Egerváry Research Group
on Combinatorial Optimization

Technical reportS

TR-2012-05. Published by the Egerváry Research Group, Pázmány P. sétány 1/C, H–1117, Budapest, Hungary. Web site: www.cs.elte.hu/egres . ISSN 1587–4451.

Strongly Rigid Tensegrity Graphs on the Line

Bill Jackson, Tibor Jordán, and Csaba Király

May 31, 2012

Strongly Rigid Tensegrity Graphs on the Line

Bill Jackson[⋆], Tibor Jordán[⋆⋆], and Csaba Király[⋆⋆⋆]

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 strongly rigid in $\mathbb{R}^d$ if every generic realization in $\mathbb{R}^d$ as a tensegrity framework is infinitesimally rigid. In this note we show that it is NP-hard to test whether a given tensegrity graph is strongly rigid in $\mathbb{R}^1$.

1 Introduction

A *tensegrity graph* $T = (V; B, C, S)$ is a graph on vertex set $V = \{v_1, v_2 \dots, v_n\}$ whose edge set is partitioned into three 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*[1]. 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 $\mathbb{R}^d$. We will also refer to (T, p) as 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 $q : V \to \mathbb{R}^d$ of infinitesimal velocities to the vertices, such that

$$\begin{aligned} (p_i - p_j)(q_i - q_j) &= 0 \quad \text{for all } v_iv_j \in B, \\ (p_i - p_j)(q_i - q_j) &\le 0 \quad \text{for all } v_iv_j \in C, \\ (p_i - p_j)(q_i - q_j) &\ge 0 \quad \text{for all } v_iv_j \in S, \end{aligned}$$

⋆School of Mathematical Sciences, Queen Mary University of London, Mile End Road, London E1 4NS, England, UK. e-mail: `b.jackson@qmul.ac.uk`

⋆⋆Department of Operations Research, Eötvös University, Pázmány Péter sétány 1/C, 1117 Budapest, Hungary. e-mail: `jordan@cs.elte.hu`.

⋆⋆⋆Department of Operations Research, Eötvös University, Pázmány Péter sétány 1/C, 1117 Budapest, Hungary. e-mail: `csabi@cs.elte.hu`

[1] In what follows we shall call a partition of the edge set of a graph into k classes a *k-labeling* and say that the graph is *k-edge-labeled*. Thus tensegrity graphs are 3-edge-labeled and cable-strut tensegrity graphs are 2-edge-labeled.

where $p(v_i) = p_i$ and $q(v_i) = q_i$ for all $1 \leq i \leq n$. 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. It is well-known that the infinitesimal rigidity of tensegrity frameworks is not a generic property: the same tensegrity graph may possess infinitesimally rigid as well as infinitesimally non-rigid generic realizations in any fixed dimension $d \geq 1$. Thus we may define two (different) families of tensegrity graphs in dimension d: we say that a tensegrity graph T is *rigid* in $\mathbb{R}^d$ if it has an infinitesimally rigid generic realization (T, p) in $\mathbb{R}^d$ and *strongly rigid* if all generic realizations (T, p) in $\mathbb{R}^d$ are infinitesimally rigid. We refer the reader to [1, 2, 6, 7] for more details on the rigidity of tensegrity frameworks.

It is not known whether the rigidity of tensegrity graphs can be tested in polynomial time 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 3-dimensional version remains one of the major open problems in combinatorial rigidity.

In this note we show that it is co-NP-complete to test whether a tensegrity graph is strongly rigid in $\mathbb{R}^1$.

2 Tensegrities on the line and alternating cycles

Let $G = (V; C, S)$ be a 2-edge-labeled graph. A cycle in G is *alternating* if no two incident edges along the cycle have the same label. Note that a pair of parallel edges with different labels form an alternating cycle. We say that G has the *alternating cycle property* if for all proper bipartitions $(U, V - U)$ of V there is an alternating cycle in the bipartite subgraph $H = (V, E(U, V - U)$ of G induced by the bipartition. For such a bipartition of G let D be the directed graph obtained from H by orienting the edges in C from U to $V - U$ and the edges in S from $V - U$ to U. It is easy to verify that a cycle of H is alternating if and only if the corresponding oriented cycle in D is a directed cycle.

Theorem 2.1. *Let $G = (V; C, S)$ be a cable-strut tensegrity graph. Then G is strongly rigid in $\mathbb{R}^1$ if and only if G, with the cable-strut labeling, has the alternating cycle property.*

Proof. We first suppose that G has a bipartition $(U, V - U)$ such that the graph $H = (V, E(U, V - U))$ contains no alternating cycle. Let D be the digraph obtained from H by orienting the edges in C from U to $V - U$ and the edges in S from $V - U$ to U. Then D has no directed cycles and hence there exists a topological ordering of the vertices of D, that is, an ordering $v_1, v_2, ..., v_n$ for which $i < j$ whenever $v_i v_j$ is an arc of D. Thus, for each edge $e = v_i v_j$ of H, we have: $v_i \in U$ and $v_j \in V - U$ if $e \in C$; $v_i \in V - U$ and $v_j \in U$ if $e \in S$. Let (G, p) be a realization of G on the line in which $p(v_i) < p(v_j)$ if $i < j$, for all $1 \leq i < j \leq n$. We may suppose that p is generic. Define infinitesimal velocities $q : V \to \mathbb{R}$ by letting $q(v) = 1$ if $v \in U$ and $q(v) = 0$ otherwise. Then q is a non-trivial infinitesimal motion of (G, p). Hence (G, p) is not infinitesimally rigid and G is not strongly rigid.

We next suppose that G has the alternating cycle property. Let (G, p) be a 1-dimensional (generic) realization of G as a tensegrity framework. Suppose that q is a non-trivial (i.e. non-constant) infinitesimal motion of (G, p). Let $U = \{v \in V : q(v) < 0\}$. By combining q with a suitable translation of $\mathbb{R}^1$, we may assume that $\emptyset \neq U \neq V$. Let $X = u_1v_1u_2v_2...u_tv_tu_1$ be an alternating cycle in G with $u_i \in U$ and $v_i \in V - U$ for $1 \leq i \leq t$, and $u_1v_1 \in C$. We must have $p(v_1) < p(u_1)$. The facts that X is alternating, $q(u_i) < 0$ and $q(v_i) \geq 0$ for all $1 \leq i \leq t$ imply that $p(u_{i+1}) < p(v_i) < p(u_i)$ and $p(v_{i+1}) < p(u_{i+1}) < p(v_i)$ for all $1 \leq i \leq t-1$. Thus $p(u_1), p(v_1), p(u_2), ..., p(v_t)$ is a strictly decreasing sequence. This contradicts the fact that we must have $p(u_1) < p(v_t)$ since $u_1v_t \in S$, $q(u_1) < 0$ and $q(v_t) = 0$. □

3 The hardness proof

In this section we prove the hardness result. First we show that the following problem is co-NP-complete.

ACP. Given a 2-edge-labeled graph $G = (V; C, S)$, decide whether G has the alternating cycle property.

We shall reduce the following satisfiability problem to ACP. It is well-known that 3-SAT is NP-complete, see for example [3].

3-SAT. Given a Boolean formula in conjunctive normal form, in which each clause contains exactly three literals, decide whether there is a truth assignment for the variables which makes the formula true.

Theorem 3.1. *ACP is co-NP-complete.*

Proof. It is easy to see that ACP is in co-NP: given a bipartition $(U, V - U)$ of V, one can verify that $H = (V, E(U, V - U))$ contains no alternating cycles by showing that the directed graph D (defined in the proof of Theorem 2.1) is acyclic. It will be convenient to call a bipartition $(U, V - U)$ with no alternating cycles *pure*.

Consider a formula φ, an instance of 3-SAT. Suppose that φ contains c clauses and n variables and let c_i and $\bar{c}_i$ denote the number of occurrences of variable x_i and $\bar{x}_i$ in φ, respectively, for $1 \leq i \leq n$. Construct a 2-edge-labeled graph $G = (V; C, S)$ as follows. Take $2n + 2$ vertex-disjoint paths $T, F, P_1, ..., P_n, \bar{P}_1, ..., \bar{P}_n$ such that the number of vertices on T and F are $2n + 1 + 3c$ and $2n + 1$, respectively, while P_i ($\bar{P}_i$) has $3 + c_i$ (resp. $3 + \bar{c}_i$) vertices for $1 \leq i \leq n$. The paths $P_i, \bar{P}_i$ correspond to the variables and the long paths T, F correspond to the true and false assignments, as we shall see later. All vertices of G lie on these paths and we put an edge in C and an edge in S for each edge of the paths. This ensures that none of these paths can cross a pure bipartition of G.

Next we describe the additional edges of G that connect these paths. We connect the first two vertices of each path $P_i, \bar{P}_i$, $1 \leq i \leq n$, to the first vertex of path T and F, respectively, and label them so that $\{v_{i,1}t_1, \bar{v}_{i,1}t_1, v_{i,2}f_1, \bar{v}_{i,2}f_1\} \subset C$ and $\{v_{i,2}t_1, \bar{v}_{i,2}t_1, v_{i,1}f_1, \bar{v}_{i,1}f_1\} \subset S$ hold. These edges ensure that T and F belong to different sides of a pure bipartition of G. We then connect the third vertex of each path $P_i, \bar{P}_i$, $1 \leq i \leq n$, to the vertices t_{2i}, t_{2i+1} and f_{2i}, f_{2i+1} of the path T and F, respectively, and label them so that $\{v_{i,3}t_{2i}, v_{i,3}f_{2i}, \bar{v}_{i,3}t_{2i+1}, \bar{v}_{i,3}f_{2i+1}\} \subset C$ and $\{v_{i,3}t_{2i+1}, v_{i,3}f_{2i+1}, \bar{v}_{i,3}t_{2i}, \bar{v}_{i,3}f_{2i}\} \subset S$ hold. These edges ensure that in a pure bipartition of G exactly one of the paths P_i and $\bar{P}_i$ will belong to the same side as path T, for all $1 \leq i \leq n$.

Finally, we add an alternating cycle of length six corresponding to each clause of φ. For example, for the clause $(x_2 \vee x_5 \vee \bar{x_8})$ we add a cycle as follows: we take the next three unused vertices on T, say a, b, c, and the next unused vertex on each of the paths $P_2, P_5, \bar{P}_8$, call them x, y, z, respectively. We add the the edges ax, by, cz, label them C, and add az, bx, cy, and label them S. This ensures that in a pure bipartition at least one of $P_2, P_5, \bar{P}_8$ must belong to the same side as path T.

The construction of G and the arguments above imply that there is a truth assignment for the variables which makes formula φ true if and only if G has a pure bipartition, i.e. G does not have the alternating cycle property. Therefore ACP is co-NP-hard. □

It follows that it is also hard to test whether a tensegrity graph is strongly rigid on the line, even if it has no bars. (Note that by replacing each bar by a cable and a parallel strut, we may always suppose that B is empty.) The decision problem is as follows.

STRONGLY RIGID TENSEGRITY. Given a cable-strut tensegrity graph $G = (V; C, S)$, decide whether G is strongly rigid in $\mathbb{R}^1$.

Theorem 3.2. *STRONGLY RIGID TENSEGRITY is co-NP-complete.*

Proof. Directly from Theorems 2.1 and 3.1. □

4 Concluding remarks

The proof of Theorem 2.1 shows that if we replace 'generic' by 'general position' in the definition of strong rigidity in $\mathbb{R}^1$ then the family of strongly regular graphs will reamin the same. This is not true when $d \geq 2$, even for bar graphs.

Let $G = (V; C, S)$ be a 2-edge-labeled graph and let B be a set of bars corresponding to the alternating 2-cycles of G. Theorem 2.1 implies that if G is strongly rigid in $\mathbb{R}^1$ then each vertex of G is incident with an edge of B. Let c be the number of components of the graph (V, B). Theorem 2.1 gives a polynomial algorithm for checking if G is strongly rigid when c is fixed since we need only check 2^c bipartite graphs for alternating cycles (which can be done by checking if the associated digraph is acyclic).

We can use Theorem 2.1 to construct graphs which are not strongly rigid in $\mathbb{R}^1$ and for which the connectivity of both (V, C) and (V, S) is arbitrarily large. For example let (V, B) consist of two large complete graphs and join them by $2k$ independent edges, half in C and half in S.

A natural open question is whether our hardness result concerning strong rigidity extends to higher dimensions $d \geq 2$.

Another question is whether the corresponding labeling problem is hard: given a graph G, decide whether the edges set of G can be 2-edge-labeled so that the resulting cable-strut tensegrity graph is strongly rigid in $\mathbb{R}^1$ (and find a good labeling, if it exists). If we replace strongly rigid by rigid, we obtain a labeling problem which is polynomially solvable in $\mathbb{R}^d$ for $d = 1, 2$, see [4].

5 Acknowledgements

This work was supported by the MTA-ELTE Egerváry Research Group on Combinatorial Optimization, the Hungarian Scientific Research Fund grant no. K81472, and the National Development Agency of Hungary, grant no. CK 80124, based on a source from the Research and Technology Innovation Fund.

References

[1] R. Connelly, Rigidity, *in:* P.M. Gruber and J.M. Wills, *(eds) Handbook of convex geometry*, Elsevier, 1993, pp. 223-271.

[2] R. Connelly and W. Whiteley, Second-order rigidity and prestress stability for tensegrity frameworks, *SIAM J. Disc. Math.*, Vol. 9, Issue 3, 453-491, 1996.

[3] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979.

[4] T. Jordán, A. Recski, and Z. Szabadka, Rigid tensegrity labelings of graphs, *European J. Combinatorics* 30 (2009) 1887-1895.

[5] A. Recski and O. Shai, One-dimensional synthesis of graphs as tensegrity frameworks, *4th Japanese-Hungarian Symposium on Discrete Mathematics and its Applications*, Budapest, 2005, 284-288. (See also `http://www.cs.bme.hu/ recski/`.)

[6] B. Roth and W. Whiteley, Tensegrity frameworks, *Trans. Amer. Math. Soc.*, Vol. 265 (1981), Number 2, 419-446.

[7] W. Whiteley, Rigidity and scene analysis, *in* J. E. Goodman and J. O'Rourke, *(eds.): Handbook of Discrete and Computational Geometry,* 2nd ed., CRC Press, Boca Raton, 2004, pp. 1327-1354.