A Semidefinite Programming Approach to Tensegrity Theory and Realizability of Graphs

Anthony Man–Cho So*, Yinyu Ye†

Abstract

Recently, Connelly and Sloughter [14] have introduced the notion of *d–realizability* of graphs and have, among other things, given a complete characterization of the class of 3–realizable graphs. However, their work has left open the question of finding an algorithm for realizing those graphs. In this paper, we resolve that question by showing that the semidefinite programming (SDP) approach of [11, 33] can be used for realizing 3–realizable graphs. Specifically, we use SDP duality theory to show that given a graph G and a set of lengths on its edges, the optimal dual multipliers of a certain SDP give rise to a *proper equilibrium stress* for some realization of G. Using this result and the techniques in [14, 32], we then obtain a polynomial time algorithm for (approximately) realizing 3–realizable graphs. Our results also establish a little–explored connection between SDP and tensegrity theories and allow us to derive some interesting properties of tensegrity frameworks.

1 Introduction

Due to its fundamental nature and versatile modelling power, the *Graph Realization* problem has received a lot of attention lately. In that problem, one is given a graph $G = (V, E)$ and a set of non–negative edge weights $\{d_{ij} : (i, j) \in E\}$, and the goal is to compute a realization of G in the Euclidean space $\mathcal{R}^d$ for a given dimension d, i.e. to place the vertices of G in $\mathcal{R}^d$ such that the Euclidean distance between every pair of adjacent vertices v_i, v_j equals to the prescribed weight d_{ij}. The Graph Realization problem and its variants arise from applications in various areas, the two most prominent of which being molecular conformation and wireless sensor network localization. In the former, the vertices of G correspond to atoms, the edges correspond to chemical bonds, and the weights correspond to inter–atomic distances. In the latter, the vertices of G correspond to sensors, the edges correspond to communication links, and the weights correspond to distances between sensors. Furthermore, the vertices are partitioned into two sets – one is the *anchors*, whose exact positions are known; and the other is the *sensors*, whose positions are unknown. In these contexts, an algorithm for finding a realization of the vertices in the required dimension will have interesting biological and engineering consequences.

It turns out that very early on, Schoenberg [30] and Young and Householder [35] have studied the case where all pairwise distances are given (i.e. the underlying graph is complete). They have shown that the given pairwise distances arise from points in $\mathcal{R}^d$ (but not $\mathcal{R}^{d-1}$) iff a certain matrix is positive semidefinite and has rank d. Such a characterization forms the basis of various multidimensional scaling algorithms (see, e.g., [15]). More recently, the problem is being tackled in two directions: one is the development of fast and robust heuristics (see, e.g., [16, 19, 21, 25, 31]), and the other is a more theoretical study of the problem. Regarding the latter, Barvinok [9] and Alfakih and Wolkowicz [1, 2] (also see [17]) have used SDP theory to show that the problem is solvable in polynomial time if the dimension of the realization is not restricted. Moreover, they have given bounds on the dimension needed to realize the given distances. However, if we require the realization to be in $\mathcal{R}^d$ for some fixed d, then the problem becomes NP–complete [7, 8, 29]. Thus, a natural problem is to identify families of graphs that admit polynomial time algorithms for computing a realization in the required dimension. For instance, in [33], the authors have introduced the notion of *unique localizability* and shown that a certain SDP formulation can be used to find a realization of a uniquely localizable graph in the required dimension. Their model requires the presence of *anchors*, i.e. vertices that are pinned to fixed positions, but this requirement is later eliminated and an alternative notion is being developed in Biswas et. al. [10]. On another front, Connelly and Sloughter [14] have introduced the notion of d–realizability and given a complete characterization of the class of d–realizable graphs, where $d = 1, 2, 3$. Informally, a

*Department of Computer Science, Stanford University, Stanford, CA 94305. E–mail: manchoso@cs.stanford.edu

†Department of Management Science and Engineering and, by courtesy, Electrical Engineering, Stanford University, Stanford, CA 94305. E–mail: yinyu-ye@stanford.edu

graph G is d–realizable if, given any set of edge lengths, whenever G is realizable at all, it can also be realized in $\mathcal{R}^d$. Their proof employs elements of tensegrity theory (in particular the notion of stress) and at first sight seems to have nothing to do with the approach in [33]. As far as algorithmics are concerned, it is trivial to find a realization of an 1–realizable graph in $\mathcal{R}$, since a graph is 1–realizable iff it is a forest [14]. It is also mentioned in [14] that a polynomial time algorithm for realizing 2–realizable graphs exists. However, finding a corresponding algorithm for 3–realizable graphs is posed as an open question.

1.1 Our Contribution In this paper, we resolve the above question by giving a polynomial time algorithm for (approximately) realizing 3–realizable graphs. To explain our approach, it would be helpful to first review the high–level ideas in the characterization of 3–realizable graphs [14, 32]. The main bottleneck in the proof is to show that two graphs, V_8 and $C_5 \times C_2$, are 3–realizable (see Figures 1(c) and 1(d) in Section 5). In order to achieve that, one first argues that there exists a realization $\mathbf{p}$ of $H \in \{V_8, C_5 \times C_2\}$ (not necessarily in $\mathcal{R}^3$) such that the distance between a certain pair of non–adjacent vertices (i,j) is maximized. Then, using the Inverse Function Theorem, it can be shown that such a realization induces a *non–zero* equilibrium force on the graph H' obtained from H by adding the edge (i,j). One can then use this equilibrium force to argue that H can be realized in $\mathcal{R}^3$. The obstacle to obtaining an algorithm from the above arguments lies of course in the computation of $\mathbf{p}$, and one of our main results is to show that this can be overcome. Specifically, using the ideas introduced in [33], we show that the problem of computing the desired $\mathbf{p}$ can be formulated as an SDP. What is even more interesting is that the optimal dual multipliers of our SDP give rise to a non–zero equilibrium force. In fact, we have established this result under a more general setting. Consequently, it allows us to take a more algorithmic approach to tensegrity theory (previous results in that area usually assume that a starting configuration $\mathbf{p}$ is given) and enables us to derive some interesting properties of tensegrity frameworks. Last but not least, it allows us to apply the techniques developed in [14, 32] to obtain an algorithm for realizing 3–realizable graphs.

1.2 Outline of the Paper The rest of the paper is organized as follows. In Section 2, we define the notions of *d–realizability*, *equilibrium stress* and *unyielding tensegrity* and introduce the notations that will be used in the paper. In Section 3, we formulate the problem of finding an unyielding tensegrity as an SDP and discuss some of its properties. In Section 4 we analyze the SDP and show how the optimal dual multipliers give rise to a non–zero equilibrium stress. In addition, we discuss how our results are related to those in tensegrity theory. We then show how to use these results and the techniques in [14, 32] to develop an algorithm for realizing 3–realizable graphs in Section 5. Lastly, we summarize our results in Section 6 and discuss some possible future directions.

2 Preliminaries

We begin with some notations and definitions that will be used throughout the paper. The inner product of two matrices P and Q is denoted by $P \bullet Q \equiv \text{Trace}(P^T Q)$. Now, let $G = (V, E)$ be a connected graph that contains neither loops nor multiple edges, and let $\mathbf{d} = \{d_{ij} \geq 0 : (i,j) \in E\}$ be a set of non–negative weights on the edges of G. We shall refer to d_{ij} as the *length* of the edge (i,j). A *realization* $\mathbf{p} = (p_i)_{i \in V}$ of G with respect to $\mathbf{d}$ is a function that assigns to each vertex i a point p_i in some Euclidean space, such that $\|p_i - p_j\| = d_{ij}$ for all $(i,j) \in E$. We say that G is *d–realizable* if the following holds: given any set of lengths $\mathbf{d}$ on the edges of G, if there exists a realization $\mathbf{p}$ of G with respect to $\mathbf{d}$ in $\mathcal{R}^{d'}$ for some d', then there exists a realization $\mathbf{q}$ of G with respect to $\mathbf{d}$ in $\mathcal{R}^d$. The notion of d–realizability is introduced by Connelly and Sloughter [14], and in this paper we are interested in its algorithmic aspects. Towards that end, we need some definitions from tensegrity theory (see, e.g., [13, 28]).

Definition 2.1. *A tensegrity $G(\mathbf{p})$ is a graph $G = (V, E)$ together with a configuration $\mathbf{p} = (p_i) \in \mathcal{R}^D \times \cdots \times \mathcal{R}^D = \mathcal{R}^{|V|D}$ such that each edge is labelled as a cable, strut, or bar, and each vertex is labelled as pinned or unpinned. $G(\mathbf{p})$ is the realization of G in $\mathcal{R}^D$ obtained by locating vertex i at point $p_i \in \mathcal{R}^D$.*

The label on each edge is intended to indicate its functionality. Cables (resp. struts) are allowed to decrease (resp. increase) in length (or stay the same length), but not to increase (resp. decrease) in length. Bars are forced to remain the same length. Regarding the vertices, a pinned vertex is forced to remain where it is. Given a graph $G = (V, E)$ and a set of lengths $\mathbf{d}$, if (i,j) is a cable (resp. strut), then d_{ij} will be the upper (resp. lower) bound on its length. If (i,j) is a bar, then d_{ij} will simply be its length.

An important concept in the study of tensegrities is that of an equilibrium stress.

Definition 2.2. *An equilibrium stress for $G(\mathbf{p})$ is an assignment of real numbers $\omega_{ij} = \omega_{ji}$ to each edge $(i,j) \in E$ such that for each unpinned vertex i of G,* *we have $\sum_{j:(i,j)\in E} \omega_{ij}(p_i - p_j) = \mathbf{0}$. Furthermore, we say that the equilibrium stress $\omega = \{\omega_{ij}\}$ is proper if $\omega_{ij} = \omega_{ji} \geq 0$ (resp. ≤ 0) if (i,j) is a cable (resp. strut).*

Clearly, the zero stress $\omega = \mathbf{0}$ is a proper equilibrium stress, but it is not too interesting. Thus, it would be nice to have conditions that guarantee the existence of a non–zero proper equilibrium stress. It turns out that the concept of an unyielding tensegrity is useful for that purpose.

Definition 2.3. *Let $G = (V, E)$ be a graph, and let $\mathbf{p}$ and $\mathbf{q}$ be two configurations of G. We say that $G(\mathbf{p}) \succeq G(\mathbf{q})$ if for every pinned vertex i, we have $p_i = q_i$, and for every edge $(i,j) \in E$, we have:*

$$|p_i - p_j| \left\{ \begin{array}{c} \geq \\ = \\ \leq \end{array} \right\} |q_i - q_j| \quad \textit{if } (i,j) \textit{ is a } \left\{ \begin{array}{c} \textit{cable} \\ \textit{bar} \\ \textit{strut} \end{array} \right\}$$

We say that $G(\mathbf{p})$ is unyielding if any other configuration $\mathbf{q}$ with $G(\mathbf{p}) \succeq G(\mathbf{q})$ has the same edge lengths as $\mathbf{p}$.

As shown in [32], an unyielding tensegrity with at least one cable or strut has a non–zero proper equilibrium stress. This result is crucial in the characterization of 3–realizable graphs [14, 32], as it allows one to argue that a low–degree vertex and its neighbors can only span a low dimensional subspace. As we shall see, given a graph $G = (V, E)$ with at least one cable or strut and a set of lengths $\mathbf{d}$ on the bars (i.e. there are no bounds on the lengths of the cables and struts), one can efficiently compute (or more precisely, approximate to arbitrary accuracy) a realization $\mathbf{p}$ of G such that the tensegrity $G(\mathbf{p})$ is unyielding. Moreover, the associated non–zero proper equilibrium stress will arise as a by–product of the computation.

3 A Semidefinite Programming Formulation

Let $G = (V, E)$ and $\mathbf{d}$ be as above, where all edges are bars. Consider a partition of the vertices into two *non–empty* sets: the set $A = \{a_1, \ldots, a_m\}$ of pinned vertices, and the set $S = \{x_1, \ldots, x_n\}$ of unpinned vertices. We assume that the coordinates of all the $a_i \in \mathcal{R}^d$ are given, and that $a_1 = \mathbf{0}$. Let $\bar{d}_{kj}$ be the length between a_k and x_j, and let d_{ij} be the length between x_i and x_j. Let $N_a = \{(k,j) : (a_k, x_j) \in E\}$ and $N_x = \{(i,j) : i < j,\ (x_i, x_j) \in E\}$. Clearly, we have $E = N_a \cup N_x$. We set $N_a^c = \{(k,j) : (a_k, x_j) \notin E\}$, and define N_x^c analogously. Given a graph G and a set of lengths on its edges, the relation $\succeq$ induces a partial order on the set of tensegrities $G(\mathbf{p})$, where $\mathbf{p}$ is a configuration that satisfies the edge constraints. In particular, an unyielding tensegrity can be viewed as an extremal element in this partial order. This observation motivates us to formulate the problem of finding an unyielding tensegrity as an optimization problem. Specifically, let C_1, S_1 be disjoint subsets of N_a^c, and let C_2, S_2 be disjoint subsets of N_x^c. Consider the following semidefinite program, where Z is the optimizing variable (we refer the reader to [3, 20, 34] for surveys on semidefinite programming and to [33] for a motivation of the following formulation):

(3.1)
$$\begin{array}{lll} \sup & \sum_{(k,j)\in S_1} (a_k; e_j)(a_k; e_j)^T \bullet Z & \\ & + \sum_{(i,j)\in S_2} (\mathbf{0}; e_{ij})(\mathbf{0}; e_{ij})^T \bullet Z & \\ & - \sum_{(k,j)\in C_1} (a_k; e_j)(a_k; e_j)^T \bullet Z & \\ & - \sum_{(i,j)\in C_2} (\mathbf{0}; e_{ij})(\mathbf{0}; e_{ij})^T \bullet Z & \\ \text{s.t.} & Z_{1:d,1:d} = I_d & \\ & (\mathbf{0}; e_{ij})(\mathbf{0}; e_{ij})^T \bullet Z = d_{ij}^2 & \forall (i,j) \in N_x \\ & (a_k; e_j)(a_k; e_j)^T \bullet Z = \bar{d}_{kj}^2 & \forall (k,j) \in N_a \\ & Z \succeq \mathbf{0} & \end{array}$$

Here, $e_{ij} \in \mathcal{R}^n$ is the vector with 1 at the i–th position, -1 at the j–th position and zero elsewhere; $e_j \in \mathcal{R}^n$ is the vector of all zeros except an -1 at the j–th position; $(a_k; e_j) \in \mathcal{R}^{d+n}$ is the vector of a_k on top of e_j; and I_d is the d–dimensional identity matrix. To gain some intuition on (3.1), suppose that $x_1, \ldots, x_n \in \mathcal{R}^{d'}$ ($d' \geq d$) are such that $\|x_i - x_j\|^2 = d_{ij}^2$ for all $(i,j) \in N_x$, and that $\|x_j - (a_k; \mathbf{0})\|^2 = \bar{d}_{kj}^2$ for all $(k,j) \in N_a$ (i.e. we augment $a_k \in \mathcal{R}^d$ to $(a_k; \mathbf{0}) \in \mathcal{R}^{d'}$). Let $x_i = (x_{i1}, x_{i2}, \ldots, x_{id'}) \in \mathcal{R}^{d'}$ for $i = 1, 2, \ldots, n$. Consider the matrices:

$$\begin{array}{rcl} X & = & \left[\begin{array}{cccc} x_{11} & x_{21} & \cdots & x_{n1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1d} & x_{2d} & \cdots & x_{nd} \end{array} \right] \\ X' & = & \left[\begin{array}{cccc} x_{1,d+1} & x_{2,d+1} & \cdots & x_{n,d+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1d'} & x_{2d'} & \cdots & x_{nd'} \end{array} \right] \\ Y & = & X^T X + (X')^T X' \end{array}$$

Then, it is straightforward to verify that the matrix:

(3.2)
$$Z = \left[\begin{array}{cc} I_d & X \\ X^T & Y \end{array} \right]$$

is a feasible solution to (3.1). Conversely, suppose that a matrix Z of the form (3.2) is feasible for (3.1). Then, we have $Y - X^T X \succeq \mathbf{0}$, and hence there exists an $X' \in \mathcal{R}^{r \times n}$ such that $Y - X^T X = (X')^T X'$, where $r = \text{rank}(Y - X^T X)$. Now, let $X = [x_1, \ldots, x_n]$ and $X' = [x_1', \ldots, x_n']$. Since $N_a \neq \emptyset$, the arguments in [33] show that the points $\tilde{x}_j = (x_j; x_j') \in \mathcal{R}^{d+r}$ satisfy $\|\tilde{x}_j - \tilde{x}_i\|^2 = d_{ij}^2$ for all $(i,j) \in N_x$, and $\|\tilde{x}_j - (a_k; \mathbf{0})\|^2 = \bar{d}_{kj}^2$ for all $(k,j) \in N_a$. Thus, we may interpret problem (3.1) as follows. In essence, we are placing cables between pairs of vertices in $C_1 \cup C_2$ and struts between pairs of vertices in $S_1 \cup S_2$, and problem (3.1) attempts to find a realization $\mathbf{p}$ such that the tensegrity $G(\mathbf{p})$ is unyielding. Note that we do not assume the sets C_1, C_2, S_1, S_2 are non–empty.

It turns out that it is very instructive to consider the dual of (3.1), which is given by:

(3.3)
$$\begin{array}{rl} \inf & I_d \bullet V + \sum_{(i,j)\in N_x} \theta_{ij} d_{ij}^2 + \sum_{(k,j)\in N_a} w_{kj} \bar{d}_{kj}^2 \\ \text{s.t.} & U \equiv -\sum_{(k,j)\in S_1} (a_k; e_j)(a_k; e_j)^T \\ & \quad - \sum_{(i,j)\in S_2} (\mathbf{0}; e_{ij})(\mathbf{0}; e_{ij})^T \\ & \quad + \sum_{(k,j)\in C_1} (a_k; e_j)(a_k; e_j)^T \\ & \quad + \sum_{(i,j)\in C_2} (\mathbf{0}; e_{ij})(\mathbf{0}; e_{ij})^T \\ & \quad + \left[\begin{array}{cc} V & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{array} \right] + \sum_{(i,j)\in N_x} \theta_{ij} (\mathbf{0}; e_{ij})(\mathbf{0}; e_{ij})^T \\ & \quad + \sum_{(k,j)\in N_a} w_{kj} (a_k; e_j)(a_k; e_j)^T \succeq \mathbf{0} \end{array}$$

As we shall see, the optimal dual multipliers are closely related to an equilibrium stress for a certain realization of G.

4 Analysis of the SDP Formulation

4.1 Dual Multipliers and Equilibrium Stress

The following fact is established in [32] and is crucial to the proofs in [14, 32]:

Fact 4.1. *An unyielding tensegrity with at least one cable or strut has a non–zero proper equilibrium stress.*

Unfortunately, the proof involves the use of the Inverse Function Theorem and hence is not constructive. To obtain a constructive version of Fact 4.1, we proceed as follows. Let G be a graph with at least one cable or strut and has at least one pinned vertex. Let $\mathbf{d}$ be the set of lengths on the bars of G. We then formulate the SDP (3.1) for G, where $C_1 \cup C_2$ contains all the cables and $S_1 \cup S_2$ contains all the struts in G. Suppose that problem (3.1) is strictly feasible, i.e. there exists a primal feasible Z with $Z \succ \mathbf{0}$, and that the graph $G \backslash \{a_2, \ldots, a_m\}$ is connected. Under these assumptions, we can establish the following:

Theorem 4.1. *Let $\tilde{X} = [\tilde{x}_1, \ldots, \tilde{x}_n]$ be the positions of the unpinned vertices obtained from the optimal primal matrix $\bar{Z}$ (cf. Section 3), and let $\{\bar{\theta}_{ij}, \bar{w}_{kj}\}$ be the optimal dual multipliers. Suppose that we assign the stress $\bar{\theta}_{ij}$ (resp. $\bar{w}_{kj}$) to the bar $(i,j) \in N_x$ (resp. $(k,j) \in N_a$), a stress of 1 to all the cables, and a stress of -1 to all the struts. Then, the resulting assignment yields a non–zero proper equilibrium stress for the realization $\{(a_1; \mathbf{0}), \ldots, (a_m; \mathbf{0}), \tilde{x}_1, \ldots, \tilde{x}_n\}$.*

Before we outline the proof of Theorem 4.1, several remarks are in order. First, the intuition behind Theorem 4.1 is simple. Suppose that (3.1) and (3.3) achieve the same optimal value, and that the common optimal value is attained by the primal matrix $\bar{Z}$ and the dual matrix $\bar{U}$. Then, the desired result should follow from one of the conditions for strong duality, namely the identity $\bar{Z}\bar{U} = \mathbf{0}$. Of course, strong duality for SDP does not necessarily hold, and even when it does, there is no guarantee that the optimal value is attained by any matrix (see [20] or the appendix for some examples). Thus, some additional technical assumptions are needed, and as we shall see below, those that appear in the statement of Theorem 4.1 are sufficient for our purposes. We remark that in our applications of Theorem 4.1, we will have only one pinned vertex, namely $a_1 = \mathbf{0}$. Thus, primal strict feasibility can be ensured if the given lengths $\mathbf{d}$ admit a realization whose vertices are in general position, and the connectivity condition is simply the statement that G is connected. Secondly, we have assumed that the graph G has at least one pinned vertex, while there is no such restriction for Fact 4.1 to hold. Such an assumption is necessary in order to ensure that the entries of $\bar{Z}$ are bounded and to guarantee that the points $\tilde{x}_1, \ldots, \tilde{x}_n$ satisfy the length constraints (see [33]). On the other hand, we can no longer argue that the net stress exerted on a pinned vertex is zero. However, if there is only one pinned vertex in G, then the net stress exerted on it will be zero (see Corollary 4.1 and the remark after it), and thus the conclusion of Theorem 4.1 will coincide with that of Fact 4.1 applied to the case where none of the vertices in G are pinned. Moreover, such an assumption buys us some more information that is not offered by Fact 4.1. Specifically, the equilibrium stress obtained in Theorem 4.1 is non–zero on all the cables and struts, and the magnitudes of the stress on all the cables and struts are identical. To the best of our knowledge, this is the first time such a property is observed.

The following proposition guarantees that the gap between the optimal values of (3.1) and (3.3) is zero, i.e. strong duality holds.

Proposition 4.1. *Suppose that $a_1 = \mathbf{0}$, and that the* *graph $G\backslash\{a_2,\dots,a_m\}$ is connected. Then, the dual problem (3.3) is strictly feasible, i.e. there exists a dual feasible U with $U \succ \mathbf{0}$.*

Proof. It suffices to show that there exists a choice of V, θ_{ij} and w_{kj} such that:

$$\begin{aligned} U' \quad &\equiv \quad \begin{bmatrix} V & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \sum_{(i,j)\in N_x} \theta_{ij}(\mathbf{0};e_{ij})(\mathbf{0};e_{ij})^T \\ &\quad + \sum_{(k,j)\in N_a} w_{kj}(a_k;e_j)(a_k;e_j)^T \\ &\quad - \sum_{(k,j)\in S_1} (a_k;e_j)(a_k;e_j)^T \\ &\quad - \sum_{(i,j)\in S_2} (\mathbf{0};e_{ij})(\mathbf{0};e_{ij})^T \succ \mathbf{0} \end{aligned}$$

Note that:

$$U' = \begin{bmatrix} U'_{11} & U'_{12} \\ (U'_{12})^T & U'_{22} \end{bmatrix}$$

where:

$$\begin{aligned} U'_{22} \quad = \quad & \sum_{(i,j)\in N_x} \theta_{ij} e_{ij} e_{ij}^T + \sum_{(k,j)\in N_a} w_{kj} e_j e_j^T \\ & - \sum_{(k,j)\in S_1} e_j e_j^T - \sum_{(i,j)\in S_2} e_{ij} e_{ij}^T \end{aligned}$$

Let us choose $\theta_{ij} = \mu_{ij} > 0$ for all $(i,j) \in N_x$, $w_{1s} = \nu_s > 0$ for all s such that $(1,s) \in N_a$, and $w_{kj} = 0$ for all other $(k,j) \in N_a$. Using the connectivity hypothesis, one can show that:

$$M \equiv \sum_{(i,j)\in N_x} \mu_{ij} e_{ij} e_{ij}^T + \sum_{s:(1,s)\in N_a} \nu_s e_s e_s^T \succ \mathbf{0}$$

Now, let $u \in \mathcal{R}^n\backslash\{\mathbf{0}\}$. Observe that:

$$\text{(4.4)} \quad u^T \left(\sum_{(k,j)\in S_1} e_j e_j^T \right) u = \sum_{(k,j)\in S_1} u_j^2 \le m\|u\|^2$$

(4.5)

$$u^T \left(\sum_{(i,j)\in S_2} e_{ij} e_{ij}^T \right) u = \sum_{(i,j)\in S_2} (u_i - u_j)^2 \le 4n\|u\|^2$$

Let $v^* = \min_{\|u\|=1} u^T M u$. Clearly, we have $v^* > 0$. Let $\alpha > 0$ be such that $\alpha v^* > 4n + m$. Then, it follows from (4.4) and (4.5) that for $u \in \mathcal{R}^n\backslash\{\mathbf{0}\}$ with $\|u\| = 1$, we have:

$$u^T \left(\alpha M - \sum_{(k,j)\in S_1} e_j e_j^T - \sum_{(i,j)\in S_2} e_{ij} e_{ij}^T \right) u > 0$$

In particular, we see that by taking $\theta_{ij} = \alpha\mu_{ij}$ and $w_{1s} = \alpha\nu_s$, we have $U'_{22} \succ \mathbf{0}$.

To complete the proof of Proposition 4.1, let $V = I_d + \sum_{(k,j)\in S_1} a_k a_k^T$. Since $a_1 = \mathbf{0}$ by assumption, we have $U'_{11} = I_d$, and $U'_{12} = -\sum_{(k,j)\in S_1} a_k e_j^T$. By a theorem in linear algebra (cf. Theorem 7.7.6 of [23]), we have $U' \succ \mathbf{0}$ iff $U'_{22} - (U'_{12})^T U'_{12} \succ \mathbf{0}$. Now, note that $(U'_{12})^T U'_{12}$ does not depend on θ_{ij} and w_{kj}. Let $w^* = \max_{\|u\|=1} u^T (U'_{12})^T U'_{12} u$. Since $\{u : \|u\| = 1\}$ is compact, we see that $w^* < \infty$. Thus, by taking a sufficiently large α (i.e. $\alpha v^* - 4(n+m) - w^* > 0$), we see that:

$$u^T U'_{22} u - u^T (U'_{12})^T U'_{12} u \ge \alpha v^* - 4(n+m) - w^* > 0$$

for all $u \in \mathcal{R}^n\backslash\{\mathbf{0}\}$ with $\|u\| = 1$. In particular, we have $U'_{22} - (U'_{12})^T U'_{12} \succ \mathbf{0}$ as desired.

Proof of Theorem 4.1. By Proposition 4.1 and our assumptions, both (3.1) and (3.3) are strictly feasible. Thus, it follows from the duality theory for SDP (see, e.g., [20]) that there is no gap between the optimal values of (3.1) and (3.3), and that the common optimal value is attained in both problems. Let $\bar{Z}$ (resp. $\bar{U}$) be the optimal primal (resp. dual) matrix. Then, the absence of a duality gap implies that $\bar{Z}\bar{U} = \mathbf{0}$. Since $\bar{Z}$ takes the form (3.2), we obtain:

(4.6)

$$\begin{aligned} \bar{Z}\bar{U} \quad = \quad & \begin{bmatrix} V & \mathbf{0} \\ X^T V & \mathbf{0} \end{bmatrix} + \sum_{(i,j)\in N_x} \theta_{ij} \begin{bmatrix} \mathbf{0} & X_{ij} \\ \mathbf{0} & Y_{ij} \end{bmatrix} \\ & + \sum_{(k,j)\in N_a} w_{kj} \begin{bmatrix} * & \bar{X}_{kj} \\ * & \bar{Y}_{kj} \end{bmatrix} \\ & - \sum_{(k,j)\in S_1} \begin{bmatrix} * & \bar{X}_{kj} \\ * & \bar{Y}_{kj} \end{bmatrix} - \sum_{(i,j)\in S_2} \begin{bmatrix} \mathbf{0} & X_{ij} \\ \mathbf{0} & Y_{ij} \end{bmatrix} \\ & + \sum_{(k,j)\in C_1} \begin{bmatrix} * & \bar{X}_{kj} \\ * & \bar{Y}_{kj} \end{bmatrix} + \sum_{(i,j)\in C_2} \begin{bmatrix} \mathbf{0} & X_{ij} \\ \mathbf{0} & Y_{ij} \end{bmatrix} \\ = \quad & \mathbf{0} \end{aligned}$$

where:

(a) X_{ij} is an $d \times n$ matrix with $x_i - x_j$ on the i–th column and $-(x_i - x_j)$ on the j–th column;

(b) Y_{ij} is an $n \times n$ matrix with $(y_{1i} - y_{1j}, \dots, y_{ni} - y_{nj})$ on the i–the column and $-(y_{1i} - y_{1j}, \dots, y_{ni} - y_{nj})$ on the j–th column;

(c) $\bar{X}_{kj}$ is an $d \times n$ matrix with $x_j - a_k$ on the j–th column;

(d) $\bar{Y}_{kj}$ is an $n \times n$ matrix with $(y_{1j} - x_1^T a_k, \dots, y_{nj} - x_n^T a_k)$ on the j–th column.

Now, equation (4.6) implies that:

$$\begin{aligned} \mathbf{0} &= \sum_{(i,j)\in N_x} \theta_{ij} X_{ij} + \sum_{(k,j)\in N_a} w_{kj}\bar{X}_{kj} \\ &\quad - \sum_{(k,j)\in S_1}\bar{X}_{kj} - \sum_{(i,j)\in S_2} X_{ij} \\ &\quad + \sum_{(k,j)\in C_1}\bar{X}_{kj} + \sum_{(i,j)\in C_2} X_{ij} \end{aligned}$$

Consider a fixed column s of the matrix on the right–hand side. We have:

$$\begin{aligned} (4.7) \quad & \sum_{j:(j,s)\in N_x} \theta_{js}(x_s - x_j) + \sum_{k:(k,s)\in N_a} w_{ks}(x_s - a_k) \\ & - \sum_{k:(k,s)\in S_1} (x_s - a_k) - \sum_{j:(j,s)\in S_2} (x_s - x_j) \\ & + \sum_{k:(k,s)\in C_1} (x_s - a_k) + \sum_{j:(j,s)\in C_2} (x_s - x_j) = \mathbf{0} \end{aligned}$$

Thus, if $\mathrm{rank}(\bar{Z}) = d$, then we have just shown that the optimal dual multipliers $\{\theta_{ij}, w_{kj}\}$ yield a non–zero proper equilibrium stress for $\{a_1, \dots, a_m, x_1, \dots, x_n\}$. Note, however, that there are no restrictions placed on the pinned vertices $a_1, \dots, a_m$.

Now, suppose that $\mathrm{rank}(\bar{Z}) > d$. Then, following the arguments in [33], we have $Y - X^T X = (X')^T X'$, where $Y - X^T X$ has rank r. Now, consider the set $\tilde{x}_j = (x_j; x'_j) \in \mathcal{R}^{d+r}$, where $j = 1, 2, \dots, n$. Clearly, we have $r \le n$. Also, we have:

$$\|\tilde{x}_j\|^2 = Y_{jj} \quad \text{and} \quad \tilde{x}_i^T \tilde{x}_j = Y_{ij}$$

Using equation (4.6), we have:

$$\begin{aligned} \mathbf{0} &= \sum_{(i,j)\in N_x} \theta_{ij} Y_{ij} + \sum_{(k,j)\in N_a} w_{kj}\bar{Y}_{kj} \\ &\quad - \sum_{(k,j)\in S_1}\bar{Y}_{kj} - \sum_{(i,j)\in S_2} Y_{ij} \\ &\quad + \sum_{(k,j)\in C_1}\bar{Y}_{kj} + \sum_{(i,j)\in C_2} Y_{ij} \end{aligned}$$

Consider a fixed column s of the matrix on the right–hand side. Using the identities:

$$\begin{aligned} y_{ts} - y_{tj} &= x_t^T x_s + (x'_t)^T x'_s - \left(x_t^T x_j + (x'_t)^T x'_j \right) \\ &= x_t^T (x_s - x_j) + (x'_t)^T \left(x'_s - x'_j \right) \\ y_{ts} - x_t^T a_k &= x_t^T (x_s - a_k) + (x'_t)^T x'_s \end{aligned}$$

we conclude that for each $t = 1, 2, \dots, n$, we have:

(4.8)

$$\begin{aligned} (x'_t)^T \Bigg\{ & \sum_{j:(j,s)\in N_x} \theta_{js} \left(x'_s - x'_j \right) + \sum_{k:(k,s)\in N_a} w_{ks} x'_s \\ & - \sum_{k:(k,s)\in S_1} x'_s - \sum_{j:(j,s)\in S_2} \left(x'_s - x'_j \right) \\ & + \sum_{k:(k,s)\in C_1} x'_s + \sum_{j:(j,s)\in C_2} \left(x'_s - x'_j \right) \Bigg\} = 0 \end{aligned}$$

In other words, the bracketed term is perpendicular to x'_t for all $t = 1, 2, \dots, n$. Thus, we conclude that the bracketed term is zero. In particular, we have shown that the optimal dual multipliers $\{\theta_{ij}, w_{kj}\}$ give rise to a non–zero proper equilibrium stress for the points $\{(a_1; \mathbf{0}), \dots, (a_m; \mathbf{0}), \tilde{x}_1, \dots, \tilde{x}_n\}$. This completes the proof.

Corollary 4.1. *Let $\tilde{X}$ and $\{\bar{\theta}_{ij}, \bar{w}_{kj}\}$ be as in Theorem 4.1. Let*

$$\begin{aligned} \omega((a_k; \mathbf{0})) \quad \equiv & \sum_{s:(k,s)\in N_a} w_{ks}((a_k; \mathbf{0}) - \tilde{x}_s) \\ & - \sum_{s:(k,s)\in S_1} ((a_k; \mathbf{0}) - \tilde{x}_s) \\ & + \sum_{s:(k,s)\in C_1} ((a_k; \mathbf{0}) - \tilde{x}_s) \end{aligned}$$

be the net stress exerted on the pinned vertex $(a_k; \mathbf{0})$. Then, we have $\sum_{k=1}^m \omega((a_k; \mathbf{0})) = \mathbf{0}$.

Proof. The desired result follows upon summing (4.7) and the bracketed term in (4.8) over $s = 1, \dots, n$ and then interchanging the order of summation.

In particular, we see that if there is only one pinned vertex a_1, then the net stress exerted on a_1 is zero at the solution $(\tilde{X}, \bar{\theta}_{ij}, \bar{w}_{kj})$.

4.2 Computational Aspects

We now discuss the computational aspects of (3.1). In general, semidefinite programs can only be approximated in polynomial time to within an additive error of ϵ for any given $\epsilon > 0$. More precisely, we have the following (see, e.g., [12]):

Fact 4.2. *Let $\epsilon > 0$ be given. Then, the worst–case number of total arithmetic operations needed to compute a feasible solution (Z, U) to (3.1) and (3.3) with duality gap accuracy ϵ (i.e. $Z \bullet U \le \epsilon$) is $O(\sqrt{n}(n^3 + n^2|E| + |E|^3)\log(1/\epsilon))$, where $|E|$ is the number of edges in G.*

Definition 4.1. *We say that (Z, U) is an ϵ–approximate solution if (i) Z (resp. U) is a feasible solution to (3.1) (resp. (3.3)), and (ii) $Z \bullet U \leq \epsilon$ for some $\epsilon > 0$.*

To see how an ϵ–approximate solution affects the results of Theorem 4.1, we first observe the following:

Proposition 4.2. *Let (Z, U) be an ϵ–approximate solution for some $\epsilon \in (0, 1)$. Then, all entries of the matrix ZU have absolute value at most ϵ.*

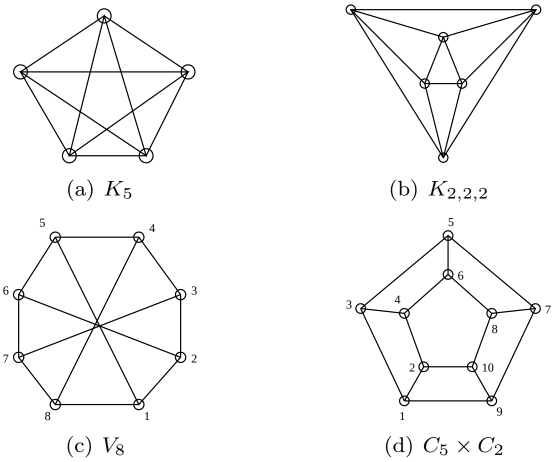
Now, using the arguments in the proof of Theorem 4.1, we can establish the following theorem. We omit the details here.

Theorem 4.2. *Let $\epsilon > 0$ be given. Then, there exists an $\delta > 0$ such that if $(\tilde{X}, \bar{\theta}_{ij}, \bar{w}_{ij})$ is extracted from an δ–approximate solution $(\bar{Z}, \bar{U})$ and $\omega(\tilde{x}_i)$ is the net stress exerted on the unpinned vertex $\tilde{x}_i$, we have $\|\omega(\tilde{x}_i)\| \leq \epsilon$ for all $i = 1, \ldots, n$. Moreover, δ can be chosen to be polynomially bounded.*

Thus, by using SDP, we can only find a solution that solves a "nearby" problem in polynomial time. We shall investigate this in greater detail in the full version of this paper.

5 An Algorithm for Realizing 3–Realizable Graphs

In this section, we give an algorithm for realizing 3–realizable graphs and hence resolve an open question posed in [14]. Before we proceed, let us recall some of the structural properties of an 3–realizable graph. It is clear from the definition that the class of d–realizable graphs is closed under the operation of taking minors. Thus, by the celebrated Graph Minor Theorem of Robertson and Seymour [26], there exists a finite list of forbidden minors for d–realizability. Using the results of Arnborg et. al. [5], Connelly and Sloughter [14] were able to show that a graph is 3–realizable iff it does not contain K_5 or $K_{2,2,2}$ as a minor (see Figure 1). Now, let G be an 3–realizable graph. Then, it either (i) contains an V_8 or an $C_5 \times C_2$ as a minor (see Figure 1), or (ii) does not contain either graphs as a minor. If it is the latter, then by the results of Arnborg et. al. [5] G is a *partial 3–tree* (to be defined below). Otherwise, by Proposition 1.7.2 of [18], G contains a *subdivision* of V_8 or $C_5 \times C_2$ (a graph H is a subdivision of a graph G if H can be obtained from G by replacing every edge (i, j) of G with a path from i to j). Now, let G' be the graph obtained by removing one such subdivision (call it H) from G. Then, it is shown in [14] that each component of G' can connect to only one of the subdivided edges of H in G. The preceding discussion suggests that we only need to find algorithms for the following tasks: (i) realizing a partial 3–tree; (ii) realizing an V_8 and its subdivisions; (iii) realizing an $C_5 \times C_2$ and its subdivisions; and (iv) finding a subdivision of V_8 or $C_5 \times C_2$ in an 3–realizable graph. We now formalize these ideas and treat each of these items in turn. In particular, we will show that if an optimal solution to (3.1) and (3.3) is available to us, then each of the tasks (i)–(iv) above can be achieved in polynomial time.



(a) K_5

(b) $K_{2,2,2}$

(c) V_8

(d) $C_5 \times C_2$

Figure 1: Forbidden Minors for Partial 3–Trees

5.1 Realizing Partial 3–Trees We first introduce the notions of k–tree and partial k–tree.

Definition 5.1. *An k-tree is defined recursively as follows. The complete graph on k vertices is an k-tree. An k–tree with $n + 1$ vertices (where $n \geq k$) can be constructed from an k–tree with n vertices by adding a vertex adjacent to all vertices of one of its k–vertex complete subgraphs, and only to those vertices. A partial k–tree is a subgraph of an k–tree.*

Suppose that we are given an 3–tree G with feasible edge lengths (i.e. there exists a realization of G in $\mathcal{R}^{d'}$ for some d' with those edge lengths), and that G is constructed by adding the vertices $v_1, v_2, \ldots, v_n$, in that order. It is then easy to find a realization of G in $\mathcal{R}^3$ in linear time. First, we realize the triangle v_1, v_2, v_3 in $\mathcal{R}^3$. Now, suppose that for $k \geq 3$, $v_1, \ldots, v_k$ has been realized. Then, in order to realize v_{k+1}, we simply compute its position via its neighbors $v_{i_1}, v_{i_2}, v_{i_3}$, where $1 \leq i_1, i_2, i_3 \leq k$. If the order of the vertices is not given, we can find an ordering in linear time [4, 27] and then proceed as above.

Now, suppose that we are given a partial 3–tree G with feasible edge lengths. We can first use, e.g., the SDP model in [33] to find a realization of G in $\mathcal{R}^d$ for some d. This way we also obtain all pairwise distances among the vertices in G. Now, we embed G into an 3–tree $\widehat{G}$ using the algorithm in [4]. Since this embedding does not introduce any extra vertices, we can use the relevant distance information on the edges of $\widehat{G}$ to realize $\widehat{G}$, and hence G, in $\mathcal{R}^3$ via the algorithm given in the preceding paragraph.

5.2 Realizing V_8 and its Subdivisions It is shown in [32] that V_8 (see Figure 1(c)) is 3–realizable. We now use the machinery developed in Sections 3 and 4 to give an algorithm for realizing V_8 in $\mathcal{R}^3$. We first augment V_8 to V_8' by adding a strut between vertices 1 and 4 (such a device is first introduced in [32]). Then, we pin vertex 1 at the origin (i.e. $a_1 = \mathbf{0}$). Suppose that the given edge lengths admit a realization whose vertices are in general position. Then, the graph V_8' satisfies the assumptions of Theorem 4.1. Now, consider the following semidefinite program:

(5.9)

$$\begin{array}{lll} \text{sup} & (a_1; e_4)(a_1; e_4)^T \bullet Z & \\ \text{s.t.} & Z_{1:3,1:3} = I_3 & \\ & (0; e_{ij})(0; e_{ij})^T \bullet Z = d_{ij}^2 & (i,j) \in E(V_8) \\ & & i,j \neq 1 \\ & (a_1; e_j)(a_1; e_j)^T \bullet Z = \bar{d}_{1j}^2 & (1,j) \in E(V_8) \\ & Z \succeq 0 & \end{array}$$

In other words, we would like to find a realization that maximizes the length of the strut. Let $\mathbf{p} = (\mathbf{0}, \tilde{x}_2, \ldots, \tilde{x}_8)$ be a realization obtained from an optimal solution to (5.9). Then, Theorem 4.1 and Corollary 4.1 guarantee that the corresponding dual multipliers yield a non–zero proper equilibrium stress ω for the tensegrity $V_8'(\mathbf{p})$. Once such an ω is given, we can use the procedure described in [32] to "round" the realization $\mathbf{p}$ into $\mathcal{R}^3$ in constant time. We omit the details here.

Now, consider a subdivision H of V_8. We label the vertices of H by $\{1, 2, \ldots, 8, s_1, \ldots, s_l\}$, where vertices $1, 2, \ldots, 8$ are those from V_8 and $s_1, \ldots, s_l$ are those that arise from subdivisions of edges. Note that we can determine the vertices $\{1, 2, \ldots, 8\}$ in time linear in the size of H. To realize H in $\mathcal{R}^3$, we can use the idea in Section 5.1, namely, we first find some realization of H and obtain the distance between vertices i and j in H, where $1 \leq i, j \leq 8$ and $(i,j) \in E(V_8)$. Then, we can feed that information to the SDP (5.9) and realize the vertices $1, 2, \ldots, 8$ in $\mathcal{R}^3$. To realize the remaining vertices in $\mathcal{R}^3$, it suffices to solve the following problem: given a path $u_1, \ldots, u_k$ with feasible edge lengths and pinned vertices u_1 and u_k, find a realization of the unpinned vertices such that the affine span of $\{u_1, \ldots, u_k\}$ is $\mathcal{R}^2$. This is equivalent to finding a realization in $\mathcal{R}^2$ of the graph G defined as follows: it has unpinned vertices $w_1, \ldots, w_k$ and edges $\{(w_i, w_{i+1}) : 1 \leq i \leq k-1\} \cup \{(w_1, w_k)\}$. The edge lengths are given by $\|w_{i+1} - w_i\| = \|u_{i+1} - u_i\|$ for $1 \leq i \leq k-1$, and $\|w_k - w_1\| = \|u_k - u_1\|$ (recall that u_1 and u_k are pinned and hence we can compute $\|u_k - u_1\|$ from their coordinates). Since G is series–parallel, it is 2–realizable by the result of [14]. Moreover, the desired realization can be computed in linear time (see Section 6 of [14]).

5.3 Realizing $C_5 \times C_2$ and its Subdivisions The graph $C_5 \times C_2$ (see Figure 1(d)) is 3–realizable [32], and we can realize it in $\mathcal{R}^3$ using similar ideas as above. We again assume that the assumptions of Theorem 4.1 are satisfied. To begin, we first augment $C_5 \times C_2$ to G by adding a strut between vertices 1 and 6, and we pin vertex 1 at the origin. Then, we formulate a semidefinite program similar to (5.9) and obtain a non–zero proper equilibrium stress ω for some realization of G in polynomial time. Then, we can again use ω and the arguments in [32] to "round" the solution into $\mathcal{R}^3$ in constant time. Lastly, we can realize subdivisions of $C_5 \times C_2$ in $\mathcal{R}^3$ using the ideas described in Section 5.2.

5.4 Finding a Subdivision of V_8 or $C_5 \times C_2$ in an 3–Realizable Graph Let G be an 3–realizable graph. We now show how the algorithm of Matoušek and Thomas [24] can be used to obtain a subgraph of G that is a subdivision of V_8 or $C_5 \times C_2$. To avoid clumsy sentences, we shall also use the term "homeomorphic" for subdivision – a graph H_1 is homeomorphic to H_2 if H_1 is a subdivision of H_2. We first recall the following facts:

FACT 5.1. *The following statements are true.*

(a) *(Asano [6]) For an 3–connected graph H, a graph H' has a subgraph homeomorphic to H iff there is an 3–connected component of H' that has a subgraph homeomorphic to H.*

(b) *(Connelly and Sloughter [14]) If an edge is added between a non–adjacent pair of vertices of V_8 (resp. $C_5 \times C_2$), then the resulting graph has K_5 (resp. K_5 or $K_{2,2,2}$) as a minor.*

(c) *(Connelly and Sloughter [14]) Let G be an 3–realizable graph. Suppose that G contains a subdivision of H, where $H \in \{V_8, C_5 \times C_2\}$. Remove the subdivision of H from G and consider the components of the resulting graph. Then, each component* *is connected in G to exactly one of the subdivided edges of H.*

Using these facts, we can establish the following:

Proposition 5.1. *Let G be an 3–realizable graph containing a subgraph homeomorphic to $H \in \{V_8, C_5 \times C_2\}$. Then, one of the triconnected components of G is isomorphic to H.*

Proof. Since V_8 and $C_5 \times C_2$ are 3–connected, if G has a subgraph homeomorphic to either one, then by Fact 5.1(a) one of the triconnected components will have a subgraph homeomorphic to H. Let G' be one such component. We claim that G' is isomorphic to H. Suppose that this is not the case. Let us remove the subdivision of H from G', and let G'' be one of the resulting components. (We may assume without loss that G'' is non–null, for a subdivision of H is either isomorphic to H or is not triconnected.) Using Facts 5.1(b) and 5.1(c), it can be shown that G'' is connected in G' to only one of the subdivided edges of H. However, if we remove the two end vertices of that subdivided edge in G', then we will disconnect G'' from the rest of G', which contradicts the fact that G' is triconnected. This establishes the claim.

Armed with Proposition 5.1, we propose the following algorithm. First, decompose G into triconnected components using the linear time algorithm in [22]. Then, we check each of the triconnected components for the presence or absence of V_8 or $C_5 \times C_2$. For this we can run the linear time algorithm in [24] on each of those components and see if the component reduces to a null graph or not. If the component does not reduce to a null graph, then it is isomorphic to either V_8 or $C_5 \times C_2$, and the number of vertices in the component will determine which one it is. The desired subdivision can then be extracted from G using the method outlined in [6]. We summarize as follows:

Proposition 5.2. *Let G be an 3–realizable graph with n vertices. Then, a subdivision of V_8 or $C_5 \times C_2$ in G can be found in $O(n)$ time.*

5.5 Putting Everything Together We are now ready to give the algorithm for realizing an 3–realizable graph G. First, we find a subgraph H of G that is homeomorphic to either V_8 or $C_5 \times C_2$ in G. If there is no such H, then G is a partial 3–tree, and hence we can use the algorithm in Section 5.1. Otherwise, we proceed as follows. For concreteness' sake, suppose that H is a subdivision of V_8, and its vertices are labelled using the convention in Section 5.2. We first realize vertices $1, 2, \ldots, 8$ of H in $\mathcal{R}^3$ as in Section 5.2. As a by–product, we also obtain the distance between vertices i and j, where $1 \le i < j \le 8$ and $(i,j) \in E(V_8)$. Now, consider the components that arise from the removal of H from G. By Fact 5.1(c), we can associate each of these components to a subdivided edge (i,j). Let $V_{(i,j)}$ be the union of all vertices from the components associated with (i,j) and all the vertices on the subdivided edge (i,j). Let $H_{(i,j)}$ be the induced subgraph of G on the vertices in $V_{(i,j)}$, and let $H'_{(i,j)}$ be obtained from $H_{(i,j)}$ by adding the edge (i,j) if it is not already there and assigning it the length $\|x_i - x_j\|$, where x_i is the position of vertex i computed above. Then, we can apply our algorithm on $H'_{(i,j)}$ and repeat until all vertices of G are realized.

6 Conclusion

In this paper we have studied a connection between SDP and tensegrity theories, as well as the notion of d–realizability of graphs, which is introduced in [14]. We have shown that the problem of finding an unyielding tensegrity can be formulated as an SDP, and that the optimal dual multipliers give rise to a non–zero proper equilibrium stress. This gives a constructive proof of (a variant of) a result in tensegrity theory that is previously established by non–constructive means, e.g. the Inverse Function Theorem. We then combine this result with the techniques in [14, 32] to design an algorithm for realizing 3–realizable graphs, thus answering an open question posed in [14]. We believe that our techniques can be applied to derive some other interesting properties of tensegrity frameworks, and this will be a subject of further research.

References

[1] Abdo Y. Alfakih, Henry Wolkowicz. On the Embeddability of Weighted Graphs in Euclidean Spaces. *Research Report CORR 98–12, University of Waterloo, Dept. of Combinatorics and Optimization*, 1998.

[2] Abdo Y. Alfakih, Henry Wolkowicz. Euclidean Distance Matrices and the Molecular Conformation Problem. *Research Report CORR 2002–17, University of Waterloo, Dept. of Combinatorics and Optimization*, 2002.

[3] Farid Alizadeh. Interior Point Methods in Semidefinite Programming with Applications to Combinatorial Optimization. *SIAM J. Opt.* 5(1):13–51, 1995.

[4] Stefan Arnborg, Andrzej Proskurowski. Characterization and Recognition of Partial 3–Trees. *SIAM J. Alg. Disc. Meth.* 7(2):305–314, 1986.

[5] Stefan Arnborg, Andrzej Proskurowski, Derek G. Corneil. Forbidden Minors Characterization of Partial 3–Trees. *Disc. Math.* 80:1–19, 1990.

[6] Takao Asano. An Approach to the Subgraph Homeomorphism Problem. *Theor. Comput. Sci.* 38:249–267, 1985.

[7] James Aspnes, David Goldenberg, Yang Richard Yang. On the Computational Complexity of Sensor Network Localization. *Proc. 1st ALGOSENSORS*, in *LNCS* 3121:32–44, 2004.

[8] Mihai Bădoiu, Erik D. Demaine, Mohammad Taghi Hajiaghayi, Piotr Indyk. Low–Dimensional Embedding with Extra Information. *Proc. 20th SoCG*, 2004.

[9] A. I. Barvinok. Problems of Distance Geometry and Convex Properties of Quadratic Maps. *Disc. Comput. Geom.* 13:189–202, 1995.

[10] Pratik Biswas, Tzu–Chen Liang, Kim–Chuan Toh, Yinyu Ye. An SDP Based Approach for Anchor–Free 3D Graph Realization. *Preprint*, 2005.

[11] Pratik Biswas, Yinyu Ye. Semidefinite Programming for Ad Hoc Wireless Sensor Network Localization. *Proc. 3rd IPSN*, 2004.

[12] Steven J. Benson, Yinyu Ye, Xiong Zhang. Solving Large–Scale Sparse Semidefinite Programs for Combinatorial Optimization. *SIAM J. Opt.* 10(2):443–461, 2000.

[13] Robert Connelly. Rigidity and Energy. *Invent. Math.* 66:11–33, 1982.

[14] Robert Connelly, Maria Sloughter. Realizability of Graphs. *Preprint*, 2004.

[15] Trevor F. Cox, Michael A. A. Cox. Multidimensional Scaling (2nd Edition). *Chapman & Hall/CRC*, 2000.

[16] Gordon M. Crippen, Timothy F. Havel. Distance Geometry and Molecular Conformation. *John Wiley & Sons, Inc.*, 1988.

[17] Jon Dattorro. *Convex Optimization & Euclidean Distance Geometry.* Ph.D. Thesis, Stanford, CA 94305, 2005.

[18] Reinhard Diestel. Graph Theory (2nd Edition). *Graduate Texts in Mathematics Volume 173, Springer–Verlag*, 2000.

[19] Lance Doherty, Kristofer S. J. Pister, Laurent El Ghaoui. Convex Position Estimation in Wireless Sensor Networks. *Proc. 20th INFOCOM*, 2001.

[20] Christoph Helmberg. Semidefinite Programming for Combinatorial Optimization. *ZIB–Report ZR–00–34, Konrad–Zuse–Zentrum Berlin*, 2000.

[21] Bruce Hendrickson. The Molecule Problem: Exploiting Structure in Global Optimization. *SIAM J. Opt.* 5(4):835–857, 1995.

[22] J. E. Hopcroft, R. E. Tarjan. Dividing a Graph into Triconnected Components. *SIAM J. Comput.* 2(3):135–158, 1973.

[23] Roger A. Horn, Charles R. Johnson. Matrix Analysis. *Cambridge University Press*, 1985.

[24] Jiří Matoušek, Robin Thomas. Algorithms Finding Tree–Decompositions of Graphs. *J. Algorithms* 12(1):1–22, 1991.

[25] Jorge J. Moré, Zhijun Wu. Global Continuation for Distance Geometry Problems. *SIAM J. Opt.* 7:814–836, 1997.

[26] Neil Robertson, P. D. Seymour. Graph Minors. XX. Wagner's Conjecture. *J. Comb. Theory, Ser. B* 92:325–357, 2004.

[27] Donald J. Rose. On Simple Characterizations of k–Trees. *Disc. Math.* 7:317–322, 1974.

[28] B. Roth, W. Whiteley. Tensegrity Frameworks. *Trans. AMS* 265(2):419–446, 1981.

[29] J. B. Saxe. Embeddability of Weighted Graphs in k–Space is Strongly NP–Hard. *Proc. 17th Allerton Conf. Commun. Control Comput.*, 1979.

[30] I. J. Schoenberg. Remarks to Maurice Fréchet's Article "Sur la Définition Axiomatique d'une Classe d'Espace Distanciés Vectoriellement Applicable sur l'Espace de Hilbert". *Ann. Math.* 36(3):724–732, 1935.

[31] Yi Shang, Wheeler Ruml. Improved MDS–Based Localization. *Proc. 23rd INFOCOM*, 2004.

[32] Maria Sloughter. Realizability of Graphs in Three Dimensions. *Preprint*, 2004.

[33] Anthony Man–Cho So, Yinyu Ye. Theory of Semidefinite Programming in Sensor Network Localization. *Proc. 16th SODA*, 2005.

[34] Lieven Vandenberghe, Stephen Boyd. Semidefinite Programming. *SIAM Review* 38(1):49–95, 1996.

[35] G. Young, A. S. Householder. Discussion of a Set of Points in terms of Their Mutual Distances. *Psychometrika* 3:19–22, 1938.

7 Appendix

7.1 A Pathological Case for the SDP In the discussion of Theorem 4.1 we have assumed that problem (3.1) is strictly feasible. It is interesting to see what could go wrong if that assumption is not satisfied. Consider the graph in Figure 2. Here, the vertices $1, 2$ and 3 are collinear, and the edges $(1, 2)$, $(2, 3)$ and $(1, 3)$ are all present. We would like to maximize the length of the strut $(2, 4)$. Clearly, such a configuration can span at most two dimensions, and hence the corresponding primal SDP (3.1) is not strictly feasible. Suppose now that we pin vertex 1 at the origin and try to solve the SDP. As argued before, the algorithm will assign a stress of -1 on the strut $(2, 4)$. However, the stress around vertex 2 will not be at equilibrium. The issue here is that when (3.1) is not strictly feasible, the optimal value may not be attained by any dual matrix U, even though there is a sequence of dual–infeasible matrices $\{U_n\}$ such that their associated objective values tend to the optimal value.

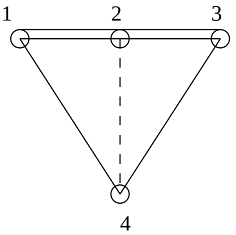


Figure 2: A Pathological Case for the SDP