Rigid tensegrity labelings of graphs

Tibor Jordán\textsuperscript{a}, András Recski\textsuperscript{b}, Zoltán Szabadka\textsuperscript{a}

\textsuperscript{a} Department of Operations Research, Eötvös University, Pázmány Péter sétány 1/C, 1117 Budapest, Hungary
\textsuperscript{b} Department of Computer Science and Information Theory, Budapest University of Technology and Economics, H-1521 Budapest, Hungary

\textbf{ARTICLE INFO}

\textbf{Article history:}
Available online 8 January 2009

\textbf{ABSTRACT}

Tensegrity frameworks are defined on a set of points in $\mathbb{R}^d$ and consist of bars, cables, and struts, which provide upper and/or lower bounds for the distance between their endpoints. The graph of the framework, in which edges are labeled as bars, cables, and struts, is called a tensegrity graph. It is said to be rigid in $\mathbb{R}^d$ if it has an infinitesimally rigid realization in $\mathbb{R}^d$ as a tensegrity framework. The characterization of rigid tensegrity graphs is not known for $d \geq 2$.

A related problem is how to find a rigid labeling of a graph using no bars. Our main result is an efficient combinatorial algorithm for finding a rigid cable–strut labeling of a given graph in the case when $d = 2$. The algorithm is based on a new inductive construction of redundant graphs, i.e. graphs which have a realization as a bar framework in which each bar can be deleted without increasing the degree of freedom. The labeling is constructed recursively by using labeled versions of some well-known operations on bar frameworks.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

A tensegrity graph $T = (V; B, C, S)$ is a simple graph on vertex set $V = \{v_1, v_2, \ldots, v_n\}$ whose edge set is partitioned into three pairwise disjoint sets $B, C,$ and $S$, called bars, cables, and struts, respectively. The elements of $E = B \cup C \cup S$ are the members of $T$. A tensegrity graph containing no bars is called a cable–strut tensegrity graph. The underlying graph of $T$ is the (unlabeled) graph $\overline{T} = (V; E)$. A $d$-dimensional tensegrity framework is a pair $(T, p)$, where $T$ is a tensegrity graph and $p$ is a map from $V$.
to $\mathbb{R}^d$. $(T, p)$ is also called a realization of $T$. If $T$ has neither cables nor struts then we call it a bar graph and a realization $(T, p)$ is said to be a bar framework.

An infinitesimal motion of a tensegrity framework $(T, p)$ is an assignment of infinitesimal velocities $u_i \in \mathbb{R}^d$ to the vertices, such that

$$(p_i - p_j)(u_i - u_j) = 0 \quad \text{for all } ij \in B,$$
$$(p_i - p_j)(u_i - u_j) \leq 0 \quad \text{for all } ij \in C,$$
$$(p_i - p_j)(u_i - u_j) \geq 0 \quad \text{for all } ij \in S.$$ An infinitesimal motion is trivial if it can be obtained as the derivative of a rigid congruence of all of $\mathbb{R}^d$ restricted to the vertices of $(T, p)$. The tensegrity framework $(T, p)$ is infinitesimally rigid in $\mathbb{R}^d$ if all of its infinitesimal motions are trivial. A tensegrity graph $T$ is said to be rigid in $\mathbb{R}^d$ if it has an infinitesimally rigid realization $(T, p)$ in $\mathbb{R}^d$. We refer the reader to [1–4] for more details on the rigidity of tensegrity frameworks.

The characterization of rigid tensegrity graphs is not known for $d \geq 2$. (The solution for $d = 1$ can be found in [5]). The special case of rigid bar graphs has been solved for $d = 2$, but the three-dimensional version remains one of the major open problems in combinatorial rigidity.

In this paper we consider a related labeling problem. Given a graph $G = (V, E)$, how to find a cable–strut labeling $E = C \cup S$ of the edges for which the resulting tensegrity graph $T = (V; C \cup S)$ is rigid in $\mathbb{R}^d$. We observe that $G$ has such a rigid cable–strut labeling if and only if $G$ has a redundantly rigid realization as a bar framework in $\mathbb{R}^d$.

Our main result is an efficient combinatorial algorithm for finding a rigid cable–strut labeling, if it exists, in the case when $d = 2$. The algorithm is based on a new inductive construction of redundant graphs, i.e. graphs which have a realization as a bar framework in which each bar can be deleted without increasing the degree of freedom. The labeling is constructed recursively, following the steps of the construction, by using labeled versions of some well-known operations on bar frameworks. We note that our algorithm does not yield an infinitesimally rigid realization of the labeled graph but is designed to only find the labels.

2. Preliminaries

A stress of a tensegrity framework $(T, p)$ is an assignment of scalars $\omega_{ij}$ to the members $ij$ of $T$ satisfying $\omega_{ij} \leq 0$ for cables, $\omega_{ij} \geq 0$ for struts and

$$\sum_{ij \in E} \omega_{ij}(p_i - p_j) = 0 \quad \text{for each } i \in V.$$ Following [3] we say that $\omega = (\ldots, \omega_{ij}, \ldots) \in \mathbb{R}^E$ is a proper stress, if $\omega_{ij} \neq 0$ for all $ij \in C \cup S$. (Note that more recent papers, e.g. [6,1,2,4], use proper stress and strict proper stress for what we call stress and proper stress, respectively.) The following basic results on infinitesimally rigid tensegrity frameworks are due to Roth and Whiteley, see [3, Theorem 5.2(c), Corollary 5.3].

**Theorem 2.1** ([3]). Suppose that $(T, p)$ is a tensegrity framework in $\mathbb{R}^d$. Then

(i) $(T, p)$ is infinitesimally rigid in $\mathbb{R}^d$ if and only if the bar framework $(\bar{T}, p)$ is infinitesimally rigid in $\mathbb{R}^d$ and there exists a proper stress of $(T, p)$;

(ii) If $(T, p)$ is infinitesimally rigid then the bar framework obtained by deleting any cable or strut of $T$ and replacing the remaining members of $T$ by bars is infinitesimally rigid in $\mathbb{R}^d$.

The rigidity matrix of a tensegrity framework $(T, p)$ on $n$ vertices is the matrix $R(T, p)$ of size $|E| \timesdn$, where, for each member $v_i v_j \in E$, in the row corresponding to $v_i v_j$, the entries in the $d$ columns corresponding to vertices $i$ and $j$ contain the $d$ coordinates of $(p(v_i) - p(v_j))$ and $(p(v_i) - p(v_j))$, respectively, and the remaining entries are zeros. Note that a stress of $(T, p)$ corresponds to a row dependency of $R(T, p)$.

A configuration $p \in \mathbb{R}^{dn}$ is a regular point of $T$ if rank $R(T, p) = \max \{\text{rank} R(T, q) : q \in \mathbb{R}^{dn}\}$. It is generic if it also gives rise to a regular point for all non-empty edge-induced subgraphs of $T$. Note that the regular (generic) points of $T$ form an open and dense subset of $\mathbb{R}^{dn}$. We also say that a realization
(T, p) is regular (generic) if p is a regular (generic, respectively) point of T. A member e of T is redundant in (T, p) if rank R(T, p) = rank R(T − e, p). The proof of the next lemma is implicit in the proof of [3, Theorem 5.4].

**Theorem 2.2** ([3]). Let (T, p) be a regular realization of tensegrity graph T in \( \mathbb{R}^d \), let \( \omega \) be a proper stress of (T, p), and let e be a redundant member in (T, p). Then the set

\[
\{ q \in \mathbb{R}^{dn} : (T, q) \text{ is a regular realization of } T \text{ which has a proper stress } \omega' \text{ with } \omega'(e) = \omega(e) \}
\]

is open in \( \mathbb{R}^{dn} \).

Note that the infinitesimal velocities of a bar framework \((G, p)\) are the vectors in the null space of \( R(G, p) \). Hence \((G, p)\) is infinitesimally rigid in \( \mathbb{R}^d \) if and only if rank \( R(G, p) = \max\{ \text{rank } (K_q, q) : q \in \mathbb{R}^{dn} \} \), where \( K_q \) is the complete graph on \( n \) vertices. It also follows that infinitesimal rigidity of graphs is a generic property: \( G \) has an infinitesimally rigid realization if and only if all generic realizations are infinitesimally rigid. A graph \( G \) is said to be rigid in \( \mathbb{R}^d \) if it has an infinitesimally rigid realization as a bar framework in \( \mathbb{R}^d \). The characterization of rigid graphs in \( \mathbb{R}^d \) is known for \( d \leq 2 \), see e.g. [7]. We say that \( G \) is redundantly rigid if \( G - e \) is rigid for all \( e \in E \). Clearly, \( G \) is the underlying graph of a rigid tensegrity graph if and only if \( G \) is rigid — simply label each edge as a bar. If bars are not allowed then we may use **Theorem 2.1** to deduce the following characterization.

**Theorem 2.3.** A graph \( G = (V, E) \) is the underlying graph of a rigid cable–strut tensegrity graph in \( \mathbb{R}^d \) if and only if \( G \) is redundantly rigid in \( \mathbb{R}^d \).

**Proof.** Necessity follows from **Theorem 2.1**(ii). To prove sufficiency consider a generic realization \((G, p)\) of \( G \) in \( \mathbb{R}^d \) as a bar framework. Since \( G \) is redundantly rigid, \((G, p)\) is infinitesimally rigid and each edge of \( G \) is redundant in \((G, p)\). This implies that for each edge \( e \in E \) there is a set of edges \( C_e \subseteq E \) with \( e \in C_e \) for which there is a proper stress \( \omega_e \) in the subframework consisting of the bars of \( C_e \). By using the fact that the stresses of the bar framework \((G, p)\) form a linear subspace of \( \mathbb{R}^d \), we can deduce that there exist non-zero coefficients \( \lambda_e \in \mathbb{R}, e \in E \), for which \( \omega = \sum_{e \in E} \lambda_e \omega_e \) is a proper stress for \((G, p)\). Let \( C = \{ (i, j) \in E : \omega_{ij} < 0 \} \) and \( S = \{ (i, j) \in E : \omega_{ij} > 0 \} \). This labeling gives rise to a rigid cable–strut tensegrity graph \( T \) with underlying graph \( G \) by **Theorem 2.1**(i). \( \square \)

3. Redundant graphs in two dimensions

In the rest of the paper we shall assume that \( d = 2 \). By **Theorem 2.3** we may suppose that the input of the cable–strut labeling problem is a redundantly rigid graph \( G \). Our goal is to find an inductive construction of \( G \), i.e. a sequence \( G_1, G_2, \ldots, G_r \) of graphs for which \( G_{r+1} \) is obtained from \( G_r \) by some graph operation, \( 1 \leq i \leq r - 1 \), and \( G_r = G \), in such a way that (i) each \( G_i \) has a ‘good’ cable–strut labeling, (ii) the operations are chosen so that, given a ‘good’ cable–strut labeling of \( G_i \), a ‘good’ cable–strut labeling of \( G_{r+1} \) is easy to find, (iii) the cable–strut labeling of \( G \) gives rise to a rigid tensegrity graph. To make this idea work we need to consider redundant graphs, a family of graphs which properly contains redundantly rigid graphs, and call a labeling ‘good’ if the corresponding tensegrity graph has a regular realization with a proper stress.

We say that a graph \( H = (V, E) \) is a circuit if for all generic realizations \((H, p)\) of \( H \) the rows of \( R(H, p) \) form a minimal linearly dependent set of vectors in \( \mathbb{R}^{2n} \). We say that \( G \) is redundant if it has at least one edge and each edge of \( G \) is in a circuit. It follows that a graph \( G \) is redundantly rigid if and only if \( G \) is rigid and redundant. It is known that circuits are redundantly rigid graphs. See [8,9] for more details on the properties of circuits (which are also called 2-circuits, M-circuits, or generic cycles in the literature).

A \( j \)-separation of a graph \( G = (V, E) \) is a pair \((G_1, G_2)\) of edge-disjoint subgraphs of \( G \) each with at least \( j + 1 \) vertices such that \( G = G_1 \cup G_2 \) and \( |V(G_1) \cap V(G_2)| = j \). We say that \( G \) is \( 3 \)-connected if \( G \) has at least 4 vertices and has no \( j \)-separation for all \( 0 < j \leq 2 \). If \( G_1, G_2 \) is a \( 2 \)-separation of \( G \), then we say that \( V(G_1) \cap V(G_2) \) is a \( 2 \)-separator of \( G \).

Suppose that \( G = (V, E) \) is a 2-connected graph and let \((G_1, G_2)\) be a 2-separation of \( G \) with \( V(G_1) \cap V(G_2) = \{ u, v \} \). For \( 1 \leq i \leq 2 \), let \( G'_i = G_i + uv \) if \( uv \notin E(G_i) \) and otherwise put \( G'_i = G_i \). We say that \( G'_1, G'_2 \) are the cleavage graphs obtained by cleaving \( G \) along \( \{ u, v \} \).
Fig. 2. Suppose that $G$ is a 2-connected redundant graph. Let $\tilde{1}$ denote struts (resp. cables).

six possible labeled 1-extensions on a strut $u \rightarrow v$ of vertex $t \in V$.

Lemma 3.1. Suppose that $G$ is a 2-connected redundant graph. Let $\{u, v\}$ be a 2-separator of $G$ and let $H_1$ and $H_2$ be the cleavage graphs obtained by cleaving $G$ along $\{u, v\}$. Then at least one of the following holds:

(i) $H_i$ is redundant for $i = 1, 2$;

(ii) there is a 2-separation $(H_1, H_2)$ of $G$ with $V(H_1) \cap V(H_2) = \{u, v\}$ for which $H_i$ is redundant for $i = 1, 2$.

Proof. First we prove that each edge $f \in E(\tilde{H}_1) - uv$ belongs to a circuit in $\tilde{H}_1$. Since $G$ is redundant, there is a circuit $C$ in $G$ which contains $f$. If $C$ is a subgraph of $\tilde{H}_1$ then we are done. If not, then $\{u, v\}$ is a 2-separator of $C$. In this case it follows from [10, Lemma 4.2] that the cleavage graphs $C_1$ and $C_2$ obtained by cleaving $C$ along $\{u, v\}$ are both circuits. Hence $C_1$ is a circuit in $\tilde{H}_1$ which contains $f$. By symmetry we also have that each edge $f' \in E(\tilde{H}_2) - uw$ belongs to a circuit in $\tilde{H}_2$.

Thus, if $uv$ belongs to a circuit in both cleavage graphs then (i) holds. Now suppose that, say, $uv$ is in no circuit in $\tilde{H}_1$. As above, this implies that if $uv \in E(G)$ then all circuits of $G$ containing $uv$ must be in $\tilde{H}_2$ and if $uv \notin E(G)$ then all circuits of $G$ containing some edge of $E(\tilde{H}_1) - uv$ must be in $\tilde{H}_1 - uv$.

By moving the edge $uv$ from one side of the 2-separation to the other, if necessary, we may assume that there is a 2-separation $(H_1, H_2)$ of $G$ with $V(H_1) \cap V(H_2) = \{u, v\}$ and $uv \notin E(H_1)$. The arguments above now imply that $H_1$ and $H_2$ are both redundant. Thus (ii) holds. □

We need the following result on redundant graphs. It follows by observing that the proof of [9, Theorem 3.2] goes through under the weaker hypothesis that $G$ is redundant, and by using [9, Lemma 3.1].

Theorem 3.2 ([9]). Suppose that $G$ is a 3-connected redundant graph. Then $G$ is redundantly rigid.

The 1-extension operation (on edge $uw$ and vertex $t$) deletes an edge $uw$ from a graph $G$ and adds a new vertex $v$ and new edges $uv, vw, vt$ for some vertex $t \in V(G) - \{u, w\}$. The following result gives an inductive construction for 3-connected redundantly rigid graphs.

Theorem 3.3 ([9, Theorem 6.15]). Let $G$ be a 3-connected redundantly rigid graph. Then $G$ can be obtained from $K_4$ by a sequence of 1-extensions and edge additions.

4. Operations on tensegrity graphs

In this section we introduce the ‘labeled generalizations’ of the 1-extension and 2-sum operations and show that they preserve rigidity when applied to tensegrity graphs. These operations, whose unlabeled versions are well-known in combinatorial rigidity, will be used in the next section to define rigid cable–strut labelings of graphs.

Let $T = (V; B \cup C \cup S)$ be a tensegrity graph, let $uw \in C \cup S$ be a cable or strut of $T$ and let $t \in V - \{u, w\}$ be a vertex. The labeled 1-extension operation deletes the member $uw$, adds a new vertex $v$ and new members $u\tilde{v}$, $\tilde{v}u$, $\tilde{v}t$, satisfying the condition that if $uw$ is a cable then at least one of $u\tilde{v}$, $v\tilde{u}$ is not a strut, and if $uw$ is a strut then at least one of $v\tilde{u}$, $u\tilde{v}$ is not a cable. The new member $\tilde{v}t$ may be arbitrary. For example, if we consider cable–strut tensegrity graphs, this definition leads to six possible labeled 1-extensions on a strut $uw$, as illustrated in Fig. 2.
**Lemma 4.1.** Let $T$ be a rigid tensegrity graph and let $T'$ be a tensegrity graph obtained from $T$ by a labeled 1-extension. Then $T'$ is also rigid.

**Proof.** Since infinitesimal rigidity (and the labeled 1-extension operation) is preserved by interchanging cables and struts, we may assume that the 1-extension is made on a strut $uw$ of $T$ and vertex $t \in V - \{u, w\}$. Let $(T, p)$ be an infinitesimally rigid realization of $T$ in $\mathbb{R}^2$. By Theorem 2.1 there is a proper stress $\omega$ of $(T, p)$ and $(G, p)$ is an infinitesimally rigid bar framework, where $G = \overline{T}$ is the underlying graph of $T$. By Theorem 2.2 we may assume that $p(u), p(w), p(t)$ are not collinear. In the rest of the proof we shall also assume that the new members $vu, vw, vt$ are all struts. The proof is similar for each of the six possible labeled 1-extensions.

Let us extend the configuration $p$ by putting $p(v) = \alpha p(u) + (1 - \alpha)p(w)$ for some $0 < \alpha < 1$. Let $G'$ be the underlying graph of $T'$, which can be obtained from $G$ by a 1-extension. We can also extend the stress $\omega$ of $(T, p)$ to $(T', p)$ by defining $\omega_{vu} = \omega_{uw}/(1 - \alpha)$, $\omega_{vw} = \omega_{uw}/\alpha$ and $\omega_{vt} = 0$.

Since $p(u), p(w), p(t)$ are not collinear, the bar framework $(G', p)$ is infinitesimally rigid, see [7, Theorem 2.2.2]. Furthermore, the extended stress is nearly proper on $(T', p)$: the only member with a zero stress is $vt$. This implies that $(T'', p)$ is infinitesimally rigid, where $T''$ is obtained from $T'$ by replacing the strut $vt$ by a bar.

To obtain a proper stress we need to modify the realization a bit by replacing $p(v)$ by a point in the interior of the triangle $p(u)p(w)p(t)$. By Theorem 2.2 this can be done without destroying the infinitesimal rigidity of $(T'', p)$. Consider a proper stress $\omega'$ of this modified realization of $T''$. Since we have three members incident with $v$, and $vu, vw, vt$ are struts, we must have a positive stress on $vt$. Thus we may replace $vt$ by a strut and obtain the required infinitesimally rigid realization of $T'$.

The other labeled 1-extensions can be treated in a similar manner by appropriately defining $\alpha \in \mathbb{R} - \{0, 1\}$ and moving $p(v)$ out of the line of $p(u)p(w)$ in such a way that the signs of the stresses on the members incident to $v$ are as required. See Fig. 2. □

We shall also need an operation that glues together two tensegrity graphs along a pair of members.

Let $T_1 = (V_1; B_1, C_1, S_1)$ and $T_2 = (V_2; B_2, C_2, S_2)$ be two tensegrity graphs with $V_1 \cap V_2 = \emptyset$ and let $u_1v_1 \in S_1$ and $u_2v_2 \in C_2$ be two designated members, a strut in $T_1$ and a cable in $T_2$. The 2-sum of $T_1$ and $T_2$ (along the strut–cable pair $u_1v_1$ and $u_2v_2$) is the tensegrity graph obtained from $T_1 - u_1v_1$ and $T_2 - u_2v_2$ by identifying $u_1$ with $u_2$ and $v_1$ with $v_2$. See Fig. 3. We denote a 2-sum of $T_1$ and $T_2$ by $T_1 \oplus_2 T_2$. Since we shall apply the 2-sum operation to non-rigid tensegrity graphs as well, we first prove the following lemma.

**Lemma 4.2.** Let $(T_1, p_1)$ and $(T_2, p_2)$ be regular realizations of tensegrity graphs $T_1, T_2$ with a proper stress. Then $T = T_1 \oplus_2 T_2$ also has a regular realization with a proper stress.

**Proof.** By Theorem 2.2 we may assume that $(T_i, p_i)$ is generic for $i = 1, 2$. Let $\omega_i$ be a proper stress of $(T_i, p_i), i = 1, 2$. By scaling, translating, and rotating the frameworks, if necessary, we may assume that $p_1(u_1) = p_2(u_2)$ and $p_1(v_1) = p_2(v_2)$. These operations will not destroy genericity and $\omega_i$ remains a proper stress in the realization of $T_i$ for $i = 1, 2$. By scaling the stresses we can also assume that $\omega_1(u_1v_1) = -\omega_2(u_2v_2) = 1$. Since the realizations are generic, it follows from Theorem 2.1(ii) that $u_1v_1$ and $u_2v_2$ are both redundant.
Let $T'$ be the tensegrity graph obtained from $T_1, T_2$ by identifying $u_1$ with $u_2$, and $v_1$ with $v_2$. Consider the realization $(T', p)$ of $T'$ obtained by merging the frameworks $(T_i, p_i), i = 1, 2$, along the points $p_1(u_1), p_1(v_1)$. We can find generic realizations of $T'$ arbitrarily close to $(T', p)$ without changing the positions of $p(u), p(v)$. Now we can use Theorem 2.2, applied to each $(T_i, p_i)$, and the fact that $u_i v_i$ is redundant in $(T_i, p_i), i = 1, 2$, to deduce that there is an $\epsilon > 0$ for which any regular realization of $T'$ in the $\epsilon$-neighbourhood has a proper stress whose value is equal to 1 on the strut $u_1 v_1$ and $-1$ on the cable $u_2 v_2$. Since the stresses on $u_1 v_1$ and $u_2 v_2$ cancel each other, we have that $(T, p)$ is a regular realization of $T = T_1 \oplus T_2$ with a proper stress. This proves the lemma. \hfill \Box

Theorem 2.1 and the gluing lemma [7, Lemma 3.1.4] give the following corollary.

**Lemma 4.3.** Suppose that $T_1$ and $T_2$ are rigid tensegrity graphs. Then $T = T_1 \oplus T_2$ is also rigid.

5. Cable–strut labelings of redundant graphs

In this section we give an algorithmic proof for the existence of a 'good' labeling of a redundant graph $G$. It will also imply an inductive construction for redundant graphs as well as an efficient combinatorial algorithm for finding a rigid cable–strut labeling in the special case when $G$ is redundantly rigid.

**Theorem 5.1.** Let $G = (V, E)$ be a redundant graph in $\mathbb{R}^2$. Then the edge set of $G$ has a cable–strut labeling $E = C \cup S$ for which the tensegrity graph $T = (V; C \cup S)$ has a regular realization with a proper stress.

**Proof.** We prove the theorem by induction on $|V|$. Since $G$ is redundant and the smallest circuit is $K_4$, we must have $|V| \geq 4$ with equality only if $G = K_4$. The statement is straightforward for $K_4$ (see Fig. 1), so we may assume that $|V| \geq 5$ and that the theorem holds for all redundant graphs containing less vertices than $G$.

First suppose that $G$ has at least two blocks (i.e. maximal 2-connected subgraphs), denoted by $H_1, H_2, \ldots, H_s$. Since circuits are 2-connected, each block is redundant. Thus, by induction, we can find a cable–strut labeling of each block $H_i$ such that the corresponding tensegrity graph $T_i$ has a regular realization with a proper stress. Let $T$ be the tensegrity graph on $G$ whose cable–strut labeling is induced by the $T_i$'s. Since a proper stress remains a proper stress after translating a framework, and since the blocks of $G$ are edge-disjoint, we may obtain a realization $(T, p)$ of $T$ with a proper stress by simply translating and merging the realizations of the $T_i$'s at the cut-vertices of $G$. Since the regular realizations of $T$ form a dense open set, we can use Theorem 2.2, applied to each of the realizations of the $T_i$'s, to make the realization regular. This shows that $G$ has the required labeling.

Hence we may assume that $G$ is 2-connected. If $G$ is 3-connected then $G$ can be obtained from $K_4$ by 1-extensions and edge additions by Theorems 3.2 and 3.3. Thus we can obtain a rigid cable–strut tensegrity graph $T$ with underlying graph $G$ by starting with a rigid cable–strut labeling of $K_4$ and using labeled 1-extensions as well as cable or strut additions, following the inductive construction of $G$. Lemma 4.1 implies that the labeled graph is indeed rigid. (The addition of new members clearly preserves rigidity.) Since an infinitesimally rigid realization of $T$ is regular and has a proper stress by Theorem 2.1, the existence of the required cable–strut labeling of $G$ follows.

It remains to consider the case when $G$ is 2-connected and has a 2-separation $\{u,v\}$. Let $\tilde{H}_1, \tilde{H}_2$ be the cleavage graphs obtained by cleaving $G$ along $\{u,v\}$. By Lemma 3.1 either $G$ can be obtained as the edge-disjoint union of two redundant graphs with two vertices in common or both cleavage graphs are redundant. In the former case we can proceed as in the case of 1-separations: by induction, we
can find cable–strut labelings of the smaller graphs for which the required realizations exist. These labelings induce a cable–strut labeling \( T \) of \( G \). Furthermore, by first rotating, translating, and scaling the frameworks, if necessary, we can merge the realizations to obtain a realization of \( T \) with a proper stress. By perturbing this realization, and using Theorem 2.2, we can make the realization regular, too. This shows that \( G \) has the required labeling.

In the latter case we can also find, by induction, cable–strut labelings of the cleavage graphs which have the desired realizations. By Theorem 2.2 we may assume that these realizations are generic. After interchanging cables and struts in one of the cleavage graphs, if necessary, we can take the 2-sum of the labeled cleavage graphs to obtain a labeling \( T' = G - uv \) which has a regular realization \((T', p)\) with a proper stress, by Lemma 4.2. If \( uv \notin E(G) \) then this provides the required labeling of \( G \).

Now suppose that \( uv \in E(G) \). By Theorem 2.2 we may assume that \( p \) is chosen so that \((T' + uv, p)\) is generic. Then \( uv \) is redundant in \((T' + uv, p)\), and hence there is a stress \( \omega' \) of \((T' + uv, p)\) whose value on \( uv \) is not zero. By adding \( \omega' \) to \( \omega \) with a small coefficient we obtain a proper stress of \((T' + uv, p)\). Thus adding a new cable (or strut) \( uv \) to the labeled graph \( T' \) gives the required labeling of \( G \) in this case. This completes the proof of the theorem.

The proof implies that if \( G \) is redundant then \( G \) be obtained from disjoint \( K_4 \)'s by recursively applying 1-extensions or edge additions within some connected component, 2-sums to two connected components, and merging components along at most two vertices. This inductive construction can be obtained in polynomial time by using efficient algorithms to find 2-separators [11] and test redundancy [12]. By following the steps of the construction it is then straightforward to find a good labeling of \( G \) by applying labeled 1-extensions, cable or strut additions, taking 2-sums (possibly after interchanging cables and struts in one of the summands), and merging.

When \( G \) is redundantly rigid, Theorems 2.1 and 5.1, and the above argument imply:

**Theorem 5.2.** Let \( G = (V, E) \) be a redundantly rigid graph in \( \mathbb{R}^2 \). Then the edge set of \( G \) has a cable–strut labeling \( E = C \cup S \) for which the tensegrity graph \( T = (V; C \cup S) \) is rigid. Furthermore, such a cable–strut labeling of \( E \) can be found in polynomial time.

We note that it is fairly easy to extend the above results to the case when the input graph may contain multiple edges and/or a designated set of edges labeled as bars and the goal is to find a cable–strut labeling of the remaining edges so that the union of the bars, cables, and struts gives rise to a rigid tensegrity graph.

We also remark that the proof of Theorem 5.1 can be used to verify the existence of rigid cable–strut labelings with various structural properties. For example, consider a 3-connected circuit \( G \) on at least five vertices. Then \( G \) has a rigid cable–strut labeling in which the cables as well as the struts induce a spanning tree of \( G \). This follows from the facts that (i) the wheel graph \( W_5 \) on five vertices (which is obtained from \( K_4 \) by a 1-extension) has a rigid cable–strut labeling in which the cables as well as the struts induce a spanning tree (c.f. Fig. 4), (ii) \( G \) can be obtained from \( W_5 \) by 1-extensions by Theorem 3.3, (iii) when labeling the edges of \( G \) using this inductive construction it is possible to choose labeled 1-extensions so that the spanning trees are 'preserved'. One can similarly verify that if \( G \) contains a triangle then \( G \) has a rigid cable–strut labeling in which the cables induce a single triangle and all other members are struts. This follows by using [10, Theorem 5.9], which implies that the inductive construction can be chosen so that the edges of a designated triangle of the starting \( K_4 \) are never involved in the 1-extensions.

6. Rigid cable–strut tensegrity graphs

As we noted earlier, the characterization of the rigid (cable–strut) tensegrity graphs is still open, even in two dimensions. In one dimension it turns out that a cable–strut tensegrity graph \( T \) is rigid if and only if its underlying graph is rigid (i.e. connected) and each of its \( M \)-connected components\(^{1}\)

---

\(^{1}\) A graph is \( M \)-connected if each pair of its edges belongs to a circuit. Hence circuits are \( M \)-connected. The \( M \)-connected components of a graph \( G \) are the maximal \( M \)-connected subgraphs of \( G \). It is known that \( M \)-connected graphs are redundantly rigid and 3-connected redundantly rigid graphs are \( M \)-connected. It is also known that the 2-sum of two \( M \)-connected graphs is \( M \)-connected. See [9] for more details.
Conjecture 6.1. There exists an (smallest) integer \( k \) such that every tensegrity graph \( T \) containing at least \( kc \) cables and at least \( ks \) struts, and with a 3-connected and redundantly rigid underlying graph, is rigid in \( \mathbb{R}^2 \).

The characterization of rigid tensegrity graphs whose underlying graph is a complete graph or a wheel has been given in [13]. The wheels show that if \( k \) exists, it must be at least five. See drawing (J) of Fig. 4.

Acknowledgements

The first author was supported by the MTA-ELTE Egerváry Research Group on Combinatorial Optimization, and the Hungarian Scientific Research Fund Grant No. T49671, K60802. The second author was supported by the Hungarian Scientific Research Fund Grant No. K60802, T67651.

References


Fig. 4. The non-rigid tensegrity graphs on \( W_5 \).

