@uah.es`

Primera versió rebuda el 10 de març de 2005,
darrera versió rebuda el 13 de març de 2006.