Mathematical Programming manuscript No.
(will be inserted by the editor)

On Stress Matrices of $(d+1)$-lateration Frameworks in General Position

A. Y. Alfakih · Nicole Taheri · Yinyu Ye

Received: date / Accepted: date

Abstract Let (G, P) be a bar framework of n vertices in general position in $\mathbb{R}^d$, for $d \leq n-1$, where G is a $(d+1)$-lateration graph. In this paper, we present a constructive proof that (G, P) admits a positive semidefinite stress matrix with rank $(n-d-1)$. We also prove a similar result for a sensor network, where the graph consists of $m(\geq d+1)$ anchors.

Mathematics Subject Classification (2000) 52C25 · 05C62 · 15A57 · 90C22

Keywords Universal Rigidity · Stress Matrices · General Position · Semidefinite Programming

1 Introduction

Let $V(G)$ and $E(G)$ be, respectively, the vertex set and the edge set of a simple edge-weighted graph G, where each edge (i, j) has a positive weight d_{ij}. The *graph realization problem* (GRP) is the problem of determining whether there exists a realization of G in Euclidean space $\mathbb{R}^d$, for a given dimension d. A

Alfakih's research is supported by the Natural Sciences and Engineering Research Council of Canada. Taheri's research is supported in part by DOE Grant DE-SC0002009. Ye's research supported in part by NSF Grant GOALI 0800151 and DOE Grant DE-SC0002009. This paper is a shortened version of [7].

A. Y. Alfakih
Department of Mathematics and Statistics, University of Windsor, Windsor, Ontario N9B 3P4, Canada. E-mail: : alfakih@uwindsor.ca

Nicole Taheri
Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305. E-mail: ntaheri@stanford.edu

Yinyu Ye
Department of Management Science and Engineering, Stanford University, Stanford, CA 94305. E-mail: yinyu-ye@stanford.edu

(matrix) *realization* $P = [p_1, \dots, p_n]$ of G in $\mathbb{R}^d$ is a mapping $P : V(G) \to \mathbb{R}^{d\times n}$ such that, if vertices i and j of G are adjacent, then the Euclidean distance between points $p_i \in \mathbb{R}^d$ and $p_j \in \mathbb{R}^d$ is equal to the prescribed weight d_{ij} on the edge (i, j). We always assume that the points $p_1, \dots, p_n$ affinely span $\mathbb{R}^d$. In order words, P is a realization of G if and only if $p_1, \dots, p_n$ affinely span $\mathbb{R}^d$ and

$$\|p_i - p_j\| = d_{ij} \quad \text{for each edge } (i,j) \in E(G).$$

Throughout this paper, $\|x\|$ denotes the 2-norm of a vector x. Also, we use $\mathbf{0}$ to denote the matrix of all zeros of the appropriate dimension. See, e.g., [17, 18, 2, 3, 19, 14, 11, 21, 22, 4, 16, 23].

The GRP and its variants arise from applications in various areas, such as molecular conformation, dimensionality reduction, Euclidean ball packing, and more recently, wireless sensor network localization [5, 9, 8, 21, 10, 22, 20].

Let P be a given realization of graph G with n vertices in $\mathbb{R}^d$. A realization P together with a graph G is often referred to as a *bar framework* (or *framework*), and is denoted by (G, P). If P is the only realization of G in $\mathbb{R}^d$, up to a rigid motion (e. g., translation or rotation), then we say that the framework (G, P) is *globally rigid*. However, if P is the only realization of G, up to a rigid motion, in all dimensions, then we say that the framework (G, P) is *universally rigid*.

For a given framework (G, P) in $\mathbb{R}^d$, define the $(d + 1) \times n$ matrix A such that

$$A = \begin{bmatrix} P \\ e^T \end{bmatrix}, \tag{1}$$

where e is the vector of all 1's in $\mathbb{R}^n$. Matrices P and A are also respectively called the *position matrix* and the *extended position matrix* of the framework (G, P). The notion of a stress matrix plays a critical role in the characterization of the universal, as well as the global, rigidity of frameworks. An $n \times n$ symmetric matrix S is called a *stress matrix* of framework (G, P) if and only if

$$AS = \mathbf{0}, \tag{2}$$

and

$$S_{ij} = 0, \ \forall (i,j) \notin E(G), \tag{3}$$

where A is the extended position matrix of (G, P). Note that the highest possible rank of a stress matrix S is $(n-d-1)$, and the zero matrix is a trivial stress matrix.

The following theorem characterizes the universal rigidity of generic frameworks in terms of stress matrices. A framework (G, P) is said to be *generic*, or in *generic position*, if the coordinates of $p_1, \dots, p_n$ are algebraically independent over the integers, i.e., if there does not exist a non-zero polynomial f with integer coefficients such that $f(p_1, \dots, p_n) = 0$.

Theorem 1 *Let (G, P) be a framework of n vertices in generic position in $\mathbb{R}^d$, $d \le n - 1$. Then (G, P) is universally rigid if and only if there exists a stress matrix S of (G, P) such that S is positive semidefinite (PSD) and the rank of S is $n - d - 1$.*

The "if" part of this theorem was proved independently in [13] and [4], while the "only if" part was proved in [15].

One of the major research topics in rigidity is whether a result similar to Theorem 1 holds if the assumption of a framework in generic position is replaced by the weaker assumption of a framework in general position. We say that framework (G, P) is in *general position* in $\mathbb{R}^d$ if no $(d+1)$ points of $p_1, \ldots, p_n$ are affinely dependent. For example, points are in general position in $\mathbb{R}^2$ if no 3 of them are collinear. It then easily follows that if framework (G, P) in $\mathbb{R}^d$ is in general position, then every $(d+1) \times (d+1)$ square sub-matrix of the extended position matrix A, defined in (1), has rank $(d+1)$. Note that whether or not n rational points are in general position can be checked in time polynomial in n for any fixed dimension d, while the generic position condition is *uncheckable*. The following theorem, proved in [6] recently, shows that the "if" part of Theorem 1 still holds true under the general position assumption.

Theorem 2 *Let (G, P) be a framework of n vertices in general position in $\mathbb{R}^d$, $d \leq n-1$. Then (G, P) is universally rigid if there exists a stress matrix S of (G, P) such that S is positive semidefinite and the rank of S equals $n-d-1$.*

However, it remains an open question whether or not the converse of Theorem 2 holds true. In this paper, we settle this question in the *affirmative* for frameworks (G, P) in general position when G is a $(d+1)$-latertation graph. A graph of n vertices is called a $(d+1)$*-lateration graph* if there is a permutation π of the vertices, $\pi(1), \pi(2), \ldots, \pi(n)$, such that

- the first $(d+1)$ vertices, $\pi(1), \ldots, \pi(d+1)$, form a clique, and
- each remaining vertex $\pi(j)$, for $j = (d+2), \ldots n$, is adjacent to $(d+1)$ vertices in the set $\{\pi(1), \pi(2), \ldots, \pi(j-1)\}$.

Such frameworks were shown to be universally rigid in [20] and [23], where several classes of universally rigid frameworks in general position were identified.

In particular, we present a *constructive* proof that a framework (G, P) of n vertices in general position in $\mathbb{R}^d$, where G is a $(d+1)$-lateration graph, admits a PSD and rank $(n-d-1)$ stress matrix S. We show that such a stress matrix S can be computed in strongly polynomial time, if the $(d+1)$-lateration ordering is known. We also show that, if a graph G contains a $(d+1)$-lateration graph as a spanning subgraph, then the framework (G, P) in general position also admits a PSD stress matrix of rank $(n-d-1)$. Finally, a similar result for sensor networks, where the graph consists of $m(\geq d+1)$ anchors is also given.

2 The GRP and Semidefinite Programming (SDP)

If the graph realization problem is relaxed to the problem of determining whether a realization of the given edge-weighted graph G exists in some unspecified Euclidean space, then this relaxed problem can be modeled as a semidefinite programming problem (SDP). Furthermore, one can find a stress matrix with the maximum rank for any given framework by solving a pair of semidefinite programs (see also [21,22,4,10]). In particular, one can formulate a pair of dual SDPs where $A^T A$ is a solution to the primal problem, and the stress matrix is a solution to the dual problem. Here, A is the extended position matrix defined in (1). Next, we present one such formulation (for other SDP formulations of the same problem, see [1,5,9,21]).

Let the inner product of two matrices R and Q be defined by $R \cdot Q = \text{Trace}(R^T Q)$. An SDP for the relaxed graph realization problem attempts to find a symmetric matrix $Y \in \mathbb{R}^{n\times n}$ that solves

$$\begin{aligned} &\text{maximize } \mathbf{0} \cdot Y \\ &\text{subject to } (e_i - e_j)(e_i - e_j)^T \cdot Y = d_{ij}^2, \ \forall\ (i < j, j) \in E(G) \\ &\qquad\qquad Y \succeq \mathbf{0} \end{aligned} \tag{4}$$

where $e_j \in \mathbb{R}^n$ is the vector of all zeros except 1 at the jth position, and $Y \succeq \mathbf{0}$ constrains Y to be symmetric PSD. $A^T A$ and $P^T P$ are both feasible solutions to Problem (4) since

$$(e_i - e_j)(e_i - e_j)^T \cdot A^T A = \|a_i - a_j\|^2 = \|p_i - p_j\|^2 = d_{ij}^2, \ \forall\ (i < j, j) \in E(G).$$

The dual of Problem (4) is:

$$\begin{aligned} &\text{minimize } \textstyle\sum_{(i<j,j)\in E(G)} w_{ij} d_{ij}^2 \\ &\text{subject to } S := \sum_{(i<j,j)\in E(G)} w_{ij}(e_i - e_j)(e_i - e_j)^T \succeq \mathbf{0} \end{aligned} \tag{5}$$

Note that the dual problem is always feasible, since $w_{ij} = 0$ for all $(i, j) \in E(G)$ is a feasible solution. In fact, this solution is also optimal, since by the weak duality theorem, 0 is a lower bound on the objective.

From the duality theorem, any optimal solution S of (5) and any feasible solution Y of (4) will satisfy $Y \cdot S = \mathbf{0}$. This implies $A^T A \cdot S = ASA^T = 0$, or $AS = \mathbf{0}$. Moreover, $S_{ij} = 0,\ \forall (i, j) \notin E(G)$, so that any dual optimal solution is a PSD stress matrix. We say that the SDPs (4) and (5) admit a strictly complementary solution pair when their respective solutions (Y, S) satisfy $\text{rank}(Y) + \text{rank}(S) = n$.

Thus, the question of determining whether there is a non-trivial PSD stress matrix is equivalent to determining whether there is a non-trivial dual optimal solution, given that the primal problem is feasible. In particular, when the framework is universally rigid in $\mathbb{R}^d$, the primal problem (4) has a solution $Y = A^T A$ with rank $(d + 1)$. Hence, the SDP Problems (4) and (5) admit a strictly complementary solution pair if and only if there is a dual optimal solution S for (5) with rank $(n - d - 1)$.

Proposition 1 *A universally rigid framework of n vertices in $\mathbb{R}^d$, $d \le n - 1$, always admits a non-trivial positive semidefinite stress matrix.*

Proof. This follows simply from Theorem 6 in [1], which states that a framework (G, P) in $\mathbb{R}^d$ admits a non-trivial PSD stress matrix if and only if there does not exist a framework (G, Q) in $\mathbb{R}^{n-1}$ such that $||q_i - q_j|| = ||p_i - p_j||$ for all $(i, j) \in E(G)$.

□

The following result, stated in [21], answers the question of whether we could find a non-trivial PSD stress matrix, if it exists.

Proposition 2 *A primal solution Y of (4) that has the highest possible rank among all primal feasible solutions, together with a dual solution S of (5) that has the highest possible rank among all dual optimal solutions, can be computed approximately by an SDP interior-point algorithm in polynomial time of n, d, and* $\log(1/\epsilon)$ *with error ϵ.*

Proposition 2 also implies that if a universally rigid framework of n vertices in $\mathbb{R}^d$, $d \leq n-1$, admits a rank $(n-d-1)$ and PSD stress matrix, then such a stress matrix can be computed approximately in polynomial time. However, we may not be able to compute such a stress matrix exactly using the SDP algorithm, even when Y is known.

3 Main Result

The following theorem, whose proof is given at the end of this section, is our main result.

Theorem 3 *Let (G, P) be a framework of n vertices in general position in $\mathbb{R}^d$, $d \leq n-1$, where G is a $(d+1)$-lateration graph. Then (G, P) admits a positive semidefinite stress matrix with rank $(n-d-1)$. Moreover, such a stress matrix can be computed* exactly *in* strongly *polynomial time, $\mathcal{O}(n^3 + nd^3)$ arithmetic operations, if the lateration ordering and the position matrix P are known.*

An $n \times n$ symmetric matrix S that satisfies condition (2), i.e., $AS = \mathbf{0}$, is called a *pre-stress matrix.*[1] Our constructive proof of Theorem 3 first generates a PSD pre-stress matrix with rank $(n-d-1)$, then uses this pre-stress matrix as a basis to generate a PSD stress matrix with rank $(n-d-1)$. Recall that a stress matrix is a pre-stress matrix which also satisfies condition (3), i.e., $S_{ij} = 0,\ \forall (i, j) \notin E(G)$.

The following result follows from basic linear algebra.

Proposition 3 *For any framework in $\mathbb{R}^d$, there exists a pre-stress matrix which is positive semidefinite and has rank $(n-d-1)$. Moreover, a universally rigid framework in $\mathbb{R}^d$ on a complete graph has a rank $(n-d-1)$ positive semidefinite stress matrix.*

[1] The term *pre-stress* has been used by Connelly *et al* to mean something different, see [12].

For example, the projection matrix

$$I - A^T(AA^T)^{-1}A,$$

where A is the extended position matrix, is a PSD pre-stress matrix with rank $(n-d-1)$. Clearly, the projection matrix can be constructed in $\mathcal{O}(n^3)$ arithmetic operations.

Under the general position assumption, one can find a matrix $L \in \mathbb{R}^{n\times(n-d-1)}$ of the form

$$L = \begin{pmatrix} * & * & \cdots & * & * \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ * & \vdots & \ddots & \vdots & \vdots \\ 1 & * & \cdots & * & * \\ 0 & 1 & \cdots & * & * \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & * \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix},$$

that is, for $k = 1, ..., (n-d-1)$, $L_{ik} = 1$ for $i = d+1+k$ and $L_{ik} = 0$ for $i > (d+k+1)$, such that

$$AL = \mathbf{0},$$

where A is the extended position matrix. Clearly, L has rank $(n-d-1)$, thus $S = LL^T$ is a PSD pre-stress matrix with rank $(n-d-1)$. Such a matrix L is called a Gale matrix of framework (G, P) since its columns form a basis for the null space of A [1].

For a $(d+1)$-lateration graph G with lateration ordering $1, 2, \ldots, n$, and for a vertex $k \in V(G)$, let

$$N(k) = \{i \in V(G) : i \leq k-1 \text{ and } (i,k) \in E(G)\}. \tag{6}$$

Thus, for such a graph, $|N(k)| = d+1$ for each vertex $k = d+2, \ldots, n$. Furthermore, one can generate the kth column of L, L_k, for $k = 1, ..., (n-d-1)$, by solving the system of linear equations

$$\sum_{i\in N(k)} L_{ik} a_i = -a_{d+k+1}, \tag{7}$$

where a_i is ith column of the extended position matrix A, and assigning $L_{ik} = 0$ for all $i \notin N(k)$. The above $d \times d$ linear equation system can be solved in $\mathcal{O}(d^3)$ operations and there are $n-d-1$ many of them to solve, and the formation of S takes at most $\mathcal{O}(n^3)$ operations. Therefore, we have

Lemma 1 *The linear system (7) has a unique solution under the general position condition. Moreover, the matrix*

$$S^n = LL^T = \sum_{k=1}^{n-d-1} L_k L_k^T \succeq \mathbf{0}$$

is a pre-stress matrix with rank $(n-d-1)$*, and can be computed in* $\mathcal{O}(n^3+nd^3)$ *arithmetic operations.*

Next, we present an algorithm which uses S^n of Lemma 1 as a basis to generate the desired stress matrix.

3.1 A Purification Algorithm

If the pre-stress matrix S^n, as constructed in Lemma 1, satisfies condition (3), i.e., $S_{ij} = 0,\ \forall (i,j) \notin E(G)$, then it is the desired stress matrix. This is true if the graph is a $(d+1)$-tree graph, that is, if there is a permutation π of the vertices such that,

- the first $(d+1)$ vertices, $\pi(1), \ldots, \pi(d+1)$, form a clique, and
- each vertex $\pi(j)$, for $j = (d+2), \ldots, n$, is adjacent to the $(d+1)$ vertices of a $(d+1)$-clique in the set $\{\pi(1), \pi(2), \cdots, \pi(j-1)\}$.

In this case, any entry in $S^n = LL^T$, for $i < j$ and $(i,j) \notin E(G)$, is zero.

However, if S^n is not a stress matrix, we need to zero out the entries which should be zero but are not, i.e., the entries $S^n_{ij} \neq 0$, $i < j$ and $(i,j) \notin E(G)$. We do this in reverse order by column; first, we zero out the entries $S^n_{in} \neq 0$, for $i < n$ and $(i,n) \notin E(G)$, and then do the same for columns $(n-1), (n-2), \ldots, (d+3)$. This "purification" process will keep the pre-stress matrix PSD and maintain rank $(n-d-1)$.

If S^n is constructed from L as in the previous section, there is no need for purification of the last column (or row), since any entry in LL^T for $i < n$ and $(i,n) \notin E(G)$ is zero. But for general pre-stress matrices, this may not be the case. Therefore, we first show how to purify the last column (or row) of a PSD pre-stress matrix with rank $(n-d-1)$. We construct a vector $s^n \in \mathbb{R}^n$ with the elements,

$$s^n_i = -S^n_{in}\ , \forall (i,n) \notin E(G) \quad \text{and} \quad s^n_n = 1,$$

and solve the following system of linear equations for the remaining entries in s^n,

$$\sum_{i \in N(n)} s^n_i a_i = - \sum_{(i,n) \notin E(G)} s^n_i a_i. \tag{8}$$

The right-hand-side of the equation can be formed in at most $\mathcal{O}(nd)$ operations, and the $d \times d$ linear system can be solved in $\mathcal{O}(d^3)$ operations. Thus, s^n can be computed in at most $\mathcal{O}(nd + d^3)$ operations.

The linear system (8) has a unique solution under the general position condition, and by construction, $As^n = \mathbf{0}$.

Lemma 2 *Let* $S^{n-1} = S^n + s^n(s^n)^T$*. Then*

- $AS^{n-1} = \mathbf{0}$.
- $S^{n-1} \succeq \mathbf{0}$ *and the rank of* S^{n-1} *remains* $(n-d-1)$*.*
- $S^{n-1}_{in} = 0$ *for all* $i < n$*,* $(i,n) \notin E(G)$*.*

Proof. The first statement holds, since

$$AS^{n-1} = AS^n + As^n(s^n)^T = As^n(s^n)^T = \mathbf{0},$$

where the last step follows from the construction of s^n, so that $As^n = \mathbf{0}$.

The second statement follows from $S^{n-1} = S^n + s^n(s^n)^T \succeq S^n \succeq \mathbf{0}$. Thus, $\text{rank}(S^{n-1}) \geq \text{rank}(S^n) = (n-d-1)$, but $AS^{n-1} = \mathbf{0}$ implies that the rank of S^{n-1} is bounded above by $(n-d-1)$.

The third statement is also true by construction. In the last column (or row) of $s^n(s^n)^T$, the ith entry, where $i \neq n$ and $(i,n) \notin E(G)$, is precisely $-S^n_{in}$, i.e.,

$$\left(s^n(s^n)^T\right)_{in} = s^n_i s^n_n = s^n_i = -S^n_{in},$$

so that it is canceled out in the last column (or row) of matrix $S^{n-1} = S^n + s^n(s^n)^T$.

□

Note that update $S^{n-1} = S^n + s^n(s^n)^T$ uses $\mathcal{O}(n^2)$ arithmetic operations.

We continue this purification process for $(n-1), \ldots, k, \ldots, (d+3)$. Before the kth purification step, we have $S^k \succeq \mathbf{0}$, $AS^k = \mathbf{0}$, $\text{rank}(S^k) = (n-d-1)$, and

$$S^k_{ij} = 0, \ \forall j > k, i < j \text{ and } (i,j) \notin E(G)$$

We then construct a vector $s^k \in \mathbb{R}^n$ with the elements,

$$s^k_i = -S^k_{ik}, \ \forall (i,k) \notin E(G), \ s^k_k = 1, \quad \text{and} \quad s^k_i = 0 \ \forall i > k,$$

and solve the system of linear equations for the remaining entries in s^k:

$$\sum_{(i,k)\in E(G)} s^k_i a_i = -\sum_{(i,k)\notin E(G)} s^k_i a_i. \tag{9}$$

Again, solving this linear system takes at most $\mathcal{O}(nd+d^3)$ operations, and by construction, we have $As^k = \mathbf{0}$.

Similarly, the following lemma shows results analogous to those in Lemma 2, for the remaining columns.

Lemma 3 *Let $S^{k-1} = S^k + s^k(s^k)^T$. Then*

- $AS^{k-1} = \mathbf{0}$.
- $S^{k-1} \succeq \mathbf{0}$ *and the rank of* S^{k-1} *remains* $(n-d-1)$.
- $S^{k-1}_{ij} = 0$ *for all* $j \geq k$ *and* $i < j$, $(i,j) \notin E(G)$.

Proof. The proof of the first two statements is identical to that in Lemma 2.

The third statement is again true by construction. Note that in the kth column (or row) of $s^k(s^k)^T$, the ith entry, $i > k$ and $(i,k) \notin E(G)$, is precisely $-S^k_{ik}$, i.e.,

$$\left(s^k(s^k)^T\right)_{ik} = s^k_i s^k_k = s^k_i = -S^k_{ik},$$

so that it is canceled out in the kth column (or row) of matrix $S^{k-1} = S^k + s^k(s^k)^T$. Furthermore, for $j = (k+1), \ldots, n$, the jth column (or row) of $s^k(s^k)^T$

has all zero entries, which means the entries in jth column (or row) of S^{k-1} remain unchanged from S^k.

□

Now we are ready to prove our main result.

Proof of Theorem 3.

Assume that the $(d+1)$-lateration graph has the lateration ordering $1, 2, \dots, n$. The matrix S^{d+2}, constructed via the process described in Lemmas 2 and 3, will be a PSD stress matrix with rank $(n-d-1)$, for the $(d+1)$-lateration graph, since after step $k = (d+3)$, we will have a matrix S^{d+2} that satisfies,

$$AS^{d+2} = \mathbf{0} \quad \text{and} \quad S^{d+2}_{ij} = 0,\ \forall (i,j) \notin E(G)$$

Note that the first $(d+2)$ vertices form a clique in G, and the principal $(d+2) \times (d+2)$ submatrix has no zero entries. This stress matrix is unique and always exists since the graph is a $(d+1)$-lateration graph, and thus there is always a unique solution to the linear equation (9). Furthermore, by Lemma 3, $S^{d+2} \succeq \mathbf{0}$ and the rank of S^{d+2} remains $(n-d-1)$.

There are $(n-d-2)$ purification steps, where each step computes a rank-one matrix $s^k(s^k)^T$ and forms a new pre-stress matrix $S^k + s^k(s^k)^T$, taking at most $\mathcal{O}(n^2 + d^3)$ arithmetic operations. Thus, the computation of the max-rank PSD stress matrix uses at most $\mathcal{O}(n^3 + nd^3)$ operations.

□

We also have the following corollary:

Corollary 1 *Any universally rigid framework (G, P) in general position admits a positive semidefinite stress matrix with rank $(n-d-1)$, if G contains a $(d+1)$-lateration graph as a spanning subgraph. Moreover, such a stress matrix can be computed* exactly *in* strongly *polynomial time, $\mathcal{O}(n^3 + nd^3)$ arithmetic operations, if the lateration ordering and the position matrix P are known. Otherwise, such a stress matrix, together with the position matrix P, can be computed approximately by an SDP interior-point algorithm in time polynomial in n, d, and $\log(1/\epsilon)$, with error ϵ.*

This secondary result holds because we can ignore all edges outside of the $(d+1)$-lateration spanning subgraph to prove the existence of a PSD stress matrix with rank $(n-d-1)$. Since finding a $(d+1)$-lateration spanning subgraph requires at least $\mathcal{O}(n^{d+2})$ operations, we cannot actually construct such a stress matrix exactly in $\mathcal{O}(n^3 + nd^3)$ operations, if either the lateration ordering or the position matrix P is unknown. However, Proposition 2 implies that such a rank $(n-d-1)$ and PSD stress matrix, together with the position matrix P, can be computed approximately in polynomial time, although not strongly polynomial.

4 Strong Localizability of $(d+1)$-Lateration Graph with Anchors

In this section we study the stress matrix of a *sensor network*, or graph localization with anchors. A sensor network consists of $m(\geq d+1)$ anchor points whose positions, $\bar{p}_1, \ldots, \bar{p}_m \in \mathbb{R}^d$, are known, and n sensor points whose locations, $x_1, \ldots, x_n \in \mathbb{R}^d$, are yet to be determined. We are given the Euclidean distance values $\bar{d}_{kj}$ between $\bar{p}_k$ and x_j for some (k, j), and d_{ij} between x_i and x_j for some $i < j$. Specifically, let

$$N_a = \{(k, j) : \bar{d}_{kj} \text{ is specified}\} \quad \text{and} \quad N_x = \{(i, j) : i < j,\, d_{ij} \text{ is specified}\}.$$

The problem is to find a realization of $x_1, \ldots, x_n \in \mathbb{R}^d$ such that

$$\begin{aligned} \|\bar{p}_k - x_j\|^2 &= \bar{d}_{kj}^2 \ \forall\ (k, j) \in N_a \\ \|x_i - x_j\|^2 &= d_{ij}^2 \ \ \forall\ (i, j) \in N_x. \end{aligned} \tag{10}$$

The semidefinite programming relaxation model for (10) attempts to find a $(d + n) \times (d + n)$ symmetric matrix

$$Z = \begin{pmatrix} I_d & X \\ X^T & Y \end{pmatrix} \succeq \mathbf{0} \tag{11}$$

that solves the SDP

$$\begin{aligned} &\text{maximize } \mathbf{0} \cdot Z \\ &\text{subject to } Z_{1:d,1:d} = I_d \\ &\qquad (\mathbf{0}; e_i - e_j)(\mathbf{0}; e_i - e_j)^T \cdot Z = d_{ij}^2 \ \ \forall\ (i, j) \in N_x \\ &\qquad (-\bar{p}_k; e_j)(-\bar{p}_k; e_j)^T \cdot Z = \bar{d}_{kj}^2 \qquad \forall\ (k, j) \in N_a \\ &\qquad Z \succeq \mathbf{0}, \end{aligned} \tag{12}$$

where $(-\bar{p}_k; e_j) \in \mathbb{R}^{d+n}$ is the vector of $-\bar{p}_k$ vertically concatenated with e_j. $Z_{1:d,1:d}$ is the $d \times d$ top-left principal submatrix of Z and I_d is the d–dimensional identity matrix. $Z_{1:d,1:d} = I_d$ can be represented as $d(d + 1)/2$ linear equality constraints.

The dual of the SDP relaxation model is given by:

$$\begin{aligned} &\text{minimize } \ I_d \cdot V + \sum_{(i,j) \in N_x} w_{ij} d_{ij}^2 + \sum_{(k,j) \in N_a} \bar{w}_{kj} \bar{d}_{kj}^2 \\ &\text{subject to } S := \begin{pmatrix} V & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} + \sum_{(i,j) \in N_x} w_{ij} (\mathbf{0}; e_i - e_j)(\mathbf{0}; e_i - e_j)^T \\ &\qquad + \sum_{(k,j) \in N_a} \bar{w}_{kj} (-\bar{p}_k; e_j)(-\bar{p}_k; e_j)^T \succeq \mathbf{0}. \end{aligned} \tag{13}$$

Note that the dual is always feasible, since the symmetric matrix with $V = \mathbf{0} \in \mathbb{R}^{d \times d}$, variables $w_{ij} = 0$ for all $(i, j) \in N_x$ and $\bar{w}_{kj} = 0$ for all $(k, j) \in N_a$, is feasible for the dual. Also, each column $j = (d + 1), \ldots, (d + n)$ of the dual matrix S has the structure:

$$\begin{aligned} S_{1:d,j} &= -\textstyle\sum_{(k,j) \in N_a} \bar{w}_{kj} \bar{p}_k, \\ S_{ij} &= -w_{ij},\ (i, j) \in N_x, \\ S_{ij} &= 0,\ (i, j) \notin N_x, \\ S_{jj} &= \textstyle\sum_{(i,j) \in N_x} w_{ij} + \sum_{(k,j) \in N_a} \bar{w}_{kj}. \end{aligned} \tag{14}$$

Let w_{ij} and $\bar{w}_{kj}$ be called stress variables, and S be called a variable stress matrix for the sensor network localization problem.

It is shown in [21] that both SDPs (12) and (13) are feasible and solvable when there is at least one anchor point, the graph is connected, and there is no duality gap between the two SDPs. Let P be a position matrix of the n sensors satisfying constraints in (10). Then, the sensor network is said to be *uniquely localizable* if

$$Z = \begin{pmatrix} I_d & P \\ P^T & P^T P \end{pmatrix} \succeq \mathbf{0} \tag{15}$$

is the only matrix solution to the primal SDP (12); this is similar to the concept of universal rigidity. The network is said to be *strongly localizable* if there is an optimal dual stress matrix S such that

- $ZS = \mathbf{0}$,
- $S \succeq \mathbf{0}$ and $\text{rank}(S) = n$.

It has been shown in [21] that strong localizability implies unique localizability.

The standard graph realization problem is equivalent to the sensor network localization problem without anchors; thus, the two problems are different, but closely related. For example, unlike in the SDP (4), Z constructed from A, where A is the extended position matrix, is no longer feasible for (12), although Z constructed from a position matrix P in (15) is feasible. Hence, the stresses of the dual on the anchors may not need to be balanced. As another example, consider a sensor network of two anchors and one sensor in $\mathbb{R}^2$, where the distances from the sensor to the two anchors are known. The network is not uniquely localizable, but it is universally rigid in graph realization, since the three points form a clique.

However, if the sensor network has at least $(d+1)$ anchors in general position, and the graph realization problem has a $(d+1)$-point clique also in general position, then unique localizability is equivalent to universal rigidity, and strong localizability is equivalent to a framework on $(n+d+1)$ points having a PSD stress matrix with rank n (see [20,23]). The latter implies that the SDP pair (12) and (13) admits a strictly complementary solution pair.

Theorem 4 *Take a graph G of $m(\geq d+1)$ anchor points and n sensor points with edges given in N_x and N_a, and let G be a $(d+1)$-lateration graph with $(\bar{P}, P)$ in general positions. Then the sensor network is strongly localizable, and a rank n optimal dual stress matrix can be computed exactly in strongly polynomial time, $\mathcal{O}(n^3 + nd^3)$ arithmetic operations, if the lateration ordering and the sensor position matrix P are known.*

Sketch of Proof. (For full proof, see [7].)

We need to show that, in $\mathcal{O}(n^3 + nd^3)$ arithmetic operations, one can compute a symmetric matrix $S \in \mathbb{R}^{(d+n)\times(d+n)}$ which satisfies $ZS = \mathbf{0}$, $S \succeq \mathbf{0}$, $\text{rank}(S) = n$, and meets the structure condition (14). The proof is more complicated than that of Theorem 3, since anchor positions appear explicitly in the dual stress matrix.

For simplicity and without loss of generality, we assume there are exactly $(d+1)$ anchors which are the first $(d+1)$ points in the lateration ordering; all other points are sensors and ordered $1, \cdots, n$. Given a position matrix P, the primal feasible solution matrix Z in (15) can be written as $Z = [I_d \;\; P]^T [I_d \;\; P]$ so that the matrix $L = [-P; \;\; I_n] \in \mathbb{R}^{(d+n)\times n}$ is in the nullspace of Z or matrix $[I_d \;\; P]$. Moreover, the matrix

$$S^n = LL^T = \begin{pmatrix} PP^T & -P \\ -P^T & I_n \end{pmatrix} \succeq \mathbf{0},$$

will also be in the nullspace of Z, where $\mathrm{rank}(S^n) = \mathrm{rank}(L) = n$. One may call S^n a pre-stress matrix for the sensor localization problem. S^n may not be a true optimal stress matrix since it may not meet the structure condition (14).

Similar to the constructed proof of Theorem 3, while maintaining its rank n and keeping it PSD, we modify each column of the pre-stress matrix S^n, starting with the last, $(d+n)$, and continuing to column $(d+1)$, to make it a true stress matrix, optimal for the dual. Again, when modify column $d+\ell$, the $(d+j)$th column (or row) of the modified pre-stress matrix is unchanged for all $j > \ell$.

More precisely, when modifying the ℓth column, we construct a vector $s^\ell \in \mathbb{R}^{d+n}$ such that $s^\ell_{d+\ell} = 1$, $s^\ell_i = 0$ for $i > (d+\ell)$, and the first $(d+\ell-1)$ entries are

$$s^\ell_{1:(d+\ell-1)} := \begin{pmatrix} -\sum_{(k,\ell)\in N_a} \bar{w}_{k\ell}\bar{p}_k \\ -\sum_{(i<\ell,\ell)\in N_x} w_{i\ell} e_i \end{pmatrix} - S^\ell_{1:(d+\ell-1),(d+\ell)}, \tag{16}$$

where the $(d+1)$ stress variables $\bar{w}_{k\ell}$ and $w_{i\ell}$ are yet to be determined, and $e_i \in \mathbb{R}^n$ is the vector of all zeros except 1 at the ith position.

For the updated matrix $S^{\ell-1} := S^\ell + s^\ell(s^\ell)^T$, adding $s^\ell(s^{(\ell)})^T$ to S^ℓ will not affect any column (or row) to the right (or below) of column (or row) $d+\ell$. In particular, the $(d+\ell)$th column of $S^{\ell-1}$ becomes

$$\begin{pmatrix} S^{\ell-1}_{1:(d+\ell-1),(d+\ell)} \\ \\ S^{\ell-1}_{(d+\ell),(d+\ell)} \\ \\ S^{\ell-1}_{(d+\ell+1):(d+n),(d+\ell)} \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} -\sum_{(k,\ell)\in N_a} \bar{w}_{k\ell}\bar{p}_k \\ -\sum_{(i<\ell,\ell)\in N_x} w_{i\ell} e_i \end{pmatrix} \\ \\ 1 + S^\ell_{(d+\ell),(d+\ell)} \\ \\ S^\ell_{(d+\ell+1):(d+n),(d+\ell)} \end{pmatrix}$$

By construction, column $(d+\ell)$ of $S^{\ell-1}$ almost meets the the structure conditions of (14).

To ensure $S^{\ell-1}$ is orthogonal to Z, or $[I_d \;\; P]$, we determine the $(d+1)$ stress variables $\bar{w}_{k\ell}$ and $w_{i\ell}$ in s^ℓ such that $[I_d \;\; P]s^\ell = \mathbf{0}$, or equivalently,

$$-\sum_{(k,\ell)\in N_a} \bar{w}_{k\ell}\bar{p}_k - \sum_{(i<\ell,\ell)\in N_x} w_{i\ell}p_i + p_\ell = S^\ell_{1:d,(d+\ell)} + \sum_{i=1}^{\ell-1} S^\ell_{d+i,(d+\ell)} p_i. \tag{17}$$

Finally, to meet the diagonal entry value condition of (14), that is, the sum of the total edge stresses of sensor ℓ equals to the value of its diagonal element, we add

$$\sum_{(k,\ell)\in N_a} \bar{w}_{k\ell} + \sum_{(i<\ell,\ell)\in N_x} w_{i\ell} = 1 + S^{\ell}_{(d+\ell),(d+\ell)} + \sum_{i=\ell+1}^{n} S^{\ell}_{d+i,(d+\ell)}. \tag{18}$$

Equations (17) and (18) are exactly $(d+1)$ linearly independent equations on the $(d+1)$ stress variables, and thus there is always a unique solution.

Note that $ZS^{\ell-1} = Z(S^{\ell} + s^{\ell}(s^{\ell})^T) = Zs^{\ell}(s^{\ell})^T = \mathbf{0}$, $S^{\ell-1} \succeq S^{\ell}$, and $\mathrm{rank}(S^{\ell-1}) = n$. Moreover, every column (or row) j of $S^{\ell-1}$, $j \geq \ell$, meets condition (14). Repeating this process from column $(d+n)$ down to $(d+1)$ will result in a modified stress matrix S^0 that satisfies

- $ZS^0 = \mathbf{0}$,
- $S^0 \succeq \mathbf{0}$, and $\mathrm{rank}(S^0) = n$,
- and all columns of S^0 meet condition (14).

That is, S^0 is now a true optimal dual stress matrix with rank n. Note that there are a total of n modification steps, and each modification step takes at most $\mathcal{O}(n^2 + d^3)$ arithmetic operations.

□

Similar to the secondary result for the standard graph realization problem, we have the following corollary.

Corollary 2 *Take a graph G of $m(\geq d+1)$ anchor points and n sensor points with edges given in N_x and N_a, and let G contain a $(d+1)$-lateration spanning subgraph with $(\bar{P}, P)$ in general positions. Then the sensor localization problem on G is strongly localizable. Moreover, a rank n optimal dual stress matrix can be computed* exactly *in* strongly *polynomial time, $\mathcal{O}(n^3 + nd^3)$ arithmetic operations, if the lateration ordering and the position matrix P are known. Otherwise, such a rank n stress matrix, together with the position matrix P, can be computed approximately by an SDP interior-point algorithm in time polynomial in n, d, and $\log(1/\epsilon)$, with error ϵ.*

The argument for Corollary 2 is analogous to that of Corollary 1.

5 Examples

Consider a 3-lateration framework in dimension 2 on $n = 7$ nodes, with ordering $1, 2, \ldots, 7$, and position matrix

$$P = \begin{pmatrix} -1 & 1 & 0 & 2 & 1 & -1 & -2 \\ 1 & 1 & 0.5 & 0 & -1 & -1 & 0 \end{pmatrix} \in \mathbb{R}^{2\times 7}.$$

Example 1 Let

$$N(4) = \{1, 2, 3\},\ N(5) = \{1, 3, 4\},\ N(6) = \{1, 2, 4\},\ N(7) = \{3, 4, 5\}.$$

For this example,

$$L = \begin{pmatrix} 1.5000 & 5.0000 & -2.0000 & 0 \\ -0.5000 & 0 & 3.0000 & 0 \\ -2.0000 & -8.0000 & 0 & -1.6000 \\ 1.0000 & 2.0000 & -2.0000 & 1.4000 \\ 0 & 1.0000 & 0 & -0.8000 \\ 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix}$$

and we have the pre-stress matrix $S^7 = LL^T$,

$$S^7 = \begin{pmatrix} 31.2500 & -6.7500 & -43.0000 & 15.5000 & 5.0000 & -2.0000 & 0 \\ -6.7500 & 9.2500 & 1.0000 & -6.5000 & 0 & 3.0000 & 0 \\ -43.0000 & 1.0000 & 70.5600 & -20.2400 & -6.7200 & 0 & -1.6000 \\ 15.5000 & -6.5000 & -20.2400 & 10.9600 & 0.8800 & -2.0000 & 1.4000 \\ 5.0000 & 0 & -6.7200 & 0.8800 & 1.6400 & 0 & -0.8000 \\ -2.0000 & 3.0000 & 0 & -2.0000 & 0 & 1.0000 & 0 \\ 0 & 0 & -1.6000 & 1.4000 & -0.8000 & 0 & 1.0000 \end{pmatrix}.$$

Note that S^7 is already a stress matrix that meets condition (3), so that no "purification" algorithm is needed. This example is not interesting, since the graph is actually a 3-tree graph.

Example 2 Let

$$N_4 = \{1,2,3\},\ N_5 = \{1,3,4\},\ N_6 = \{2,4,5\},\ N_7 = \{1,3,6\}.$$

In this example,

$$L = \begin{pmatrix} 1.5000 & 5.0000 & 0 & -1.2500 \\ -0.5000 & 0 & -1.0000 & 0 \\ -2.0000 & -8.0000 & 0 & 1.0000 \\ 1.0000 & 2.0000 & 2.0000 & 0 \\ 0 & 1.0000 & -2.0000 & 0 \\ 0 & 0 & 1.0000 & -0.7500 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix}$$

and we the have the pre-stress matrix $S^7 = LL^T =$

$$S^7 = \begin{pmatrix} 28.8125 & -0.7500 & -44.2500 & 11.5000 & 5.0000 & 0.9375 & -1.2500 \\ -0.7500 & 1.2500 & 1.0000 & -2.5000 & 2.0000 & -1.0000 & 0 \\ -44.2500 & 1.0000 & 69.0000 & -18.0000 & -8.0000 & -0.7500 & 1.0000 \\ 11.5000 & -2.5000 & -18.0000 & 9.0000 & -2.0000 & 2.0000 & 0 \\ 5.0000 & 2.0000 & -8.0000 & -2.0000 & 5.0000 & -2.0000 & 0 \\ 0.9375 & -1.0000 & -0.7500 & 2.0000 & -2.0000 & 1.5625 & -0.7500 \\ -1.2500 & 0 & 1.0000 & 0 & 0 & -0.7500 & 1.0000 \end{pmatrix}.$$

While the last column (or row) of S^7 meets condition (3), the rest does not satisfy (3). We start the purification process from $k = 6$, where $S^6 = S^7$. The column vector s^6 is generated by first assigning

$$s_1^6 = -S_{1,6}^6 = -0.9375,\ s_3^6 = -S_{3,6}^6 = 0.75,\ s_6^6 = 1,\ s_7^6 = 0,$$

and then solving for (s_2^6, s_4^6, s_5^6) from the linear system (9) to get

$$s^6 = \begin{pmatrix} -0.9375 \\ -0.0625 \\ 0.7500 \\ 0.8750 \\ -1.6250 \\ 1.0000 \\ 0 \end{pmatrix}.$$

and $S^5 = S^6 + s^6(s^6)^T =$

$$S^5 = \begin{pmatrix} 29.6914 & -0.6914 & -44.9531 & 10.6797 & 6.5234 & 0 & -1.2500 \\ -0.6914 & 1.2539 & 0.9531 & -2.5547 & 2.1016 & -1.0625 & 0 \\ -44.9531 & 0.9531 & 69.5625 & -17.3438 & -9.2188 & 0 & 1.0000 \\ 10.6797 & -2.5547 & -17.3438 & 9.7656 & -3.4219 & 2.8750 & 0 \\ 6.5234 & 2.1016 & -9.2188 & -3.4219 & 7.6406 & -3.6250 & 0 \\ 0 & -1.0625 & 0 & 2.8750 & -3.6250 & 2.5625 & -0.7500 \\ -1.2500 & 0 & 1.0000 & 0 & 0 & -0.7500 & 1.0000 \end{pmatrix}.$$

Next the column vector s^5 is generated by first assigning

$$s_2^5 = -S_{2,5}^5 = -2.1016,\ s_5^5 = 1,\ s_6^5 = s_7^5 = 0,$$

and solving for (s_1^5, s_3^5, s_4^5) from linear system (9),

$$s^5 = \begin{pmatrix} 11.3047 \\ -2.1016 \\ -16.4063 \\ 6.2031 \\ 1.0000 \\ 0 \\ 0 \end{pmatrix}$$

and $S^4 = S^5 + s^5(s^5)^T$,

$$S^4 = \begin{pmatrix} 157.4874 & -24.4489 & -230.4207 & 80.8041 & 17.8281 & 0 & -1.2500 \\ -24.4489 & 5.6705 & 35.4319 & -15.5909 & 0 & -1.0625 & 0 \\ -230.4207 & 35.4319 & 338.7275 & -119.1138 & -25.6250 & 0 & 1.0000 \\ 80.8041 & -15.5909 & -119.1138 & 48.2444 & 2.7813 & 2.8750 & 0 \\ 17.8281 & 0 & -25.6250 & 2.7813 & 8.6406 & -3.6250 & 0 \\ 0 & -1.0625 & 0 & 2.8750 & -3.6250 & 2.5625 & -0.7500 \\ -1.2500 & 0 & 1.0000 & 0 & 0 & -0.7500 & 1.0000 \end{pmatrix}.$$

One can see that S^4 is now a desired stress matrix for Example 2.

Acknowledgments

The authors would like to thank two anonymous referees for their helpful comments and quick response.

References

1. A. Y. Alfakih. On bar frameworks, stress matrices and semidefinite programming. *to appear in Mathematical Programming.*
2. A. Y. Alfakih. Graph Rigidity via Euclidean Distance Matrices. *Linear Algebra and Its Applications* 310:149–165, 2000.
3. A. Y. Alfakih. On Rigidity and Realizability of Weighted Graphs. *Linear Algebra and Its Applications* 325:57–70, 2001.
4. A. Y. Alfakih. On the Universal Rigidity of Generic Bar Frameworks. *Contribution to Discrete Mathematics* 5(3):7–17, 2010.
5. A. Y. Alfakih, A. Khandani, H. Wolkowicz. Solving Euclidean Distance Matrix Completion Problems Via Semidefinite Programming. *Comput. Opt. and Appl.* 12:13–30, 1999.
6. A. Y. Alfakih and Y. Ye. On Affine Motions of Bar Frameworks in General Position. *arXiv/1009.3318*, 2010.
7. A. Y. Alfakih, N. Taheri, Y. Ye. Toward the Universal Rigidity of General Frameworks. *arXiv/1009.1185*, 2010.
8. James Aspnes, David Goldenberg, Yang Richard Yang. On the Computational Complexity of Sensor Network Localization. *ALGOSENSORS 2004*, in *LNCS* 3121:32–44, 2004.
9. P. Biswas, Y. Ye. Semidefinite Programming for Ad Hoc Wireless Sensor Network Localization. *Proc. 3rd IPSN* 46–54, 2004.
10. P. Biswas, K. C Toh and Y. Ye. A Distributed SDP approach for Large-scale Noisy Anchor-free Graph Realization with Applications to Molecular Conformation. *SIAM Journal on Scientific Computing* 30(3): 1251–1277, 2008.
11. T. Eren, D. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Moore, B. D. O. Anderson, P. N. Belhumeur. Rigidity, Computation, and Randomization in Network Localization. *Proc. 23rd INFOCOM*, 2004.
12. R. Connelly. Second-Order Rigidity and Pre-Stress Stability for Tensegrity Frameworks. *SIAM J. Discrete Math* 9(1):453–491, 1996.
13. R. Connelly. Tensegrity Structures: Why are they stable? In M. F. Thorpe and P. M. Duxbury editors, Rigidity Theory and Applications, pages 47–54, Kluwer Academic/Plenum Publishers, 1999.
14. R. Connelly. Generic Global Rigidity. *Discrete and Computational Geometry*, 33(4):549–563, 2005.
15. S. J. Gortler and D. P. Thurston. Characterizing the universal rigidity of generic frame-works. *arXiv/1001.0172v1*, 2009.
16. S. J. Gortler, A. D. Healy, D. P Thurston. Characterizing Generic Global Rigidity. *American Journal of Mathematics*, 2010.
17. J. Graver, B. Servatius, H. Servatius. Combinatorial Rigidity. *AMS*, 1993.
18. B. Hendrickson. Conditions for Unique Graph Realizations. *SIAM J. Comput.* 21(1):65–84, 1992.
19. B. Jackson, T. Jordán. Connected Rigidity Matroids and Unique Realizations of Graphs. Preprint, 2003.
20. A. M.-C. So. A Semidefinite Programming Approach to the Graph Realization Problem: Theory, Applications and Extensions. Ph.D. Thesis, Stanford University, 2007.
21. A. M.-C. So and Y. Ye. Theory of Semidefinite Programming for Sensor Network Localization. *Proceedings of the 17th Annual ACM–SIAM Symposium on Discrete Algorithm (SODA 2005)* 2005, pp. 405–414; and *Mathematical Programming, Series B*, vol. 109, no. 2, pp. 367–384, 2007.

22. A. M.-C. So and Y. Ye. A Semidefinite Programming Approach to Tensegrity Theory and Realizability of Graphs. *Proceedings of the 18th Annual ACM–SIAM Symposium on Discrete Algorithm (SODA 2006)*, 2006, pp. 766–775.
23. Z. Zhu, A. M-C So, Y. Ye. Universal Rigidity: Towards Accurate and Efficient Localization of Wireless Networks. *Proc. IEEE INFOCOM*, 2010.