Portugal. Math. (N.S.)
Vol. 68, Fasc. 1, 2011, 53–102
DOI 10.4171/PM/1881

Portugaliae Mathematica
© European Mathematical Society

Euclidean distance matrices, semidefinite programming and sensor network localization

Abdo Y. Alfakih, Miguel F. Anjos, Veronica Piccialli and Henry Wolkowicz*

(Communicated by Luis Nunes Vicente)

Abstract. The fundamental problem of distance geometry involves the characterization and study of sets of points based only on given values of some or all of the distances between pairs of points. This problem has a wide range of applications in various areas of mathematics, physics, chemistry, and engineering. Euclidean distance matrices play an important role in this context by providing elegant and powerful convex relaxations. They play an important role in problems such as graph realization and graph rigidity. Moreover, by relaxing the embedding dimension restriction, these matrices can be used to approximate the hard problems efficiently using semidefinite programming. Throughout this survey we emphasize the interplay between these concepts and problems. In addition, we illustrate this interplay in the context of the sensor network localization problem.

Mathematics Subject Classification (2010). Primary 51K05; Secondary 15A83, 52C25, 90C22.

Keywords. Euclidean distance matrix completions, sensor network localization, fundamental problem of distance geometry, semidefinite programming.

1. Introduction

The fundamental problem of distance geometry (**FPDG**) involves the characterization and study of sets of points, $p_1, \dots, p_n \in \mathbb{R}^r$, based only on given values for some or all of the distances between pairs of points. More precisely, given only (partial, approximate) distance information $\bar{d}_{ij} \approx \|p_i - p_j\|^2$ for all $ij \in E$[1], for some given subset E of pairs of points, we need to determine whether we can realize such a set of points in a given dimension and also find these points efficiently. This problem has a wide range of applications, in various areas of mathematics,

*The authors are grateful for the support of the Natural Sciences Engineering Research Council of Canada, MITACS, the AFOSR, and the Alexander von Humboldt Foundation.

[1] We use the bar to emphasize that these distances are not necessarily exact.

physics, chemistry, astronomy, engineering, music, etc. Surprisingly, there are many classes of **FPDG** problems where this hard inverse problem with incomplete data can be solved efficiently.

Euclidean Distance Matrices (**EDMs**) play an important role in this context since they provide an elegant and strong relaxation for **FPDG**. The entries of an **EDM** consist of squared Euclidean distances between points: $D_{ij} = \|p_i - p_j\|^2$, $i, j = 1, \dots, n$. Using the squared rather than ordinary distances, and further removing the constraint on the embedding dimension r, means that completing a partial **EDM** is a convex problem. Moreover, a global solution for this relaxed problem can be found efficiently using semidefinite programming (**SDP**). This is related to problems in the area of *compressed sensing*, i.e., the restriction on the embedding dimension is equivalent to a rank restriction on the semidefinite matrix using the **SDP** formulation. (See e.g., [19], [88] for details on compressed sensing.)

A special instance of **FPDG** is the Sensor Network Localization (**SNL**) problem. **SNL** recently emerged as an important research topic. For **SNL**, the n points are sensors that are part of a wireless ad hoc sensor network. Each sensor has some wireless communication and signal processing capability. In particular, m of these sensors are anchors (or beacons) whose positions are known, and the distances between two sensors are (approximately) known if and only if the sensors are within a given radio range R.

In this survey we concentrate on the **SNL** problem and its connections with **EDM**, graph realization (**GRL**), graph rigidity (**GRD**), and **SDP**. Our goal is to show that these NP-hard problems can be handled elegantly within the **EDM** framework, and that **SDP** can be used to efficiently find accurate solutions for many classes of these problems. In particular, working within the **EDM** framework provides strong solution techniques for **SNL**.

2. Preliminaries, notation

Let $\mathcal{M}^{kl}$ denote the space of $k \times l$ real matrices, and let $\mathcal{M}^k = \mathcal{M}^{kk}$. For $M \in \mathcal{M}^n$, we let $\operatorname{diag} M$ denote the vector in $\mathbb{R}^n$ formed from the diagonal of M. Then, for any vector $v \in \mathbb{R}^n$, $\operatorname{Diag} v = \operatorname{diag}^* v$ is the adjoint linear transformation consisting of the diagonal matrix with diagonal formed from the vector v. We let $\operatorname{vec} X$ denote the vector formed from the columns of X.

We work with points (real vectors) $p_1, \dots, p_n \in \mathbb{R}^r$, where r is the *embedding dimension* of the problem. We let $P^T = [p_1, \dots, p_n] \in \mathcal{M}^{rn}$ denote the matrix with columns formed from the set of points. For **SNL**, $P = \left[\begin{smallmatrix} A \\ X \end{smallmatrix}\right]$, where the rows $p_i^T = a_i^T$, $i = 1, \dots, m$, of $A \in \mathcal{M}^{mr}$ are the positions of the m anchor nodes, and the rows $x_i^T = p_{m+i}^T$, $i = 1, \dots, n-m$, of $X \in \mathcal{M}^{(n-m)r}$ are the positions of the remaining $n-m$ sensor nodes. We let $G = (V, E)$ denote the *simple graph* on the vertices $1, 2, \dots, n$ with edge set E. Typically, for **FPDG** the distances $\|x_i - x_j\|$, $ij \in E$, are the ones that are known.

The vector space of *real symmetric* $n \times n$ *matrices* is denoted $\mathcal{S}^n$, and is equipped with the *trace inner product*, $\langle A, B\rangle = \operatorname{trace} AB$, and the corresponding *Frobenius norm*, denoted $\|A\|_F$. More generally, $\langle A, B\rangle = \operatorname{trace} A^T B$ denotes the inner product of two compatible, general, real matrices A, B, and $\|A\|_F = \sqrt{\operatorname{trace} A^T A}$ is the Frobenius norm. We let $\mathcal{S}^n_+$ and $\mathcal{S}^n_{++}$ denote the cone of positive semidefinite and positive definite matrices, respectively. In addition, $A \succeq B$ and $A \succ B$ denote the Löwner partial order, $A - B \in \mathcal{S}^n_+$ and $A - B \in \mathcal{S}^n_{++}$, respectively. Moreover, $A \geq 0$ denotes A nonnegative elementwise. We let $\mathcal{E}^n$ ($\mathcal{E}$ when the dimension is clear) denote the cone of Euclidean distance matrices $D \in \mathcal{S}^n$, i.e., the elements of a given $D \in \mathcal{E}^n$ are $D_{ij} = \|p_i - p_j\|^2$, for some fixed set of points $p_1, \dots, p_n$. We let e_i denote the i-th unit vector, e denote the vector of ones, both of appropriate dimension, and, by abuse of notation[2], $E = ee^T$; $\mathcal{R}(\mathcal{L})$, $\mathcal{N}(\mathcal{L})$ denotes the range space and nullspace of the linear transformation $\mathcal{L}$, respectively; $\mathcal{L}^*$ denotes the adjoint of $\mathcal{L}$, i.e., $\langle \mathcal{L}(x), y\rangle = \langle x, \mathcal{L}^*(y)\rangle$ for all x, y; $\mathcal{L}^\dagger$, denotes the Moore–Penrose generalized inverse of $\mathcal{L}$; and $A \circ B = (A_{ij}B_{ij})$ denotes the *Hadamard* (*elementwise*) *product* of two matrices.

We follow the notation in [70], [73]: for $Y \in \mathcal{S}^n$ and $\alpha \subseteq 1 : n$, we let $Y[\alpha]$ denote the corresponding *principal submatrix* formed from the rows and columns with indices α. If, in addition, $|\alpha| = k$ and $\bar{Y} \in \mathcal{S}^k$ is given, then we define

$$\mathcal{S}^n(\alpha, \bar{Y}) := \{Y \in \mathcal{S}^n : Y[\alpha] = \bar{Y}\}, \qquad \mathcal{S}^n_+(\alpha, \bar{Y}) := \{Y \in \mathcal{S}^n_+ : Y[\alpha] = \bar{Y}\},$$

i.e., the subset of matrices $Y \in \mathcal{S}^n$ ($Y \in \mathcal{S}^n_+$) with principal submatrix $Y[\alpha]$ fixed to $\bar{Y}$. Similar notation, $\mathcal{E}^n(\alpha, \bar{D})$, holds for subsets of $\mathcal{E}^n$.

The *centered* and *hollow* subspaces of $\mathcal{S}^n$ (and the offDiag linear operator) are defined by

$$\begin{aligned} \mathcal{S}_C &:= \{B \in \mathcal{S}^n : Be = 0\} \qquad \text{(zero row sums)}, \\ \mathcal{S}_H &:= \{D \in \mathcal{S}^n : \operatorname{diag} D = 0\} = \mathcal{R}(\text{offDiag}). \end{aligned} \tag{2.1}$$

The set $K \subset \mathbb{R}^n$ is a *convex cone* if $\mathbb{R}_+(K) \subseteq K$, $K + K \subseteq K$. cone(S) denotes the smallest convex cone containing S, i.e., the *generated convex cone of* S. A set $F \subseteq K$ is a *face of the cone* K, denoted $F \trianglelefteq K$, if

$$\left(x, y \in K, \frac{1}{2}(x + y) \in F\right) \implies (\operatorname{cone}\{x, y\} \subseteq F).$$

[2] We let E refer to the edge set of the graph G and to the matrix of ones. The meaning is clear from the context.

We write $F \lhd K$ to denote $F \unlhd K$, $F \neq K$. If $\{0\} \neq F \lhd K$, then F is a *proper face* of K. For $S \subseteq K$, we let face(S) denote the smallest face of K that contains S. For example, let L be some fixed subspace. Then the set of all positive semidefinite matrices whose null space contains L is a face of the positive semidefinite cone.

For a set $S \subset \mathbb{R}^n$, let $S^* := \{\phi \in \mathbb{R}^n : \langle \phi, S \rangle \subseteq \mathbb{R}_+\}$ denote the *dual cone of S*. That $\mathcal{S}_+^n = \mathcal{S}_+^{n*}$ is well known, i.e., the **SDP** cone is self-dual. Due to the importance of the **SDP** cone, we include the following interesting geometric result. This result emphasizes the difference between $\mathcal{S}_+^n$ and a polyhedral cone: it illustrates the nice property that the first sum using $F^\perp$ in (2.2) is always closed for any face; but the sum in (2.3) using span is never closed. The lack of closure leads to problems in duality. Here $F^c = \mathcal{S}_+^n \cap F^\perp$ denotes the *conjugate face* of F.

Lemma 2.1 ([105], [87]). *Suppose that F is a proper face of $\mathcal{S}_+^n$, i.e., $\{0\} \neq F \lhd \mathcal{S}_+^n$. Then*

$$F^+ = \mathcal{S}_+^n + F^\perp = \overline{\mathcal{S}_+^n + \operatorname{span} F^c}, \tag{2.2}$$

$$\mathcal{S}_+^n + \operatorname{span} F^c \textit{ is not closed.} \tag{2.3}$$

3. FPDG and EDM

Distance geometry involves the characterization and study of sets of points based only on given values of some or all of the distances between pairs of points. The origins of the algebra for distance geometry can be traced back to 1896 and the work of Grassmann [53] and continued in the modern era in [51], [31], [39] among others. One of the methods used to study **FPDG** is to view the problem using the squared distances, i.e., using **EDMs**. This allows the application of powerful tools from convex analysis and linear algebra, and specifically from **SDP**. This is the approach we emphasize in this survey.

Theoretical properties of **EDMs** can be found in [9], [41], [17], [52], [57], [66], [77], [93]. This includes characterizations as well as graph theoretic conditions (such as *chordality*) for the existence of completions of partial **EDMs**, i.e., for the **EDM** completion problem (**EDMC**). More information can be found in the survey article [77], and more recently in the book [33]. A discussion on the difficulty of finding efficient algorithms for **EDMC** appears in [98]. There are many algorithms that find approximate completions. For example [98], [97], [96] present results on finding **EDM** completions based on spectral decompositions. The computationally hard part is fixing the rank. Work on finding the closest **EDM** to a given symmetric matrix appears in [48], [108], [4]. (The harder global model without *squared* distances but with intervals for the distances, is used in [80], [81], [109].)

We now present **FPDG** using the *squared* distances between the points, the **EDM** model. A matrix $D = (D_{ij}) \in \mathcal{S}^n$ with nonnegative elements and zero diagonal is called a *pre-distance matrix* or a *dissimilarity matrix*. In addition, if there exist r and points $p_1, p_2, \dots, p_n \in \mathbb{R}^r$ such that

$$D_{ij} = \|p_i - p_j\|_2^2, \qquad i, j = 1, 2, \dots, n, \tag{3.4}$$

then D is called a *Euclidean distance matrix*, denoted **EDM**. The set of **EDM** matrices forms a convex cone in $\mathcal{S}^n$, denoted $\mathcal{E}^n$. This cone is closed, pointed ($\mathcal{E}^n \cap -\mathcal{E}^n = \{0\}$), but has empty interior. Given $D \in \mathcal{E}^n$, then the smallest value of r such that points p_i can be found satisfying (3.4) is called the *embedding dimension* of D.

Suppose that we are given a subset of the elements of a pre-distance matrix D, i.e., we are given a partial **EDM**, D. Then the **EDM** *completion problem* (**EDM**C) consists in finding the missing elements of D to complete the **EDM**, and/or determine that this is not possible. Equivalently, this means that we have found a set of points for (3.4). Alternatively, suppose that we are given an *approximate* pre-distance (or partial distance) matrix $\bar{D}$ and a symmetric matrix of *nonnegative weights* W. Then the *approximate* (*nearest*) **EDM** *completion problem* can be modelled as [65], [2]

$$\min \| W \circ (\bar{D} - D) \| \text{ such that } D \in \mathcal{E}. \tag{3.5}$$

The most common norms for the objective function are the Frobenius and ℓ_1 norms. The magnitude of the weights in W typically come from consideration of the magnitudes of the known distances and any knowledge of the error/noise [15]

$$W_{ij} := \begin{cases} \frac{1}{\sqrt{\bar{D}_{ij}}} & \text{if the } ij\text{-distance is approximately } \sqrt{\bar{D}_{ij}}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.6}$$

3.1. Distance geometry, EDM , and SDP. Let $P^T = [p_1 \ p_2 \ \cdots \ p_n] \in \mathcal{M}^{rn}$ be as defined above in Section 2, where p_j, $j = 1, \dots, n$, are the points used in (3.4). We assume that P is full column rank r. Let $B = PP^T$. Then $B \succeq 0$ is also of rank r. Now, define the linear operators $\mathcal{K}$ and $\mathcal{D}_e$ on $\mathcal{S}^n$ by

$$\begin{aligned} \mathcal{K}(B) &:= \mathcal{D}_e(B) - 2B \\ &:= \operatorname{diag} B e^T + e \operatorname{diag} B^T - 2B \\ &= (p_i^T p_i + p_j^T p_j - 2p_i^T p_j)_{i,j=1}^n \\ &= (\|p_i - p_j\|_2^2)_{i,j=1}^n \\ &= D. \end{aligned} \tag{3.7}$$

This illustrates the relationship between p_j, P, B, D, i.e., a mapping between $\mathscr{E}^n$, $\mathscr{S}^n_+$. Now let $J := I - \frac{1}{n}ee^T$ denote the orthogonal projection onto the subspace $\{e\}^\perp$; and, define the linear operator

$$\mathscr{T}(D) := -\frac{1}{2}J \operatorname{offDiag}(D)J, \tag{3.8}$$

where offDiag(D) replaces the diagonal of D with zeros; see (2.1). The linear operators $\mathscr{K}$, $\mathscr{T}$ are one-to-one and onto between the *centered* and *hollow* subspaces of $\mathscr{S}^n$. In the classical literature, the linear operator $\mathscr{T}$ is only defined on the subspace $\mathscr{S}_H$. We extend it to all of $\mathscr{S}^n$ with the addition of the operator (projection) offDiag. This means that we now have a simple explicit expression for the Moore–Penrose generalized inverse $\mathscr{K}^\dagger = \mathscr{T}$. See (2.1) and Proposition 3.2 below.

From the definition of the positive semidefinite matrix B, we see that the inner products in the elements B_{kl} can be used to form the squared distances D_{ij} in (3.4). Therefore, the linear operators $\mathscr{K}$, $\mathscr{T}$ map between the cones $\mathscr{S}^n_+$, $\mathscr{E}^n$. The following linear transformation $\mathscr{L}$ provides an alternative to $\mathscr{K}$.

Lemma 3.1 ([5]). *Let $X \in \mathscr{S}^{n-1}$ and partition*

$$\mathscr{L}(X) := \begin{bmatrix} 0 & \operatorname{diag} X^T \\ \operatorname{diag} X & \mathscr{D}_e(X) - 2X \end{bmatrix} = \begin{bmatrix} 0 & d^T \\ d & \bar{D} \end{bmatrix} := D. \tag{3.9}$$

Then $\mathscr{E}^n = \mathscr{L}(\mathscr{S}^{n-1}_+)$ and

$$\mathscr{L}^*(D) = 2\{\operatorname{Diag}(d) + \operatorname{Diag}(\bar{D}e) - \bar{D}\}, \qquad \mathscr{L}^\dagger(D) = \frac{1}{2}(de^T + ed^T - \bar{D}).$$

Following are several relationships for $\mathscr{K}$, $\mathscr{T}$ acting on $\mathscr{S}^n$. In particular, the adjoint and generalized inverse of $\mathscr{K}$ have explicit, easy to use, expressions.

Proposition 3.2 ([5]). *The operators $\mathscr{K}$, $\mathscr{T}$ satisfy*

$$\mathscr{K}(\mathscr{S}^n_+) = \mathscr{E}^n \cap \mathscr{S}_H = \mathscr{E}^n, \qquad \mathscr{T}(\mathscr{E}^n) = \mathscr{S}^n_+ \cap \mathscr{S}_C. \tag{3.10}$$

The adjoint and generalized inverse of $\mathscr{K}$ are, respectively,

$$\mathscr{K}^*(D) = 2\big(\operatorname{Diag}(De) - D\big), \qquad \mathscr{K}^\dagger = \mathscr{T}. \tag{3.11}$$

Moreover,

$$\mathcal{R}(\mathcal{K}) = \mathcal{S}_H, \qquad \mathcal{N}(\mathcal{K}) = \mathcal{R}(\mathcal{D}_e) \tag{3.12}$$

$$\mathcal{R}(\mathcal{K}^*) = \mathcal{R}(\mathcal{T}) = \mathcal{S}_C, \qquad \mathcal{N}(\mathcal{K}^*) = \mathcal{N}(\mathcal{T}) = \operatorname{Diag}(\mathbb{R}^n) \tag{3.13}$$

$$\mathcal{S}^n = \mathcal{S}_H \oplus \operatorname{Diag}(\mathbb{R}^n) = \mathcal{S}_C \oplus \mathcal{R}(\mathcal{D}_e). \tag{3.14}$$

3.1.1. Characterizations of the EDM cone and facial reduction. It is well known that a nonnegative, hollow matrix, $0 \le D \in \mathcal{S}_H$, is a **EDM** if and only if D is negative semidefinite on $\{e\}^\perp$, the orthogonal complement of e; see [93], [52], [57], [95]. We now collect this with other characterizations [2], [34]. First, define the $n \times n$ orthogonal matrix $Q := \left[\frac{1}{\sqrt{n}}e, | V\right]$, $Q^TQ = I$, i.e., $V^Te = 0$ and $V^TV = I$. Then the projection $J = I - \frac{ee^T}{n} = VV^T$. Now define the composite linear transformation

$$\mathcal{K}_V(B_V) := \mathcal{K}(VB_VV^T). \tag{3.15}$$

The adjoint of K_V is

$$\mathcal{K}_V^*(D) = V^T\mathcal{K}^*(D)V. \tag{3.16}$$

Let

$$\mathcal{T}_V(D) := V^T\mathcal{T}(D)V = -\frac{1}{2}V^TDV. \tag{3.17}$$

Lemma 3.3 ([2]). *$\mathcal{K}_V(\mathcal{S}^{n-1}) = \mathcal{S}_H$, $\mathcal{T}_V(\mathcal{S}_H) = \mathcal{S}^{n-1}$, and $\mathcal{K}_V$ and $\mathcal{T}_V$ are inverses of each other on these two spaces.*

Remark 3.4. To obtain a one-to-one mapping between $D \in \mathcal{E}^n$ and $B \in \mathcal{S}^n_+$, one usually adds the centering constraint $Be = 0$. However, this means that B is restricted to a face of $\mathcal{S}^n_+$ and is singular. Therefore, the Slater constraint qualification (strict feasibility) fails for a **SDP** formulation that uses $\mathcal{K}$. Lemma 3.3 shows that we can reduce the problem by projecting onto this face, i.e., we *facially reduce* the problem. The mapping $\mathcal{K}_V$ reduces the dimension of the unknown semidefinite matrix and allows for a one-to-one mapping that also has strictly feasible points, i.e., there exists $\hat{B} \in \mathcal{S}^{n-1}_{++}$ such that $\mathcal{K}_V(\hat{B}) = \hat{D} \in \mathcal{E}^n$ and $\mathcal{T}_V(\hat{D}) = \hat{B}$. This is essential for interior-point methods and for the stability of numerical methods. (See Section 3.2 below.)

This is a first step for facial reduction. We will see below, Section 5.2.3, that we can continue further to reduce the size of the problem and even solve the problem.

We now present several characterizations of $\mathcal{E}^n$. These are used to derive relaxations and algorithms.

Theorem 3.5. *The following characterizations of $D \in \mathcal{E}^n$ hold.*

(1) $D \in \mathcal{S}_H \cap \mathcal{M}^n_+ \cap \{D \in \mathcal{S}^n : v^T e = 0 \Rightarrow v^T D v \leq 0\}$.

(2) $D = \mathcal{K}(B)$, *for some* $B \succeq 0$ *with* $Be = 0$, $B \in \mathcal{S}^n$.

(3) $D = \mathcal{K}_V(B_V)$, *for some* $B_V \in \mathcal{S}^{n-1}_+$.

(4) $D = \mathcal{L}(B) := \begin{bmatrix} 0 & (\operatorname{diag} B)^T \\ \operatorname{diag} B & \mathcal{D}_e(B) - 2B \end{bmatrix}$ *for some* $B \in \mathcal{S}^{n-1}_+$.

(5) $D = \begin{bmatrix} 0 & (\operatorname{diag} B)^T + (s_B e^T - 2x_r^T) \\ \operatorname{diag} B + (s_B e - 2x_r) & \mathcal{D}_e(B) - 2B \end{bmatrix}$ *for some* $B \in \mathcal{S}^{n-1}_+$ *where* $s_B := e^T Be$, $x_r := Be$.

(6) $\mathcal{E}^n = \mathcal{K}(\mathcal{S}^n_+) = \mathcal{K}_V(\mathcal{S}^{n-1}_+)$, $\mathcal{T}_V(\mathcal{E}^n) = \mathcal{S}^{n-1}_+$.

(7) $\mathcal{E}^n = \mathcal{S}_H \cap (\mathcal{S}^{\perp}_C - \mathcal{S}^n_+) = \mathcal{S}_H \cap \big(\mathcal{R}(\mathcal{D}_e) - \mathcal{S}^n_+\big)$.

Proof. (1) Item 1 is the classical characterization of $\mathcal{E}^n$. Proofs are given in [93], [52], [57], [95]. The result also follows from (3.10) and the fact that $\mathcal{T} = \mathcal{K}^{\dagger}$.

(2) The linear transformation $\mathcal{K}$ is the standard transformation used to map between $\mathcal{E}^n$ and $\mathcal{S}^n$. Item 2 follows from the definition of $\mathcal{K}$ given in (3.7).

(3) Item 3 is proved in [2]. Also, it follows from the definition of V and Item 2.

(4) Item 4 is given in [3], [5].

(5) Item 5 is proved in [5]. It also follows from Item 4 since

$$K_V^{\dagger}\left(\begin{bmatrix} 0 & (s_B e^T - 2x_r^T) \\ (s_B e - 2x_r) & 0 \end{bmatrix}\right) = 0.$$

(6) Item 6 is proved in [2] and is also a summary of previous items.

(7) Item 7 is proved in [34]. We include a self-contained proof that uses our tools developed above. First we note that $\operatorname{cone}(E) \lhd \mathcal{S}^n_+$ and $\{E\}^{\perp} = \mathcal{S}_C$. From Lemma 2.1 and Proposition 3.2, we have that

$$(\mathcal{S}_C \cap \mathcal{S}^n)^* = \mathcal{S}_C^* + \mathcal{S}^n = \mathcal{R}(\mathcal{D}_e) + \mathcal{S}^n.$$

Now

$$\begin{aligned} \mathcal{E}^n &= -(\mathcal{S}_C \cap \mathcal{S}^n_+)^* \cap \mathcal{S}_H, && \text{by Item 1,} \\ &= (\mathcal{S}_C^{\perp} - \mathcal{S}^n_+) \cap \mathcal{S}_H \\ &= \big(\mathcal{R}(\mathcal{D}_e) - \mathcal{S}^n_+\big) \cap \mathcal{S}_H, && \text{by Proposition 3.2.} \end{aligned}$$

□

We have emphasized several times that we are using *squared* distances. The advantages are that we get a convex relaxation if we use **EDM** and relax the rank constraint. A distance geometry problem is typically specified by the distances $\sqrt{D_{ij}}$ between nodes $i, j \in V$, for edges $ij \in E$. The solution is the set of points $p_1, \dots, p_n \in \mathbb{R}^r$ that satisfy

$$\|p_i - p_j\|^2 = D_{ij} \qquad \text{for all } ij \in E.$$

In practice, the distances are only known approximately. For example, upper and lower bounds may be given

$$\underline{D}^l_{ij} \le \|p_i - p_j\| \le \bar{D}^u_{ij} \qquad \text{for all } ij \in E.$$

See also [79], [81] where the unsquared distances are used. If the rank constraint is not relaxed, then it is well known that the **FPDG** is NP-hard as it is equivalent to the set partition problem [46].

3.2. Sdp relaxation of the EDMC problem. Given a partial or approximate **EDM** $\bar{D}$, we can find the nearest **EDM** in some norm using (3.5). However, if the embedding dimension is fixed, then this is an NP-hard problem in general; see e.g., [61] for complexity issues related to **EDM**C. This formulation can be relaxed using the characterizations in Theorem 3.5 and not restricting the rank of the optimum matrix Y. We replace the unknown **EDM** $\bar{D}$ using one of the equivalent representations. For example,

$$\min \left\| W \circ \left(\bar{D} - \mathcal{K}(Y)\right) \right\|_F^2 \qquad \text{such that } Y \in \mathcal{S}^n_+, \tag{3.18}$$

where we have chosen the Frobenius norm in the objective function. Since $\operatorname{int} \mathcal{E}^k = \emptyset$ and $\mathcal{K}$ maps one-to-one between $\mathcal{E}^k$ and the face $\mathcal{S}^k_+ \cap \mathcal{S}_C \lhd \mathcal{S}^k_+$, this problem is degenerate, i.e., the optimal set contains the unbounded set $Y^* + \mathcal{N}(\mathcal{K})$, for any optimal solution Y^*. This means that the Slater constraint qualification fails for the dual problem. The following smaller dimensional and more stable problem is derived in [2]. Additional equality or upper and lower bound constraints (in $\bar{D}^u$, and $\underline{D}^l$, respectively) can be added using additional weight matrices W_E, W_{UB} and W_{LB}, respectively:

$$\begin{aligned} &\min \left\| W \circ \left(\bar{D} - \mathcal{K}_V(Y)\right) \right\|_F^2 \qquad \text{such that} \\ &W_{LB} \circ \underline{D}^l \le \begin{cases} W_E \circ \mathcal{K}_V(Y) = W_E \circ \bar{D}, \\ W_{LB} \circ \mathcal{K}_V(Y), \\ W_{UB} \circ \mathcal{K}_V(Y) \le W_{UB} \circ \bar{D}^u, \\ Y \in \mathcal{S}^{k-1}_+. \end{cases} \end{aligned} \tag{3.19}$$

Here $\mathcal{K}_V$ is defined in (3.15), and $B \le C$ denotes $C - B \ge 0$, elementwise. Though we have a convex relaxation of **EDM**C, the approximation is generally poor if the optimal solution has a large rank, see e.g. the estimates in [2], Lemma 2. Reducing the rank is an NP-hard problem and related to compressed sensing [88], [19].

In Section 5.2.3 we derive recent **SDP** relaxations of **SNL** using this approach and show how to easily obtain low rank solutions.

3.3. Applications of FPDG.

The **FPDG** and **EDMs** have a seemingly unlimited number of applications. In this section we present a few of these. It is not our objective here to present an exhaustive list. Rather, we want to demonstrate to the reader the striking variety of interesting applications.

A well-known application is in molecular conformation problems from biology and chemistry. A specific problem of interest is that of determining the structure of a protein given a (partial or complete) set of approximate pairwise distances between the atoms in the protein. Understanding the structure of a protein is key because the structure of a protein specifies its function, and hence its chemical and biological properties.

Distances between atoms in a protein can be approximated theoretically using potential energy minimization, or experimentally using X-ray crystallography or Nuclear Magnetic Resonance (NMR) spectroscopy. The **FPDG** arises via the NMR approach to the problem.

NMR spectroscopy is based on the principle that the nucleus of a hydrogen atom has two spin states. There is a fixed energy separation between the two states, and the spin flips when a particular frequency is attained. If two atoms are sufficiently close, then their spins interact and the frequency at which the spin flip occurs shifts. This causes the peaks in each atom's spectrum to shift as well. Because the intensity of this effect depends on the distance between the two atoms, the NMR analysis is able to estimate the distance between the two atoms. Thus, the outcome of NMR is a set of experimentally estimated distances between the atoms in a molecule. Given such a set of distances, the problem of interest is to deduce the three-dimensional structure of the molecule.

However, the NMR data is inexact and sparse. One of the most important problems in computational biology is the determination of the protein given only the partial inexact **EDM**. This problem is also called the *molecular distance geometry problem*. If the distances between all pairs of atoms in a molecule are known precisely, then the unique corresponding **EDM** D is known. Hence a unique molecular structure can be determined from the points in the rows of the matrix $P \in \mathcal{M}^{nr}$ found using the full rank factorization $B = K^{\dagger}(D) = PP^T$, see Theorem 3.5. However, if only a subset of the distances is known and/or the known distances have experimental errors, then the distances may be inconsistent; and even if they are consistent, the three-dimensional structure may not be unique. The early work in this area is presented in the seminal book of

Crippen and Havel [31]. There has since been huge progress in this area, see [85], [30], [60], [2], [107], [56] and the references therein.

A second application of **EDMs** we highlight is in the fields of anatomy and anthropology. This application is due to the use of so-called landmark data to analyze biological forms, in particular to study the morphological differences in the faces and heads of humans. First, one defines a set of landmarks on the biological structure; for example, the paper [42] uses 16 standardized soft-tissue facial landmarks that include the pronasale (the nasal apex, or "tip of the nose") and the soft-tissue pogonion (the most prominent point on the chin). Second, one obtains coordinates for each of these landmarks on each subject. Of course, what is really of interest is the relative position of each of these landmarks on each subject, so we need a representation that is invariant under translation, rotation, and reflection. The **EDM** representation of this data is ideal for this purpose. Finally, the researchers define various measures to compare two biological structures based on these landmarks. This allows them to quantify phenomena such as the changes in facial geometry due to growth [18], or the normal levels of facial asymmetry in humans [42].

Another application of **EDMs** is in similarity search, a common problem in the areas of databases and expert systems. The problem of similarity search consists of finding the data objects that are most similar to a given query object. This problem is of fundamental importance in applications such as data mining and geographical information systems (GIS). The objective is to carry out similarity search in an automatic manner, i.e., without manual intervention.

An **EDM**-based approach to similarity search was proposed recently in [32]. The gist of this approach is to define a similarity measure between objects. First, each object is represented as a point in a high-dimensional. *feature space*, where the dimensions correspond to features of the objects. A numerical coordinate representation table (NCRT) is defined as a matrix with one row per feature, and one column per object. Then, the similarity between two objects is defined based on the Euclidean distance between their corresponding points in the feature space. It is clear that an **EDM** containing all these distances can be generated using the NCRT.

Computing the similarities between objects is not a static problem, however. This information is then used within some form of automated learning process, and as a consequence of this learning, the information in the similarity matrix is updated. Now we are faced with the problem of ensuring that the resulting matrix remains an **EDM**. Furthermore, the updated NCRT is also of interest. This leads us right to solving an instance of the **FPDG**.

A closely related application is in the area of statistical language modelling, where a problem of interest is to predict the next word in a sentence, given knowledge of the $n-1$ previous words. Given a set of sentences, or corpus, we can determine how many words appear in the corpus. Then we define, for each word, a vector of length equal to the number of words in the corpus, with each entry of the vector containing the probability that the corresponding word follows the word for which the vector is defined. These vectors thus provide a representation of the words in the corpus under consideration.

One problem with this representation is that it is typically extremely large. It is therefore of interest to transform it into a set of vectors in a space of much smaller dimension that captures as much of the information as possible. A popular technique to do this is Principal Component Analysis (PCA). Using **EDMs**, it is actually possible to do much better. Blitzer et al. [16] propose to generate a new set of vectors such that two objectives are attained:

(1) vectors representing semantically similar words should be close to each other (in the sense of vector norm);

(2) vectors representing semantically dissimilar words should be well separated, i.e., the norm of their difference should be large.

The idea in [16] is to pursue both of these objectives via the following **SDP**:

$$\begin{aligned} \max \sum_{ij} D_{ij} \text{ such that } \mathscr{T}_V(D) \succeq 0 D_{ij} = \|p_i - p_j\|^2 \\ \text{for all similar vector pairs } p_i, p_j, \end{aligned} \tag{3.20}$$

where $\mathscr{T}_V$ is given in (3.17). Thus, if p_i and p_j lie within some given (small) neighborhood of each other, then the corresponding element D_{ij} is fixed to their current Euclidean distance. This achieves the first objective above. Simultaneously, the second objective is achieved by maximizing a weighted objective function of the non-fixed D_{ij} entries so that other pairs of words have their vector representations as far apart as possible. A closely related formulation that also preserves the angles between pairs of vectors was presented in [103].

We briefly mention the application of **EDM** to graph realization, **GRL**. Given a simple graph G with vertices $1, 2, \dots, n$ and non-negative edge weights $\{D_{ij} : ij \in E\}$, we call a realization of G in $\mathbb{R}^d$ is any placement of the vertices of G in $\mathbb{R}^d$ such that the Euclidean distance between pairs of vertices $ij \in E$ is given by the weights D_{ij}. If d is fixed, then **GRL** is NP-complete; see Saxe [92] and Aspnes et al. [8]. However, some graph families admit polynomial-time algorithms [10], [11], [12], [20], [20]. Also, Connelly and Sloughter [28] show several characterizations of r-realizable graphs for $r = 1, 2, 3$, including the fact that G is realizable for $r = 3$ if and only if it does not contain K_5 or $K_{2,2,2}$ as a minor. The graph realization problem is discussed in more detail with the **SNL** problem below. We conclude by mentioning again that our list of applications here is by no means extensive. Other applications can be obtained from our references.

4. FPDG and bar framework rigidity

In many applications of **FPDG**, one is interested in determining whether or not a given solution of **FPDG** is either locally unique, unique in the given dimension, or unique in all dimensions. These notions of uniqueness have been extensively studied for bar and tensegrity frameworks under the names rigidity, global rigidity and universal rigidity, respectively. Eren et al. [40] is an excellent paper on the study of network localizations in the context of bar framework rigidity. In this section we survey some of the known results regarding the problems of bar framework rigidity. The problems of tensegrity framework rigidity are beyond the scope of this paper. Hence in the sequel we use the terms "framework" and "bar framework" interchangeably.

A finite collection of points $p_1, \ldots, p_n$ in $\mathbb{R}^r$ which span $\mathbb{R}^r$ is called an *r-configuration p*. (Note that the absence of subscript on p indicates that we are referring to the entire configuration, while subscripts indicate the individual points in the configuration.) Let $G = (V, E)$ be a simple graph on the vertices $1, 2, \ldots, n$. A *bar framework*, denoted by $G(p)$, in $\mathbb{R}^r$, consists of a graph G together with an r-configuration p, where each vertex i of G is located at p_i. To avoid trivialities, we assume that G is not a complete graph.

Two frameworks $G(p)$ in $\mathbb{R}^r$ and $G(q)$ in $\mathbb{R}^s$ are said to be *equivalent* if $\|q_i - q_j\| = \|p_i - p_j\|$ for all $(i, j) \in E$, where $\|.\|$ denotes the Euclidean norm. The term bar is used to describe such frameworks because in any two equivalent frameworks $G(p)$ and $G(q)$, every two adjacent vertices i and j must stay the same distance apart. Thus edges of G can be thought of as stiff bars and the nodes of G can be thought of as universal joints. See Figure 1 for an example of 3 bar frameworks in the plane. Nodes (joints) of the framework are represented by little circles, while the edges (bars) are represented by line segments.

Two frameworks $G(p)$ and $G(q)$ in $\mathbb{R}^r$ are said to be *congruent* if $\|q_i - q_j\| = \|p_i - p_j\|$ for all $i, j = 1, \ldots, n$. That is, $G(p)$ and $G(q)$ are congruent

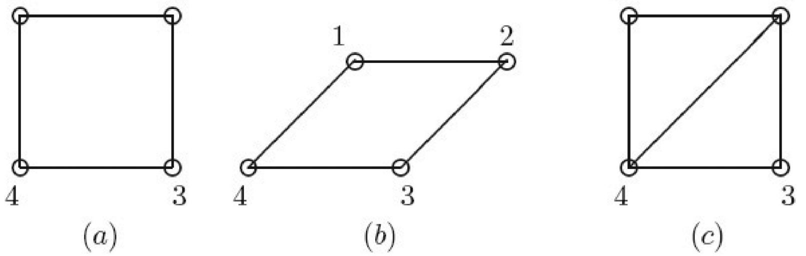

Figure 1. An example of three bar frameworks in $\mathbb{R}^2$. Frameworks (a) and (b) are equivalent and flexible; while framework (c) is rigid.

if r-configuration q can be obtained from r-configuration p by applying a rigid motion such as a translation or a rotation in $\mathbb{R}^r$. In this section we do not distinguish between congruent frameworks.

A framework $G(p)$ in $\mathbb{R}^r$ is said to be *generic* if all the coordinates of $p_1, \dots, p_n$ are algebraically independent over the integers. That is, $G(p)$ is generic if there does not exist a non-zero polynomial f of the components of the p_is with integer coefficients such that

$$f\big((p_1)_1, \dots, (p_1)_r, \dots, (p_n)_1, \dots, (p_n)_r\big) = 0.$$

We begin first by presenting some known results on bar framework rigidity or local uniqueness.

4.1. Bar framework rigidity. A framework $G(p)$ in $\mathbb{R}^r$ is said to be *rigid* (or locally unique) if for some $\varepsilon > 0$, there does not exist any framework $G(q)$ in $\mathbb{R}^r$, which is equivalent to $G(p)$, such that $\|q_i - p_i\| \le \varepsilon$ for all $i = 1, \dots, n$. Recall that we do not distinguish between congruent frameworks. If a framework is not rigid we say it is *flexible*. For other equivalent definitions of rigidity, and consequently of flexibility, see [47].

Given a framework $G(p)$, consider the following system of equations:

$$(p_i - p_j)^T(\bar{p}_i - \bar{p}_j) = 0 \qquad \text{for all } (i,j) \in E. \tag{4.21}$$

Any $\bar{p} = (\bar{p}_1, \dots, \bar{p}_n)$ that satisfies (4.21) is called an *infinitesimal flex* of $G(p)$. We say that an infinitesimal flex is *trivial* if it results from a rigid motion of $G(p)$. A framework $G(p)$ is said to be *infinitesimally rigid* if it has only trivial infinitesimal flexes. Otherwise, $G(p)$ is said to be *infinitesimally flexible* [25], [23], [31], [54], [104].

As the following theorem shows, the notion of infinitesimal rigidity of a framework is stronger than that of rigidity.

Theorem 4.1 ([47]). *If a bar framework $G(p)$ is infinitesimally rigid, then it is rigid.*

The converse of Theorem 4.1 is false. However, Asimow and Roth [6] showed that the notions of rigidity and infinitesimal rigidity coincide for generic bar frameworks.

It is well known [47], [7] that bar framework rigidity is a generic property, i.e., if a generic framework $G(p)$ in $\mathbb{R}^r$ is rigid, then all generic frameworks $G(q)$ in $\mathbb{R}^r$ having the same underlying graph G are also rigid.

Given a framework $G(p)$ in $\mathbb{R}^r$ with n vertices and m edges, let R be the $m \times nr$ matrix whose rows and columns are indexed, respectively, by the edges and the vertices of G such that for each $(i, j) \in E$, the (i, j)th row of R is given by

$$[0 \;\cdots\; 0 \;\; \overbrace{(p_i - p_j)^T}^{\text{vertex } i} \;\; 0 \;\cdots\; 0 \;\; \overbrace{(p_j - p_i)^T}^{\text{vertex } j} \;\; 0 \;\cdots\; 0]. \tag{4.22}$$

R is called the *rigidity matrix* of $G(p)$ and obviously, the space of infinitesimal flexes of a framework is the nullspace of its rigidity matrix R, i.e., an infinitesimal flex of $G(p)$ is just a linear dependency among the columns of R.

Theorem 4.2 ([6]). *Let R be the rigidity matrix of a generic bar framework $G(p)$ of n vertices in $\mathbb{R}^r$. Then $G(p)$ is rigid if and only if*

$$\operatorname{rank} R = nr - \frac{r(r+1)}{2}. \tag{4.23}$$

Therefore, the rigidity of a generic bar framework can be efficiently determined via randomized algorithms [76]. Next we consider the problem of combinatorial characterization of generic bar framework rigidity.

Let $G(p)$ be a generic bar framework in $\mathbb{R}^1$. Then obviously, $G(p)$ is rigid if and only if G is connected. For generic bar frameworks in the plane we have the following theorem.

Theorem 4.3 ([74], [78]). *Let $G(p)$ be a generic bar framework on n vertices in $\mathbb{R}^2$ $(n \geq 2)$, then $G(p)$ is rigid if and only if*

$$2n - 3 \leq \sum_{i=1}^{k} (2|V_{E_i}| - 3),$$

for every partition of the edge set E of G into nonempty subsets $E_1, \ldots, E_k$, where V_{E_i} denotes the set of nodes incident to some edge in E_i.

Thus generic bar framework rigidity in $\mathbb{R}^2$ can also be determined in polynomial time [44], [55], [76]. Obtaining a combinatorial characterization of generic bar framework rigidity in dimension 3 or higher is still an open problem.

4.2. Bar framework global rigidity. A framework $G(p)$ in $\mathbb{R}^r$ is said to be *globally rigid* if there does not exist a framework $G(q)$ in the same space $\mathbb{R}^r$ which is equivalent to $G(p)$. Recall that we do not distinguish between congruent frameworks. Obviously, rigidity is a necessary, albeit not sufficient, condition for global rigidity of a framework. Framework (c) in Figure 1 is rigid but not globally rigid.

A graph G is said to be k *vertex-connected* if G remains connected after deleting fewer than k of its vertices. A bar framework $G(p)$ is said to be *redundantly rigid* if $G(p)$ remains rigid after deleting any one edge of G. Recently, the problem of global rigidity of bar frameworks has received a great deal of attention [27], [40], [62], [63]. Hendrickson [59], [60] proved that if a generic framework $G(p)$ in $\mathbb{R}^r$ with at least $r+1$ vertices is globally rigid, then the graph $G=(V,E)$ is $r+1$ vertex-connected and $G(p)$ is redundantly rigid. Hendrickson also conjectured that $r+1$ vertex-connectivity of G and redundant rigidity of $G(p)$ are sufficient for global rigidity of a generic framework $G(p)$. This conjecture, which is obviously true for $r=1$, was shown by Connelly [24] to be false for $r \geq 3$.

Jackson and Jordán [62] proved that Hendrickson's conjecture is true for $r=2$.

Theorem 4.4 ([62], [59]). *Given a generic bar framework $G(p)$ in $\mathbb{R}^2$, then $G(p)$ is globally rigid in $\mathbb{R}^2$ if and only if G is either a complete graph on at most three vertices or G is* 3*-vertex-connected and redundantly rigid.*

Let $G(p)$ be a framework in $\mathbb{R}^r$ where G has n vertices and m edges. Associate with each edge (i,j) of G a scalar ω_{ij}. The vector $\omega=(\omega_{ij})$ in $\mathbb{R}^m$ such that

$$\sum_j \omega_{ij}(p_i - p_j) = 0 \qquad \text{for all } i = 1, \ldots, n, \tag{4.24}$$

is called an *equilibrium stress* for $G(p)$. Note that if ω is an equilibrium stress for $G(p)$ then ω belongs to the left null space of R, the rigidity matrix of $G(p)$, i.e., $R^T\omega = 0$. Given an equilibrium stress ω, let $S=(s_{ij})$ be the $n \times n$ symmetric matrix such that

$$s_{ij} = \begin{cases} -\omega_{ij} & \text{if } (i,j) \in E, \\ 0 & \text{if } (i,j) \notin E, \\ \sum_{k:(i,k)\in E} \omega_{ik} & \text{if } i = j. \end{cases}$$

S is called the *stress matrix* associated with ω. Connelly [27] gave a sufficient condition, in terms of S, for a generic framework $G(p)$ in $\mathbb{R}^r$ to be globally rigid.

Theorem 4.5 ([27]). *Let $G(p)$ be a given generic bar framework with n vertices in $\mathbb{R}^r$; and let S be the stress matrix associated with an equilibrium stress ω for $G(p)$ such that rank $S = n-1-r$. Then $G(p)$ is globally rigid in $\mathbb{R}^r$.*

Connelly also conjectured that the above sufficient condition is also necessary. This conjecture was later proved to be true by Gortler et al.

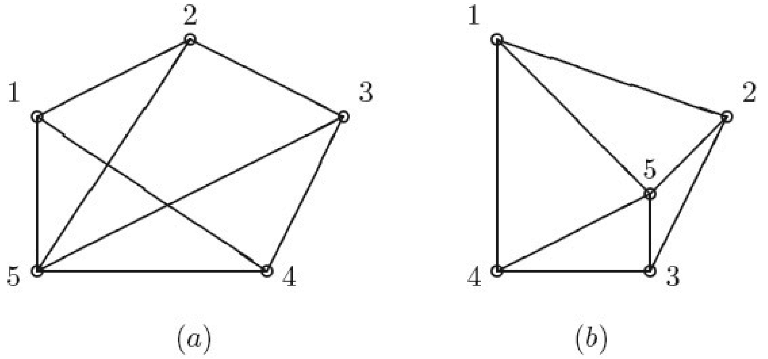

Figure 2. An example of two frameworks in $\mathbb{R}^2$. It can be shown that the framework in (a) is universally rigid while the framework in (b) is globally rigid but not universally rigid.

Theorem 4.6 ([27], [49]). *Let $G(p)$ be a given generic framework with n vertices in $\mathbb{R}^r$. Then $G(p)$ is globally rigid in $\mathbb{R}^r$ if and only if there exists a stress matrix S associated with an equilibrium stress ω for $G(p)$ such that rank $S = n - 1 - r$.*

4.3. Bar framework universal rigidity. A framework $G(p)$ in $\mathbb{R}^r$ is said to be *universally rigid* if there does not exist a framework $G(q)$ in $\mathbb{R}^s$, for any s, $1 \leq s \leq n-1$, which is equivalent to $G(p)$. It immediately follows that universal rigidity implies global rigidity but the converse is not true. The framework (b) in Figure 2 is globally rigid in $\mathbb{R}^2$ but it is not universally rigid.

Alfakih [1] presented a sufficient condition for generic universal rigidity of bar frameworks and conjectured that this condition is also necessary. This condition is given in terms of the Gale matrix Z of the configuration p (see p. 16). As it turns out, the condition can also be equivalently given in terms of the stress matrix S since Z and S are closely related as will be shown at the end of this section.

Let $G(p)$ be a given framework with n vertices in $\mathbb{R}^r$ and let e denote the vector of all 1's in $\mathbb{R}^n$. Consider the $(r+1) \times n$ matrix

$$\begin{bmatrix} P^T \\ e^T \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & \cdots & p_n \\ 1 & 1 & \cdots & 1 \end{bmatrix}.$$

Recall that $p_1, \ldots, p_n$ are not contained in a proper hyperplane in $\mathbb{R}^r$, i.e., the affine space spanned by $p_1, \ldots, p_n$ has dimension r. Then $r \leq n-1$, and the matrix $\begin{bmatrix} P^T \\ e^T \end{bmatrix}$ has full row rank. Let $\bar{r} = n - 1 - r$ and for $\bar{r} \geq 1$, let Λ be the $n \times \bar{r}$ matrix whose columns form a basis for the nullspace of $\begin{bmatrix} P^T \\ e^T \end{bmatrix}$. Λ is called a *Gale matrix* corresponding to P; and the ith row of Λ, considered as a vector in $\mathbb{R}^{\bar{r}}$, is called a *Gale transform* of p_i [45]. The Gale transform plays an important role in the theory of polytopes. We take advantage of the fact that Λ is not unique to define a special sparse Gale matrix Z which is also more convenient for our purposes.

Let us write Λ in block form as

$$\Lambda = \begin{bmatrix} \Lambda_1 \\ \Lambda_2 \end{bmatrix},$$

where Λ_1 is $\bar{r} \times \bar{r}$ and Λ_2 is $(r+1) \times \bar{r}$. Since Λ has full column rank, we can assume without loss of generality that Λ_1 is non-singular. Then Z is defined as

$$Z := \Lambda\Lambda_1^{-1} = \begin{bmatrix} I_{\bar{r}} \\ \Lambda_2\Lambda_1^{-1} \end{bmatrix}. \tag{4.25}$$

Let ${z^i}^T$ denote the ith row of Z then it readily follows that $z^1, z^2, \ldots, z^{\bar{r}}$, the Gale transforms of $p_1, p_2, \ldots, p_{\bar{r}}$ respectively, are simply the standard unit vectors in $\mathbb{R}^{\bar{r}}$.

Theorem 4.7 ([26], [1], [50]). *Let $G(p)$ be a generic bar framework with n vertices in $\mathbb{R}^r$ for some $r \leq n-2$, and let Z be the Gale matrix corresponding to $G(p)$. Recall that $\bar{r} = n-1-r$. Then $G(p)$ is universally rigid if and only if there exists a $\bar{r} \times \bar{r}$ symmetric positive definite matrix*

$$\Psi : {z^i}^T \Psi z^j = 0 \qquad \textit{for all } (i,j) \notin E, \tag{4.26}$$

where ${z^i}^T$ is the ith row of Z.

Two remarks are in order here. First, the *if* part of Theorem 4.7 was proved independently in [26] and [1], while the *only if* part was conjectured in [1] and proved in [50]. Second, the statement of Theorem 4.7 in [26], [50] was given in terms of the stress matrix S of framework $G(p)$. (See Theorem 4.11 in Section 4.4, below.)

4.4. Gale matrices and stress matrices. As we mentioned earlier, the Stress matrix S of a bar framework $G(p)$ is closely related to the Gale matrix Z corresponding to $G(p)$.

Lemma 4.8 ([1]). *Given a framework $G(p)$ with n vertices in $\mathbb{R}^r$, let Z be the Gale matrix corresponding to $G(p)$ and recall that $\bar{r} = n-1-r$. Further, let S be the stress matrix associated with an equilibrium stress ω for $G(p)$. Then*

$$S = Z\Psi Z^T \qquad \textit{for some } \bar{r} \times \bar{r} \textit{ symmetric matrix } \Psi. \tag{4.27}$$

Furthermore, let ${z^i}^T$ be the ith row of Z. If Ψ' is any $\bar{r} \times \bar{r}$ symmetric matrix such that ${z^i}^T \Psi' z^j = 0$ for all $(i,j) \notin E$, then $Z\Psi' Z^T$ is a stress matrix associated with an equilibrium stress ω for $G(p)$.

The following corollary obtained by Connelly follows immediately from the previous lemma.

Corollary 4.9 (Connelly [23]). *Let S be the stress matrix associated with an equilibrium stress ω for framework $G(p)$ with n vertices in $\mathbb{R}^r$, then*

$$\operatorname{rank} S \leq \bar{r} = n - 1 - r. \tag{4.28}$$

In light of Lemma 4.8, we can express the sufficient conditions for global rigidity and for universal rigidity of a bar framework in terms of either the stress matrix S or the Gale matrix Z. Thus Theorems 4.6 and 4.7 can be stated equivalently as follows:

Theorem 4.10. *Let $G(p)$ be a given generic framework $G(p)$ with n vertices in $\mathbb{R}^r$ for some $r \leq n-2$, and let Z be the Gale matrix corresponding to $G(p)$. Recall that $\bar{r} = n - 1 - r$. Then $G(p)$ is globally rigid in $\mathbb{R}^r$ if and only if there is a $\bar{r} \times \bar{r}$ symmetric non-singular matrix*

$$\Psi : {z^i}^T \Psi z^j = 0 \qquad \textit{for all } (i,j) \notin E, \tag{4.29}$$

where ${z^i}^T$ is the ith row of Z.

Theorem 4.11. *Let $G(p)$ be a generic framework with n vertices in $\mathbb{R}^r$. Then $G(p)$ is universally rigid if and only if there exists a positive semi-definite stress matrix S associated with an equilibrium stress ω for $G(p)$ such that* $\operatorname{rank} S = \bar{r} = n - 1 - r$.

The survey so far has focused on theoretical results concerning **EDMs**. The next section is concerned with computational algorithms for a particular type of problem that can be modelled and solved using **EDMs**, namely **SNL**.

5. Algorithms specific to SNL

One goal in this survey is to show that **EDM** is an elegant and powerful tool for looking at **FPDG** problems. There are many advantages to using the well studied linear operators $\mathcal{K}$, $\mathcal{T}$, see e.g. Proposition 3.2. Many algorithms for **EDM** can be applied to **FPDG** problems and, in particular, to the active area of research of **SNL**, the problem outlined in Section 1. Wireless sensor networks have many applications such as monitoring physical or environmental conditions (temperature, moisture, sound, vibration, pressure, battlefield surveillance, etc.), home automation, hospital patients, traffic control, etc. They are often referred to as *smart dust* as they can be used to cover large areas such farmland or chemical plant explosion sites.

> "Untethered micro sensors will go anywhere and measure anything—traffic flow, water level, number of people walking by, temperature. This is developing into something like a nervous system for the earth, *a skin for the earth*. The world will evolve this way." (See 21 Ideas for the 21st Century, Business Week. 8/23–30, 1999)

This research area has several workshops and conferences each year, e.g. MELT 2008, and dedicated publications, e.g. International Journal of Sensor Networks. Recent related theses and books include [58], [89], [33], [22], [64], [67], [21], [83], [100]. Research groups include CENS at UCLA and WEBS at UC-Berkeley. The algorithmic side has advanced quickly. From solving problems with $n = 50$ sensors with low accuracy, in the case of exact distances or low level of noise, current codes can quickly solve problems with 100,000s of sensors to high accuracy:

http://www.math.nus.edu.sg/~mattohkc/SNLSDP.html
http://www.math.uwaterloo.ca/~ngbkrisl/Publications_files/SNLSDPclique_ver01.tar

However, the performance in terms of accuracy of most of the methods quickly deteriorates if the noise level becomes significant. This is the actual research direction, and computational studies show the improvements achieved by some of the most recent methods [68], [69], [72].

The algorithms for **SNL** often use minor modifications that identify anchors with sensors. In fact, a set of anchors simply corresponds to a given fixed clique for the graph of the **EDM** problem [71], [36], [35], [73]. It can be advantageous to delay using the difference between anchors and sensors and instead solve the resulting **EDM** problem. Then, starting from the obtained solution, a best rank-r approximation is found. Finally, in order to get the sensors positioned correctly, the approximation is rotated to get the anchors (approximately) back into their original positions. In fact, it is shown in [73] that it is advantageous to also delay completing the distances; see Section 5.2.3 below.

In the literature there are many algorithms that are specific to **SNL** and are not based on **EDM**. In these algorithms, the presence of the anchors plays a fundamental role, and in some of them their position influences the quality of the solutions obtained. In addition, a significant property that makes **SNL** unique from other **FPDG** problems is its distributed nature, i.e., even for many anchor free problems, distances between sensors are known *only locally*.

The **SNL** problem presents three main difficulties. It is a nonconvex problem, and in real applications it requires the localization of a large number of sensors where, in addition, the measured distances are noisy. Therefore, the algorithms proposed in the literature on the one hand introduce convex relaxations of **SNL**, where the constraints are linear, semidefinite, conic, or polynomial; and, on the other hand they define *distributed*, rather than centralized, approaches to handle the large size of problems arising from real networks. And, finally, they try to find a *nearest* realization of the points using a measure related to a reasonable error model.

Historically, [37] is one of the early papers based on solving a convex relaxation of **SNL**. In particular, the authors use convex (SDP) constraints to model the constraints for the proximity between sensors (nodes) that are within radio range. Let $x_i, x_j \in \mathbb{R}^r$ be two sensors that communicate so that their distance apart is available, i.e., they must be within the radio range R. Then the **SDP** constraint

$$\|x_i - x_j\| \leq R \iff \begin{pmatrix} RI_r & x_i - x_j \\ (x_i - x_j)^T & R \end{pmatrix} \succeq 0 \tag{5.30}$$

must hold. As an alternative, the true distance between the two sensors may be used if available.

A different convex constraint is obtained by considering information on the angles between transmitters in the case of sensor nodes with laser transmitters and receivers that scan through some angle. The receiver first rotates its detector coarsely, until it gets a signal; and then it rotates finely to get the maximum strength signal. The angle at which the best signal is obtained provides an estimate of the angle to the transmitter and a vague estimate of the maximum distance between receiver and transmitter. This results in three linear, **LP**, constraints: two to bound the angle; and another one to bound the distance. Any combination of the **SDP** and **LP** constraints for each sensor can be used in principle to get an approximate location of the nodes. In [37], the authors consider separately the problem obtained by including only the radio range constraints, and then the problem obtained by considering only the angle derived **LP** constraints. The first set of constraints (5.30) can be solved using a second order cone programming solver, the other set uses an **LP** solver. A linear objective function is introduced and its choice is exploited in order to bound the feasible set with a rectangle parallel to the axes. In the computational tests, the network is solved many times, each time adding an anchor, until a maximum number of anchors is reached. The performance is evaluated by using the mean error from the real positions. The results show that this approach is influenced by the position of the anchors; indeed, the performance improves if the anchors are on the boundary of the feasible set, i.e., when all the localized sensors are within the convex hull of the anchors.

The importance of [37] also lies in providing the first distributive approach and in introducing the idea of dividing a large network into smaller subnetworks on the basis of connectivity information. Other papers that use a distributed approach for **SNL** include [64], [20], [90], [91]. This idea has been exploited and further developed by Ye and his coauthors in [14], [10], [11], [12], [94], [102]. Their approach is termed the Biswas–Ye (**B–Y**) **SDP** relaxation and is used as well in [86], [68], [69] among others. The above methods use localization near anchors. A distributed approach based on a natural division using just cliques and independent of the anchors is given in [73], see Section 5.2.3.

5.1. Biswas–Ye SDP relaxation, EDMC, and facial reduction.

The **B–YSDP** relaxation of **SNL** (see the discussion in Section 5 above and (5.35) below) is used in many algorithms for solving **SNL** problems. Therefore, it is of interest to understand its relationship with the classical relaxations based on **EDMC**. The **B–Y** relaxation can be derived directly from the definitions [13]. Alternatively, we can use the approach in [71], [36], [73] and derive this relaxation from the **EDM** framework. In fact, we now show that the **B–Y** relaxation can also be obtained as a *restricted* second step in facial reduction for the **EDM** relaxation, following on the one for centering in Remark 3.4. This second step is based on the fact that the anchors form a clique in the graph of the **SNL** (corresponding to a principal submatrix in the **EDM** D) with given embedding dimension r. Therefore, the corresponding principal submatrix of $\mathcal{K}^\dagger(D)$ has rank restricted to at most $r+1$. Lemma 5.2 and Remark 5.3, below, provide the details as well as a comparison between the **B–Y** relaxation and **EDMC**.

If we ignore the anchors (and temporarily ignore the upper and lower bounds) we can use the relaxation in (3.19), where the given approximate (incomplete) **EDM** $\bar{D}$ is approximated by $\mathcal{K}_V(Y) = \mathcal{K}(VYV^T)$, $Y \in \mathcal{S}_+^{n-1}$. However, we have an additional constraint to make use of, i.e., we know the distances for the clique of anchors. This allows for a *facial reduction* of **SNL**. We first give the basic result for facial reduction for **EDMC**.

Theorem 5.1 ([36], [73]). *Let $D \in \mathcal{E}^n$, with embedding dimension r. Suppose that $D[1:k] \in \mathcal{E}^k$ has embedding dimension t; and let $B := \mathcal{K}^\dagger(D[1:k]) = \bar{U}_B S \bar{U}_B^T$, where $\bar{U}_B \in \mathcal{M}^{k\times t}$, $\bar{U}_B^T \bar{U}_B = I_t$, and $S \in \mathcal{S}_{++}^t$. Furthermore, let $U_B := \begin{bmatrix} \bar{U}_B & \frac{1}{\sqrt{k}}e \end{bmatrix} \in \mathcal{M}^{k\times(t+1)}$, $U := \begin{bmatrix} U_B & 0 \\ 0 & I_{n-k} \end{bmatrix}$, and let $\begin{bmatrix} V & \frac{U^Te}{\|U^Te\|} \end{bmatrix} \in \mathcal{M}^{n-k+t+1}$ be orthogonal. Then*

$$\begin{aligned} \operatorname{face} \mathcal{K}^\dagger\big(\mathcal{E}^n(1:k, D[1:k])\big) &= (U\mathcal{S}_+^{n-k+t+1}U^T) \cap \mathcal{S}_C \\ &= (UV)\mathcal{S}_+^{n-k+t}(UV)^T. \end{aligned} \tag{5.31}$$

Theorem 5.1 shows that if we know the distances for a clique of cardinality k with embedding dimension t, then we can reduce the size of the matrix variable in the **SDP** representation of the **EDM** from n to $n-k+t$. Now suppose that we are given an **SNL** problem, i.e., we are given the position of the anchors a_j, $j = 1, \dots, m$, and a partial **EDM** $\bar{D}$, i.e., some of the elements are unknown, and, for pairs of indices in two given index sets N_a, N_x, we know the exact squared Euclidean distance values: the anchor-sensor values $\bar{D}_{ij}$ between a_i and x_j for

$(i, j) \in N_a$ and the sensor-sensor values $\bar{D}_{ij}$ between x_i and x_j for $(i, j) \in N_x$. We wish to find a realization of $x_1, \ldots, x_{n-m} \in \mathbb{R}^r$ such that

$$\begin{aligned} \|a_k - x_j\|^2 &= \bar{D}_{kj} \quad \text{for all } (k, j) \in N_a, \\ \|x_i - x_j\|^2 &= \bar{D}_{ij} \quad \text{for all } (i, j) \in N_x. \end{aligned} \tag{5.32}$$

Furthermore, there exist lower and upper bounds on some of the unknown distances between sensors and between sensors and anchors, i.e., lower bounds $\underline{r}_{kj}$ for anchor-sensors $(k, j) \in L_a$, lower bounds $\underline{r}_{ij}$ for sensor-sensors $(i, j) \in L_x$; and, upper bounds $\bar{r}_{kj}$ for anchor-sensors $(k, j) \in U_a$, and upper bounds $\bar{r}_{ij}$ for sensor-sensors $(i, j) \in U_x$. The model becomes

$$\begin{aligned} \|a_k - x_j\|^2 &= \bar{D}_{kj} \quad \text{for all } (k, j) \in N_a, \\ \|x_i - x_j\|^2 &= \bar{D}_{ij} \quad \text{for all } (i, j) \in N_x, \\ \|a_k - x_j\|^2 &\geq \underline{r}_{kj} \quad \text{for all } (k, j) \in L_a, \\ \|x_i - x_j\|^2 &\geq \underline{r}_{ij} \quad \text{for all } (i, j) \in L_x, \\ \|a_k - x_j\|^2 &\leq \bar{r}_{kj} \quad \text{for all } (k, j) \in U_a, \\ \|x_i - x_j\|^2 &\leq \bar{r}_{ij} \quad \text{for all } (i, j) \in U_x. \end{aligned} \tag{5.33}$$

Recall the description of the **SNL** problem in Section 2. The matrix P of nodes is partitioned as $P = \left[\begin{smallmatrix} A \\ X \end{smallmatrix}\right]$, where the position of the anchors $p_i = a_i$, $i = 1, \ldots, m$, are the columns of $A^T \in \mathcal{M}^{rm}$; and the unknown positions of the sensors $p_{m+i} = x_i$, $i = 1, \ldots, m - n$, are the columns of $X^T \in \mathcal{M}^{r(n-m)}$.

Note that the two terms $\|a_k - x_j\|^2$ and $\|x_i - x_j\|^2$ in (5.32) can be expressed as

$$\begin{aligned} \|a_k - x_j\|^2 &= (a_k^T - e_j^T)\begin{pmatrix} I_r & X^T \\ X & XX^T \end{pmatrix}\begin{pmatrix} a_k \\ -e_j \end{pmatrix}, \\ \|x_i - x_j\|^2 &= (e_i - e_j)^T XX^T (e_i - e_j). \end{aligned} \tag{5.34}$$

In Biswas–Ye [13], problem (5.32) is modelled using the equivalent (5.34) and is relaxed to the following **SDP** feasibility problem: find a symmetric matrix $Z \in \mathcal{S}^{n-m+r}$ such that

$$\begin{aligned} &(a_k^T - e_j^T)Z\begin{pmatrix} a_k \\ -e_j \end{pmatrix} = \bar{D}_{kj} \quad \text{for all } (k, j) \in N_a, \\ &(e_i - e_j)^T Y (e_i - e_j) = \bar{D}_{ij} \quad \text{for all } (i, j) \in N_x, \\ &Z = \begin{pmatrix} I_r & X^T \\ X & Y \end{pmatrix} \in \mathcal{S}_+^{n-m+r}. \end{aligned} \tag{5.35}$$

We emphasize that this **SDP** solves a **EDMC** problem, but it fixes the positions of the anchors explicitly. This **SDP** relaxes the equality $Y = XX^T$ to $Y \succeq XX^T$; equivalently, relaxing to $Z = \begin{bmatrix} I & X^T \\ X & Y \end{bmatrix} \succeq 0$. The rows of the X part of the resulting Z are used as the approximation of the positions of the sensors.

Note that the original P satisfies

$$\begin{aligned} 0 \preceq PP^T &= \begin{bmatrix} AA^T & AX^T \\ XA^T & XX^T \end{bmatrix} \\ &= \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & X^T \\ X & Y \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix}^T \quad \text{with } Y = XX^T. \end{aligned} \tag{5.36}$$

However, if the Y part of the Z found in (5.35) has rank larger than the embedding dimension r, then Z cannot be factored as $\begin{bmatrix} I \\ X \end{bmatrix}\begin{bmatrix} I \\ X \end{bmatrix}^T$. Therefore, it is not clear that the rows of X yield the best approximation for the localization of the sensors. For example, a better approximation might be to use the spectral decomposition of the right-handside in (5.36), i.e., the spectral decomposition of $\begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix} Z \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix}^T$. One can choose the r eigenvectors v_i corresponding to the largest r eigenvalues λ_i to form the approximation $\bar{P} = [v_1 \ \cdots \ v_r] \operatorname{Diag}(\lambda)$. In addition, it may be better not to fix the I part of Z, i.e., it may be better to allow the anchors to move during the approximation process. (We amplify on this below.)

Now let

$$U_A \in \mathscr{M}^{m \times r} \quad \text{and} \quad R \in \mathscr{M}^r \text{ satisfy } \mathscr{R}(U_A) = \mathscr{R}(A),\ U_A = AR^{-1}. \tag{5.37}$$

Define the linear transformation $\mathscr{K}_{U_A}(Z) : \mathscr{S}^{n-m+r} \to \mathscr{S}^n$ by

$$\mathscr{K}_{U_A}(Z) := \mathscr{K}\left(\begin{bmatrix} U_A & 0 \\ 0 & I_{n-m} \end{bmatrix} Z \begin{bmatrix} U_A & 0 \\ 0 & I_{n-m} \end{bmatrix}^T \right). \tag{5.38}$$

We can define the weight and bound matrices in (3.19) to coincide with the index sets and bounds in (5.33). We now combine (5.36) with Theorem 5.1. This yields the following comparison of the feasible sets in the **B–Y** and **EDM** relaxations.

Lemma 5.2. *Define the nonnegative weight matrix* $0 \leq W_E \in \mathscr{M}^n$ *by*

$$(W_E)_{ij} := \begin{cases} 1 & \textit{if } ij \in N_a \cup N_x, \\ 0 & \textit{otherwise}, \end{cases}$$

where N_a, N_x *are defined as in* (5.35). *Similarly, define the lower and upper bound weight matrices* W_{LB}, W_{UB}. *Let* U_A *be defined as in* (5.37). *Define the feasible sets*

$$\mathscr{F}_{U_A}^{\mathrm{EDM}} := \left\{ Z : \begin{array}{r} W_E \circ \mathscr{K}_{U_A}(Z) = W_E \circ D_E \\ W_{LB} \circ \underline{D}^l \leq W_{LB} \circ \mathscr{K}_{U_A}(Z) \\ W_{UB} \circ \mathscr{K}_{U_A}(Z) \leq W_{UB} \circ \bar{D}^u \\ Z \in \mathscr{S}_+^{n-m+r} \end{array} \right\} \tag{5.39}$$

and

$$\mathscr{F}_{A}^{\mathrm{BY}} := \left\{ Z : \begin{array}{r} W_E \circ \mathscr{K}_{A}(Z) = W_E \circ D_E \\ W_{LB} \circ \underline{D}^l \leq W_{LB} \circ \mathscr{K}_{A}(Z) \\ W_{UB} \circ \mathscr{K}_{A}(Z) \leq W_{UB} \circ \bar{D}^u \\ Z = \begin{bmatrix} I & X^T \\ X & Y \end{bmatrix} \in \mathscr{S}_+^{n-m+r} \end{array} \right\}. \tag{5.40}$$

Then the feasible sets $\mathscr{F}_{U_A}^{\mathrm{EDM}}$ *and* $\mathscr{F}_{A}^{\mathrm{BY}}$ *correspond to the* ***EDM*** *and* ***B–Y*** *relaxation, respectively. Moreover,*

$$Z \in \mathscr{F}_{A}^{\mathrm{BY}} \implies \begin{bmatrix} R & 0 \\ 0 & I \end{bmatrix} Z \begin{bmatrix} R & 0 \\ 0 & I \end{bmatrix}^T \in \mathscr{F}_{U_A}^{\mathrm{EDM}}, \tag{5.41}$$

where R is as defined in (5.37).

Proof. That $\mathscr{F}_{U_A}^{\mathrm{EDM}}$ corresponds to the **SDP** relaxation follows from the facial reduction in Theorem 5.1. That $\mathscr{F}_{A}^{\mathrm{BY}}$ is the **B–Y** relaxation follows upon expanding the terms.

The inclusion in (5.41) follows upon expanding the right-hand side. □

Remark 5.3. Lemma (5.2) illustrates the benefits and drawbacks of the two relaxations.

For both relaxations, the quality of the relaxation results from considering the quality of the approximation $Y \approx XX^T$; see the discussion in Section 5.1.2. Therefore, if we replace the objective functions in Lemma 5.2 with the convex function $\mathrm{trace}(Z_Y - Z_X Z_X^T)$, where Z_Y, Z_X are the appropriate blocks of the unknown matrix Z, then we get a comparison of the strength of the relaxations in the case that the weight matrix $W = 0$, i.e., in the case that only exact distances are considered.

If we choose an appropriate objective value based on minimizing an appropriate error model, then the first relaxation using **EDMC** provides a better solution for the objective value, i.e., it is a better least squares approximation. However, the optimum may have a large rank and the rank r approximation may result in a poor approximation. The Biswas–Ye relaxation fixes the upper r dimensional block of Z to I. This has the effect of fixing the anchors. (Since typically

$r \in \{2, 3\}$ this reduction in variables is small.) The optimum in the Biswas–Ye relaxation immediately yields an approximation X^*_{B-Y} for the sensors with the correct rank. There is no need to find a best rank-r approximation or the rotation Q. However, restricting this rank during the relaxation may result in a larger objective value.

The tests in [36] show empirically that the relaxation using **EDMC** is usually better on randomly generated problems, i.e., treating the anchors as sensors in the relaxation, using a best rank-r approximation and then rotating the sensors back so the anchors are as close as possible to their original position generally provides a better estimate for the sensors, compared to fixing the anchors throughout the relaxation.

5.1.1. Unique localizability. The notion of *localizability* is discussed in [40], [83]. In contrast to using the **EDMC** approach outlined in Section 5.1 and Lemma 5.2, localizability is based on finding the location of a sensor using neighboring anchors, i.e., specifically concentrating on the properties of the anchors. Once a sensor's location is found, it becomes an anchor. Results in [40] provide conditions that guarantee that all the sensors can be localized and also discuss the expense/time. (This localizability is related to the geometric build-up discussed below.)

In [94], the authors introduce the notion of a uniquely localizable problem, i.e., (5.32) is uniquely localizable if it cannot have a non-trivial localization (i.e., a localization different from the one obtained by setting $x_j = (\bar{x}_j, \mathbf{0})$, $j = 1, \ldots, n-m$ where $\bar{x}_j$ is the realization of sensor j in $\mathbb{R}^r$) in some higher dimensional space $\mathbb{R}^h$, with $h > r$. (The anchors are augmented to $\binom{a_k}{\mathbf{0}} \in \mathbb{R}^h$, $k = 1, \ldots, m$.)

If the network is connected, the authors in [94] prove that the solution matrix Z of Problem (5.40) satisfies $Y = XX^T$ if and only if Problem (5.32) is uniquely localizable. Therefore if the original problem (5.32) is uniquely localizable the solution of the **SDP** relaxation (5.40) correctly localizes all the sensors, and it can be computed in polynomial time.

The condition of unique localizability (or realizability) of a graph is then related to rigidity theory in [94]. Let $G' = (V, E)$ be the graph having n nodes corresponding to the sensors and anchors, an edge for each pair $(i, j) \in N_a \cup N_x$, $i, j \in \{1, \ldots, n\}$ and an edge for each pair (k, l), with $k, l \in \{1, \ldots, m\}$, $k \neq l$. In practice, this graph is obtained from the original one by adding the edges connecting the anchors. In [94] the authors prove that, assuming that there are sufficient anchors, problem (5.32) is uniquely localizable if and only if the corresponding graph G' is globally rigid.

The notion of unique realizability, although useful, is not stable under perturbation. For this reason the notion of *strong localizability* is introduced in [94]. Strong localizability requires that the optimal solution of the dual of problem (5.40) has rank $n-m$. This notion can be related to the linear independence of a certain system of linear equations, and it has the desirable property that if a graph contains a strongly localizable subgraph, then the **SDP** solution of (5.40) correctly localizes all the sensors in the subgraph.

5.1.2. Noise in the data. All the results in [94] assume that problem (5.32) is feasible, i.e that all the distances are exact. However, in practice the distances and the lower and upper bounds are noisy, so that (5.32) (or (5.33)) may be infeasible. For this reason, in [13] appropriate nonnegative matrices are used to represent the error due to the noise. To minimize the errors in the feasibility model (5.33) we get:

$$\begin{aligned} &\min e^T[W\circ(C_+ + C_-) + W_{LB}\circ B_- + W_{UB}\circ B_+]e \qquad \text{such that}\\ &\qquad W\circ\big(\mathscr{K}_A(Z) - C_+ + C_-\big) = W\circ\bar{D},\\ &\qquad W_{LB}\circ\big(\mathscr{K}_A(Z) + B_-\big) \geq W_{LB}\circ\underline{D}^l,\\ &\qquad W_{UB}\circ\big(\mathscr{K}_A(Z) - B_+\big) \leq W_{UB}\circ\bar{D}^u, \qquad Z=\begin{bmatrix} I & X^T\\ X & Y\end{bmatrix}\succeq 0,\\ &\qquad B_+, B_-, C_+, C_- \geq 0, \end{aligned} \tag{5.42}$$

where the matrices B_+, B_-, C_+, C_- represent the error.

Suppose that the number of known distances and number of variables are the same, we have accurate distances and linearly independent constraints, the bound constraints are feasible, and the optimal value of (5.42) is zero, then we conclude that (5.42) has a unique solution that is proven to localize the sensors exactly, see [13]. In the general case where the distances are noisy, a probabilistic analysis is carried out in [13], where each x_j is considered as a random variable $\tilde{x}_j$ due to the errors in the distances. Under this interpretation, the solution of problem (5.42) provides the first and second moment information on $\tilde{x}_j$, for all j. In particular, if we are given the solution $\bar{Z}=\left(\begin{smallmatrix} I & \bar{X}^T\\ \bar{X} & \bar{Y}\end{smallmatrix}\right)$ of (5.42), then the quantity $\bar{Y}-\bar{X}\bar{X}^T$ represents the covariance matrix of the random variable $\tilde{x}_j$, $j=1,\ldots,n$; and therefore, the quantity

$$\operatorname{trace}(\bar{Y}-\bar{X}\bar{X}^T)=\sum_{j=1}^{n}(\bar{Y}_{jj}-\|\bar{x}_j\|^2)$$

is a measure of the quality of the approximated distances; while the individual value

$$\bar{Y}_{jj}-\|\bar{x}_j\|^2 \tag{5.43}$$

is helpful in detect distance measure errors of single sensors.

The case of noisy distances is also considered in [10]. The authors introduce upper and lower bounds on the distances that represent confidence intervals for the measurements. Therefore problem (5.32) is formulated as the problem of finding an X such that

$$\begin{aligned} \underline{D}^c_{kj} \le \|a_k - x_j\|^2 \le \bar{D}^c_{kj} \qquad & \text{for all } (k,j) \in N_a, \\ \underline{D}^c_{ij} \le \|x_i - x_j\|^2 \le \bar{D}^c_{ij} \qquad & \text{for all } (i,j) \in N_x, \end{aligned} \tag{5.44}$$

where $[\underline{D}^c_{ij}, \bar{D}^c_{ij}]$ represents the confidence interval for the squared distance D_{ij}. Its **SDP** relaxation is the problem of finding $Z \in \mathcal{S}^{n+r}$ such that

$$W \circ \underline{D}^c \le W \circ \mathcal{K}_A(Z) \le W \circ \bar{D}^c, \qquad Z = \begin{bmatrix} I & X^T \\ X & Y \end{bmatrix} \succeq 0. \tag{5.45}$$

If the distance measurements are exact and the sensor network is uniquely localizable, then the **SDP** relaxations provide the exact localization. In case of noise, the model (5.45) provides a central solution that is the mean of all the **SDP** solutions. However, if the noise level is too high the results obtained by the relaxations can be unsatisfactory.

In [11], two different formulations of the sensor localization problem are considered. The first one corresponds to minimizing the sum of the absolute errors in the localization, namely

$$\min_X \sum_{(i,j)\in N_x} \gamma_{ij} \big| \|x_i - x_j\|^2 - \bar{D}_{ij} \big| + \sum_{(k,j)\in N_a} \gamma_{kj} \big| \|a_k - x_j\|^2 - \bar{D}_{kj} \big|. \tag{5.46}$$

The second one corresponds to the sum of squared errors:

$$\min_X \sum_{(i,j)\in N_x} \gamma_{ij}^2 (\|x_i - x_j\|^2 - \bar{D}_{ij})^2 + \sum_{(k,j)\in N_a} \gamma_{kj}^2 (\|a_k - x_j\|^2 - \bar{D}_{kj})^2. \tag{5.47}$$

In both formulations the weights γ_{ij} and γ_{kj} can be used to exploit the available information, if any, on the reliability of the measures. By relaxing problem (5.46) and setting the matrix $\Gamma = (\gamma_{ij})$, the following **SDP** is obtained:

$$\min e^T \big(\Gamma \circ \big| W \circ \big(\mathcal{K}_A(Z) - \bar{D}\big)\big|\big) e \qquad \text{such that } Z = \begin{bmatrix} I & X^T \\ X & Y \end{bmatrix} \succeq 0. \tag{5.48}$$

The problem (5.47) is relaxed to

$$\min \big\| \Gamma \circ \big(W \circ \big(\mathcal{K}_A(Z) - \bar{D}\big)\big) \big\|_F^2 \qquad \text{such that } Z = \begin{bmatrix} I_r & X^T \\ X & Y \end{bmatrix} \succeq 0, \tag{5.49}$$

i.e., Z, as usual, is as in (5.35). Error bounds depending on the errors in the given distances are derived for both of these formulations in [11]; and, it is empirically shown that these bounds are quite strong. Furthermore, in [11] a different objective function with a regularization term is considered. The aim of this term is to reduce the problem of *crowding*. In fact, when a high rank solution of one of the **SDP** problems is projected on $\mathbb{R}^r$, it often happens that the sensors get crowded together because a large contribution to the distances between two points could come from a projected dimension. The idea is then to penalize crowding from the start by subtracting the term $\lambda\langle I - \mathbf{a}\mathbf{a}^T, Z\rangle$ from the objective function of problem (5.48), where $\mathbf{a} = [e/n \sum_{k=1}^m a_k/\sqrt{n}]$, and $\lambda > 0$ is a regularization parameter. The (heuristic) choice of λ used in [11] is

$$\lambda^* = e^T\big(\Gamma \circ \big|W \circ \big(\mathcal{K}_A(Z^*) - \bar{D}\big)\big|\big)e\langle I - \mathbf{a}\mathbf{a}^T, Z^*\rangle,$$

where Z^* is the optimal solution of problem (5.48) without the regularization term. Furthermore, the solution obtained by solving problem (5.48) with or without the regularization term can be refined by applying a gradient descent method to the (smooth, as $\bar{D}_{ij} \neq 0$) problem

$$\min_{X \in \mathcal{M}^{n\times d}} f(X) = \sum_{(i,j)\in N_x} \gamma_{ij}^2(\|x_i - x_j\| - \bar{D}_{ij})^2 + \sum_{(k,j)\in N_a} \gamma_{kj}^2(\|a_k - x_j\| - \bar{D}_{kj})^2. \quad (5.50)$$

Also, in this case each sensor localization is moved along the negative gradient direction of $f(X)$. If there is a lot of noise, then a combination of the regularization and gradient method generally yields good accuracy in the solution.

5.2. Distributed algorithms. The bottleneck for the **SDP** relaxations is the large dimension of the relaxations and the low accuracy of the resulting solutions. For this reason a distributed **SDP** algorithm was recently developed in [64], [20] and further refined in [14] and [10]. The idea is to partition the anchors into many clusters depending on their physical position. Then each unpositioned sensor is assigned to a cluster whenever it is directly connected to an anchor in the cluster. A sensor can be assigned to more than one cluster; and some sensors can remain unassigned. The **SDP** problem corresponding to each cluster is solved separately; and this can be done in an efficient way, since the size of the clusters is kept below a certain threshold. After solving each of these **SDP** problems, the quality of localization of each unknown sensor is evaluated by considering a suitable error measure. In [14] the trace error measure (5.43) is considered; while in [10] the following error measure is introduced:

$$\mathrm{LDME}_j = \frac{\sum_{i\in N_x^j}(\|x_i - x_j\| - \bar{D}_{ij})^2 + \sum_{k\in N_a^j}(\|a_k - x_j\| - \bar{D}_{kj})^2}{|N_x^j| + |N_a^j|}, \quad (5.51)$$

where $i \in N_x^j$ if $(i, j) \in N_x$ and $k \in N_a^j$ if $(j, k) \in N_a$. If the appropriate error measure is below a certain threshold, the sensor becomes an anchor and the process is repeated.

In [14] the **SDP** model used for the k-th subproblem is based on the nearest matrix problem in (5.42). In solving the **SDP** model for each cluster, many of the lower bound constraints, namely the constraints between two sensor that do not communicate, are often redundant or not active. For this reason a strategy of constraint generation is used. First, only a subset of equality and inequality constraints is added and the problem solved. Then some of the original constraints that are violated, if any, are added to the model. The model is solved again using a *warm start* approach. One advantage of this distributed strategy is that the error does not propagate throughout the whole network, but remains in the cluster.

In [10], after the distributed method has produced a localization, a gradient based method is applied to the whole network in order to improve the solution. This phase is often called *iterative refinement*. In the computational results section of [10], the authors observe that for problems with low noise and small radio range, the **SDP** model (5.56) combined with iterative refinement is better; while for situations where there is larger noise, the **SDP** model (5.45) by itself gives a more accurate solution. In this latter approach, the position of the anchors play an important role. On the one hand, as usual, if the anchors are positioned on the boundary of the feasible set, the quality improves, while if many anchors are positioned in the interior in order to get a good localization, then it is necessary to have high connectivity (i.e a high number of anchors or a large radio range for each sensor). On the other hand, since the clusters are built using the positions of the anchors, the approach proposed in [14] and [10] works better if the anchors are uniformly distributed.

5.2.1. SPASELOC. To overcome the drawback of poorly positioned anchors, a different distributed algorithm is proposed in [20], called SPASELOC. In particular, in [20], the sensors and anchors for each subproblem are chosen dynamically according to specific rules. In this way, the resulting subproblems may have different dimensions, but always below a certain maximum value. The algorithm fixes the maximum number of unlocalized sensors to be included in the considered subproblem. During the algorithm whenever a sensor is localized with a sufficient accuracy, it is labelled as localized. If the accuracy is higher than a certain threshold, then it becomes an *acting anchor*, i.e., it is treated as an anchor for the rest of the iterations. All the acting anchors are assigned a certain level depending on what kind of anchors have been used to localize them. The original anchors are of level 1. In general, the lower the level the higher the reliability of the acting anchor. The choice of which sensors to include is based first on the number of connected anchors they have and then on the level of connected anchors. The sensors connected to at least three anchors are considered first. To localize the ones connected to less than two anchors some geometric heuristics are used. The sensors not connected to anchors are classified as outliers. Not all the candidate anchors are included in the subproblem because adding too many anchors would increase the number of distance constraints, thus increasing the time needed to solve the **SDP** problem and introducing some redundancy. However, in case of large noise, a large number of anchors improves the quality of the solution, so there is a trade off. In choosing the anchors in each subproblem, the original anchors have higher priority. Furthermore, a condition of linear independence between anchors is introduced, and its evaluation requires the computation of a QR factorization of a suitable matrix. The algorithm favors the independent anchors since they minimize the redundant information.

For each subproblem the **SDP** relaxation (5.42) is considered, where the upper bound constraints are removed, namely $W_{UB} = 0$. As for the lower bound constraints, three strategies are implemented:

(i) Problem (5.42) is solved setting $W_{LB} = 0$.

(ii) First, problem (5.42) is solved for $W_{LB} = 0$ and a certain $\hat{Z}$ is found, and then it is solved again including only the inequality constraints that are violated by $\hat{Z}$.

(iii) Problem (5.42) is solved first with $W_{LB} = 0$ and then it is solved again adding each time violated original inequality constraints until all such inequality constraints are satisfied.

The strategy of adding violated inequalities increases the solution time and does not always yield better solutions. On the other hand, using the geometric heuristics to localize sensors that are connected to less than three anchors, greatly improves the quality of the solutions.

It turns out that, in general, algorithm SPASELOC finds a better localization than the full **SDP** approach. This derives from the strategy of building each subproblem using sensors that are connected to at least three anchors. This often yields exact solutions of the **SDP** relaxations of the subproblems. Furthermore, SPASELOC is less sensitive to the number of anchors in the network; and, if the number of anchors is more than 10% of the nodes of the network, then there is no improvement derived from adding more anchors. The SPASELOC algorithm originally dealt with problems with embedding dimension $r = 2$. It has been extended for solving problems in $\mathbb{R}^3$ in [64]. A related distributed algorithm is presented in [91].

5.2.2. Multidimensional scaling. In [29] a different distributed localization algorithm is proposed, that is based on a weighted version of multidimensional scaling. Multidimensional scaling consists in finding a low dimensional representation of a group of objects such that the distances between objects fit as well as possible a given set of measured pairwise dissimilarities. When the measured dissimilarities coincide with the exact distances between sensors, classical multidimensional scaling boils down to computing the singular value decomposition of the centered squared dissimilarity matrix. When the measured dissimilarities contain noise, it consists in iteratively minimizing a loss function between dissimilarities and distances. The idea in [29] is to define a distributed algorithm where some local loss functions are minimized. The local nonlinear least squares problem is then solved by using quadratic majorizing functions. The algorithm produces a sequence of position estimates with corresponding nonincreasing global costs and limited communications between sensors. In the paper [29], the considered global cost function is:

$$\begin{aligned} S = 2 & \sum_{(i,j)\in N_x} \sum_{1\le t\le K} w_{ij}^{(t)} (\|x_i - x_j\| - \delta_{ij}^{(t)})^2 \\ & + \sum_{(k,i)\in N_a} \sum_{1\le t\le K} w_{ik}^{(t)} (\|x_i - a_k\| - \delta_{ki}^{(t)})^2 \\ & + \sum_{1\le i\le n-m} r_i \|x_i - \bar{x}_i\|, \end{aligned} \tag{5.52}$$

where the authors assume that for each distance D_{ij} there are K different measurements $\delta_{ij}^{(t)}$. The weights $w_{ij}^{(t)}, w_{ik}^{(t)} \ge 0$ characterize the accuracy of each measurement. The last term is a penalty term that takes into account prior knowledge about node locations. The cost function (5.52) can be rewritten as

$$S = \sum_{1\le i\le n-m} S_i + c, \tag{5.53}$$

where

$$\begin{aligned} S_i = & \sum_{j\ne i:(i,j)\in \mathcal{N}_x} \bar{w}_{ij} (\|x_i - x_j\| - \bar{\delta}_{ij})^2 \\ & + \sum_{k:(i,k)\in N_a}^{m} 2\bar{w}_{ik} (\|x_i - a_k\| - \bar{\delta}_{ik})^2 + \sum_{i=1}^{n-m} r_i \|x_i - \bar{x}_i\|, \end{aligned} \tag{5.54}$$

with $\bar{w}_{ij} = \sum_{t=1}^{K} w_{ij}^{(t)}$ and $\bar{\delta}_{ij} = \sum_{t=1}^{K} w_{ij}^{(t)} \delta_{ij}^{(t)} / \bar{w}_{ij}$, and c is an appropriate constant. The function S_i is the local function at node i. Therefore each sensor updates its position estimate by minimizing its function S_i on the basis of the position estimates of its neighboring nodes. A crucial issue is how to adaptively choose the neighbors of each node. Given a sensor, its neighborhood contains all the sensors that are within a certain threshold distance. However, distances contain noise, so that due to the noise in the measurements, some sensors might not be assigned to the proper neighbors. In order to take into account noise, the neighbor is built in two steps: in the first step, only the sensors with measurements below a certain threshold are included in the neighborhood, and the algorithm is run with this neighborhood setting, generating certain sensor locations $\hat{x}_1, \dots, \hat{x}_{n-m}$. Then for each sensor the neighborhood is built again on the basis of the distances $\|\hat{x}_i - \hat{x}_j\|$ and the algorithm is rerun. In this way the negative bias effect deriving from the errors in the distances is removed.

A particular choice of function (5.52) (see [10]) is obtained by setting $K = 1$, $w_{ij}^{(t)} = 1/(2\pi^{1/2}\sigma_{ij})$, $w_{ik}^{(t)} = 1/(2\pi^{1/2}\sigma_{kj})$, and $r_i = 0$ for all $i = 1, \dots, n-m$:

$$\begin{aligned} \min v(X) = \sum_{(k,j)\in N_a} \frac{1}{\sigma_{kj}^2}\varepsilon_{kj} + \sum_{(i,j)\in N_x} \frac{1}{\sigma_{ij}^2}\varepsilon_{ij} \quad & \text{such that} \\ (\|a_k - x_j\| - \sqrt{\bar{D}_{kj}})^2 = \varepsilon_{kj} \quad & \text{for all } (k,j) \in N_a, \\ (\|x_i - x_j\| - \sqrt{\bar{D}_{ij}})^2 = \varepsilon_{ij} \quad & \text{for all } (i,j) \in N_x. \end{aligned} \tag{5.55}$$

In [10], problem (5.55) is relaxed to the following **SDP** problem:

$$\begin{aligned} \min \sum_{(k,j)\in N_a} \frac{1}{\sigma_{kj}^2}\varepsilon_{kj} + \sum_{(i,j)\in N_x} \frac{1}{\sigma_{ij}^2}\varepsilon_{ij} \quad & \text{such that} \\ (-\sqrt{\bar{D}_{kj}}\,1)M_{kj}\begin{pmatrix} -\sqrt{\bar{D}_{kj}} \\ 1 \end{pmatrix} = \varepsilon_{kj} \quad & \text{for all } (k,j) \in N_a, \\ (-\sqrt{\bar{D}_{ij}}\,1)M_{ij}\begin{pmatrix} -\sqrt{\bar{D}_{ij}} \\ 1 \end{pmatrix} = \varepsilon_{ij} \quad & \text{for all } (i,j) \in N_x, \\ (a_k^T - e_j^T)Z\begin{pmatrix} a_k \\ -e_j \end{pmatrix} = v_{kj} \quad & \text{for all } (k,j) \in N_a, \\ \big(\mathbf{0}^T (e_i - e_j)^T\big)Z\begin{pmatrix} \mathbf{0} \\ (e_i - e_j) \end{pmatrix} = v_{ij} \quad & \text{for all } (i,j) \in N_x, \\ M_{kj} \succeq 0 \quad & \text{for all } (k,j) \in N_a, \\ M_{ij} \succeq 0 \quad & \text{for all } (i,j) \in N_x, \\ Z \succeq 0, & \end{aligned} \tag{5.56}$$

where Z is defined as in (5.35) and $M_{kj} = \begin{pmatrix} 1 & u_{kj} \\ u_{kj} & v_{kj} \end{pmatrix}$ for all $(k,j) \in N_a$ and $M_{ij} = \begin{pmatrix} 1 & u_{ij} \\ u_{ij} & v_{ij} \end{pmatrix}$ for all $(i,j) \in N_x$. If the noise is multiplicative, i.e., the distances are equal to

$$\sqrt{\bar{D}_{ij}} = \sqrt{D_{ij}}\left(1 + \mathcal{N}(0, \sigma^2)\right),$$

where D_{ij} and $\bar{D}_{ij}$ are the true and measured squared distances respectively, then $\sigma_{ij} = D_{ij}\sigma^2$. Since the true distances are not known, the variances can be approximated by the measured distances, and the objective function of (5.56) is given as

$$\sum_{(k,j) \in N_a} \frac{1}{\bar{D}_{kj}} \varepsilon_{kj} + \sum_{(i,j) \in N_x} \frac{1}{\bar{D}_{ij}} \varepsilon_{ij}.$$

In [10], the authors introduce a gradient local search phase to refine the obtained solution by projecting onto the two dimensional space the solution of the two SDP relaxations (5.56). The idea is to move each sensor location in the opposite direction of that of the gradient of the sum of squared errors. In particular, the maximum likelihood estimation is the solution of the unconstrained optimization problem

$$\begin{aligned} X^* \in \arg\min_X f(X) := & \sum_{(k,j) \in N_a} \frac{1}{\sigma_{kj}^2} \left(\|a_k - x_j\| - \sqrt{\bar{D}_{jk}}\right)^2 \\ & + \sum_{(i,j) \in N_x} \frac{1}{\sigma_{ij}^2} \left(\|x_i - x_j\| - \sqrt{\bar{D}_{ij}}\right)^2. \end{aligned} \tag{5.57}$$

Let the gradient with respect to sensor x_j be denoted $\nabla_{x_j} f$. This gradient can be computed in a distributed way since it depends only on sensors and anchors connected to sensor x_j. The location of sensor x_j is updated in the following way:

$$x_j = x_j - \alpha \nabla_{x_j} f,$$

where α is the step size. This updating rule improves the localization obtained by relaxation (5.56). The effects of this local phase are more pronounced when the anchors are in the interior of the network.

5.2.3. Exact SNL solutions based on facial reductions and geometric build-up. A different distributed **SDP** approach to **SNL** is presented in [73]. This successful technique uses the **EDM** model and solves the **SNL** without using an **SDP** solver. It efficiently finds high accuracy solutions of large problems.

As in [71], [36], [35], using the positions of the anchors is postponed until after the corresponding **EDMC** problem is done. A first *elimination phase* finds faces of the **SDP** cone that correspond to faces of the **EDM** cone that contain cliques in the **SNL** problem. Then, by finding the intersection of appropriate subspaces, the intersection of faces is found. Each intersection of faces corresponds to completing the missing distances in the union of the corresponding cliques. However, finding the distances is postponed. Once the appropriate face of proper dimension is found, then the second *substitution phase* solves for all the missing distances in the **EDM** at once. Then, the third *finalize phase* rotates the anchors to their (approximate) original positions. Extremely large problems can be solved to high accuracy if the distances are exact. Current tests with random data solve problems of order $n = 100{,}000$, $m = 9$ on a laptop in 5 minutes to 16 decimals accuracy. (In the presence of noise, one cannot solve such large problems and the accuracy deteriorates. But some encouraging results have been obtained very recently [72].)

The facial reduction approach is closely (dually) related to the geometric build-up; see the formulas in [9] and the algorithms in [38], [106]. The connection is through the **EDMC** problem and the factorization $PP^T = B = \mathscr{K}^\dagger(D)$.

5.3. Sparse relaxations of SNL. In [68], [69] the **SNL** problem is formulated as a quadratic optimization problem (QOP), and an effective sparse **SDP** relaxation is derived by using tools of polynomial optimization. In the case of exact distances, the constraints of the QOP corresponding to the **SNL** problem is given by the first two equations in (5.33), or equivalently,

$$\begin{aligned} \bar{D}_{kj} &= \|a_k\|^2 + \sum_{p=1}^{r} X_{jp}^2 - 2\sum_{p=1}^{r} X_{jp}A_{kp} \qquad \text{for all } (k,j) \in \mathcal{N}_a, \\ \bar{D}_{ij} &= \sum_{p=1}^{r} X_{ip}^2 + \sum_{p=1}^{r} X_{jp}^2 - 2\sum_{p=1}^{r} X_{ip}X_{jp} \qquad \text{for all } (i,j) \in \mathcal{N}_x, \end{aligned} \tag{5.58}$$

where the columns of $X^T = \{x_1, \ldots, x_{n-m}\} \in \mathbb{R}^{r \times n-m}$.

A sparse relaxation is introduced in [68] that exploits the sparsity of the underlying graph of the **SNL** problem. Let $\mathscr{I}$ be the set of subscripts of the matrix variable X. We can write

$$(X_{pi} : pi \in \mathscr{I})^T (X_{pi} : pi \in \mathscr{I}) = \operatorname{vec} X (\operatorname{vec} X)^T = \sum_{pi \in \mathscr{I}} \sum_{qj \in \mathscr{I}} E_{piqj} X_{ip} X_{jq},$$

where E_{piqj} is the $|\mathscr{I}| \times |\mathscr{I}|$ zero matrix except for the $piqj$ element that is equal to 1. Replacing each term $X_{pi}X_{qj}$ by a single variable U_{piqj}, we obtain a symmetric $|\mathscr{I}| \times |\mathscr{I}|$ matrix variable U and a linearization of the constraints, i.e., $U = xx^T$, where $x = \operatorname{vec} X$. A dense relaxation can be obtained by relaxing this equation to $U \succeq xx^T$. This is equivalent to adding to (5.58) the matrix inequality

$$\begin{pmatrix} 1 & x^T \\ x & U \end{pmatrix} \succeq 0.$$

The resulting relaxation of the constraints in (5.58) is

$$\begin{aligned}
\bar{D}_{ij} &= \sum_{p=1}^{r} U_{pipi} + \sum_{p=1}^{r} U_{pjpj} - 2\sum_{p=1}^{r} U_{pipj} \qquad \text{for all } (i,j) \in \mathcal{N}_x, \\
\bar{D}_{kj} &= \|a_k\|^2 + \sum_{p=1}^{r} U_{pjpj} - 2\sum_{p=1}^{r} X_{jp} A_{kp} \qquad \text{for all } (k,j) \in \mathcal{N}_a, \\
Z &= \begin{bmatrix} 1 & x^T \\ x & U \end{bmatrix} = \in \mathcal{S}_+^{|I|^2+1}.
\end{aligned} \tag{5.59}$$

This is proved to be at least as strong as the **B–Y** relaxation, see [68]; and, it can be equivalently obtained by using the **SDP** programming relaxation of order one in Lasserre [75]. Note that the **B–Y** relaxation has exactly the same structure except that the matrix X is not vectorized, i.e., **B–Y** uses $Z = \begin{pmatrix} I_r & X^T \\ X & Y \end{pmatrix} \succeq 0$. Relaxation (5.59) has a larger dimension than the **B–Y** relaxation. Thus it is expected to be more expensive. In [68], two sparse relaxations are derived starting from (5.59) and from the **B–Y** relaxation. These sparse relaxations can be obtained by either applying the sparse **SDP** relaxation defined in [101] for solving polynomial optimization problems, or equivalently, by using the results on the positive definite completion in [43], [82]. Here, we report only on the sparse version of the **B–Y** relaxation, SFSDP, since it is the most effective one. In order to build it, only the sensors are considered, neglecting the anchors.

Let $\mathcal{C}_1, \ldots, \mathcal{C}_k$ be the maximal cliques of the chordal extension of the graph representing the **SNL** problem where the anchors and the edges in $\mathcal{N}_a$ have been removed. Then the **B–Y** relaxation is relaxed to

$$\begin{aligned}
&\min 0 \text{ such that} \\
&\bar{D}_{ij} = \sum_{p=1}^{r} Y_{ii} + \sum_{p=1}^{r} Y_{jj} - 2\sum_{p=1}^{r} Y_{pi} Y_{pj} \qquad \text{for all } (i,j) \in \mathcal{N}_x, \\
&\bar{D}_{kj} = \|a_k\|^2 + \sum_{p=1}^{r} Y_{jj} - 2\sum_{p=1}^{r} X_{jp} A_{kp} \qquad \text{for all } (k,j) \in \mathcal{N}_a, \\
&\begin{pmatrix} 1 & (\operatorname{vec} X^T[\mathcal{C}_h])^T \\ \operatorname{vec} X^T[\mathcal{C}_h] & \\ & Y(\mathcal{C}_h) \end{pmatrix} \succeq 0 \qquad h = 1, \ldots, k,
\end{aligned} \tag{5.60}$$

where $X^T[\mathscr{C}_h]$ is the $r \times |\mathscr{C}_h|$ matrix containing the positions x_i, $i \in \mathscr{C}_h$. If the sizes of the cliques are relatively small, then this problem becomes less expensive to solve. (Note that extra work is needed to transform this problem to a standard equality form **SDP** problem, see [68].) The size of the system of equations (5.58) can be further reduced by considering only a subset of the distances, i.e., a subgraph of the original graph. This subset is chosen by exploiting the information on the anchor locations and the edges in $\mathscr{N}_a$.

This sparse variant of the **B–Y** relaxation can be generalized by using instead of $\mathscr{C}_1, \dots, \mathscr{C}_k$ a generic family Γ of nonempty subsets of the set $\{1, \dots, n-m\}$ of sensors. This generalization includes as special cases the two relaxations NSDP and ESDP proposed in [102] described in the next subsection. In particular, the node set $\Gamma = \{\{i \in V : (i,j) \in \mathscr{N}_x\} : j \in V\}$ corresponds to the NSDP relaxation, while the edge set $\Gamma = \{(i,j) : (i,j) \in \mathscr{N}_x\}$ corresponds to the ESDP relaxation.

Similar sparse relaxations can be obtained if the distances contain noise, starting from the following QOP:

$$\begin{aligned} &\min \sum_{(i,j)\in\mathscr{N}_x} \xi_{ij}^2 + \sum_{(k,j)\in\mathscr{N}_a} \xi_{kj}^2 \qquad \text{such that} \\ &\qquad \bar{D}_{ij} = \sum_{p=1}^r X_{ip}^2 + \sum_{p=1}^r X_{jp}^2 - 2\sum_{p=1}^r X_{ip}X_{jp} + \xi_{ij} \qquad \text{for all } (i,j) \in \mathscr{N}_x, \\ &\qquad \bar{D}_{kj}^2 = \|a_k\|^2 + \sum_{p=1}^r X_{jp}^2 - 2\sum_{p=1}^r X_{jp}a_{pk} + \xi_{kj} \qquad \text{for all } (k,j) \in \mathscr{N}_a, \end{aligned} \tag{5.61}$$

where ξ_{ij}, ξ_{kj} are error variables.

The sparse relaxation (5.60) of the original noiseless problem outperforms the other full **SDP** relaxations that use an **SDP** solver in terms of cpu time and accuracy [68], [69]. However, it does not do as well as the facial reduction approach which is not based on an **SDP** solver.

5.4. Weaker SNL formulations. Recently many authors focused on weaker relaxations to be solved more efficiently. In [99] a second order cone programming (**SOCP**) relaxation is introduced. It uses a different reformulation of the sensor network localization problem. As above, we let $G = (V, E)$ denote the simple graph on the sensors (and anchors) $1, 2, \dots, n$. Then the SNL problem can be formulated as

$$\min_{x_1,\dots,x_n,y_{ij}} \sum_{(i,j)\in E} |y_{ij} - d_{ij}^2| \qquad \text{such that } y_{ij} = \|x_i - x_j\|^2 \text{ for all } (i,j) \in E,$$

which can be relaxed to

$$\min_{x_1,\dots,x_n,y_{ij}} \sum_{(i,j)\in E} |y_{ij} - d_{ij}^2| \qquad \text{such that } y_{ij} \geq \|x_i - x_j\|^2 \tag{5.62}$$

for all $(i, j) \in E$, which is an **SOCP**. This relaxation is always weaker than the SDP relaxation (5.48) (where all the weights are equal to one). As for the **SDP** relaxation, the solution set is bounded if and only if each connected component of the graph contains an anchor. Indeed, in absence of anchors, the solution set is unbounded and each solution can be rotated and translated to yield another solution. For the **SOCP** (5.62), there exists a unique set $\mathscr{B} \subseteq E$ of constraints that are active in all the solutions, namely:

$$\begin{aligned} \|x_i - x_j\| = y_{ij} \qquad &\text{for all solutions} \\ &x_1, \dots, x_n, (y_{ij})_{(i,j)\in E} \text{ of } (5.62) \iff (i,j) \in \mathscr{B}. \end{aligned} \tag{5.63}$$

Any interior solution satisfies (5.63) and satisfies strictly all the other constraints of (5.62). Let

$$\mathscr{N}_{\mathscr{B}}(i) = \{j \in \{1,\dots,n\} : (i,j) \in \mathscr{B}\}, \qquad \mathscr{M}_{\mathscr{B}} = \{i \in \{1,\dots,n-m\} : \mathscr{N}_{\mathscr{B}}(i) \neq \emptyset\}.$$

In [99] it is proved that all the points with $i \in \mathscr{M}_{\mathscr{B}}$ belong to the convex hull of the points for which the distances are exact, namely satisfy

$$x_i \in \operatorname{conv}\{x_j\}_{j\in\mathscr{N}_{\mathscr{B}}(i)}, \qquad i \in \mathscr{M}_{\mathscr{B}}. \tag{5.64}$$

Furthermore, each connected component of $G_{\mathscr{B}} = (\mathscr{M}_{\mathscr{B}} \cup \{1,\dots,m\}, \mathscr{B})$ contains at least an anchor and for every $i \in \{1,\dots,n\}$, x_i is invariant over all the solutions of (5.62) if and only if $i \in \mathscr{M}_{\mathscr{B}}$. If a particular solution of (5.62) is considered, that is called the analytic center solution, i.e., the interior solution that maximizes

$$\sum_{(i,j)\in E\setminus\mathscr{B}} \log(y_{ij} - \|x_i - x_j\|^2)$$

over all the interior solutions, then

$$x_j \in \operatorname{conv}\{x_j\}_{j\in\mathscr{N}_E(i)}.$$

If the distances contain errors, in [99] it is shown that if the distance error are small, then $(x_i)_{i\in\mathscr{M}_{\mathscr{B}}}$ in a solution of (5.62) has small error that grows proportionally to the square root of the distance error. Relaxation (5.62) can be solved faster than the **B–Y** relaxation, and in [99] a smoothing coordinate gradient descent method is proposed that is effective in solving problem (5.62). The author also suggests the use of problem (5.62) as a problem preprocessor or combined with any method using stronger **SDP** relaxations.

In the same stream of research, a sum-of-squares (SOS) relaxation is proposed in [84], starting from the polynomial formulation (5.50) (with all $\gamma_{ij} = 1$) of the **SNL** Problem. The idea in [84] is to propose a sparse SOS relaxation that exploits the special structure of $f(X)$. In general, the term SOS relaxation describes the process of approximating nonnegative polynomials by polynomials that can be expressed as sum of squares. Checking whether a polynomial is SOS is done by **SDP**. In particular, a polynomial $p(z)$ ($z \in \mathbb{R}^N$) of degree $2l$ is SOS if and only if there exists a symmetric matrix $W \succeq 0$ such that

$$p(z) \equiv m_l(z)^T W m_l(z)$$

where $m(z)$ is the column vector of monomials up to degree l, of dimension up to $\binom{N+l}{l}$. The first SOS relaxation for **SNL** was introduced in [84]:

$$f^*_{\mathrm{sos}} := \max \gamma \qquad \text{such that } f(X) - \gamma = m_2(X)^T W m_2(X), \qquad W \succeq 0. \tag{5.65}$$

If the distances are exact, the SOS relaxation (5.65) is exact, and the solution of its dual can help finding the sensor locations under a technical condition on the solution of the dual (called flat extension condition). A more clever relaxation can be obtained by noting that $f(X)$ can be written in SOS form:

$$f(X) = \sum_{(i,j)\in N_x} \left\{ (\|x_i - x_j\|^2 - \bar{D}_{ij}) + \frac{1}{|S_j|} \sum_{k:(k,j)\in N_a} (\|x_j - a_k\|^2 - \bar{D}_{kj})^2 \right\}, \tag{5.66}$$

where $S_j = \{i : (i,j) \in N_x\}$. Then,

$$f(X) - \gamma = \sum_{(i,j)\in N_x} s_{ij}(x_i, x_j)$$

where $s_{ij}(x_i, x_j)$ are SOS polynomials only in variables x_i, x_j. The corresponding SOS relaxation is

$$\begin{aligned} f^{**}_{\mathrm{sos}} = \max \gamma \qquad &\text{such that } f(X) - \gamma = \sum_{(i,j)\in N_x} m_2(x_i, x_j)^T W_{ij} m_2(x_i, x_j), \\ &W_{ij} \succeq 0, \qquad (i,j) \in N_x. \end{aligned} \tag{5.67}$$

In this formulation, the size of W_{ij} is equal to $(r+1)(2r+1)$ which is independent from n, and the total number of decision variables is $O(r^4|E|)$, with $r=2$ if we are in the plane. Also in this case, if the distances are exact, the SOS relaxation (5.65) is exact, and the solution of its dual can help find the sensor locations under the same technical condition on the solution of the dual. If the distances are perturbed by random noises, it can be shown, under some technical assumptions (including the unique localizability of sensors), that the perturbed solution is accurate within a factor of the perturbation error occurring in the distances.

In [102] the **SDP** approach is further relaxed. The authors propose two new **SDP** relaxations that are obtained by relaxing the single semidefinite matrix cone into a set of small-size semidefinite matrix cones. In particular, the first relaxation is a node-based relaxation:

$$\text{(NSDP)} \qquad \begin{array}{ll} \min\langle \mathbf{0}, Z\rangle & \text{such that} \\ \quad Z_{(1,2),(1,2)} = I_2, & \\ \quad W \circ \mathcal{K}_A(Z) = W \circ D, & \\ \quad Z^i = Z_{(1,2,i,N_i)(1,2,i,N_i)} \succeq 0 & \text{for all } i, \end{array} \qquad (5.68)$$

where $N_i = \{j : (i,j) \in N_x\}$. Here the single $(n+2)$-dimensional cone is replaced by n smaller $3+|N_i|$-dimensional matrix cones, and each of these cones is a principal submatrix of Z. Problem (5.68) can be relaxed for inexact distances to:

$$\text{(NSDPOP)} \qquad \begin{array}{ll} \min e^T\big(W \circ (U+V)\big)e & \text{such that} \\ \quad Z_{(1,2),(1,2)} = I_2, & \\ \quad W \circ \big(\mathcal{K}_A(Z) - U + V\big) = W \circ \bar{D}, & \\ \quad Z^i = Z_{(1,2,i,N_i)(1,2,i,N_i)} \succeq 0 & \text{for all } i, \\ \quad U, V \geq 0. & \end{array} \qquad (5.69)$$

The second relaxation is an edge-based relaxation, as in [84]:

$$\text{(ESDP)} \qquad \begin{array}{ll} \min\langle \mathbf{0}, Z\rangle & \text{such that} \\ \quad Z_{(1,2),(1,2)} = I_2, & \\ \quad W \circ \mathcal{K}_A(Z) = W \circ \bar{D}, & \\ \quad Z_{(1,2,i,j)(1,2,i,j)} \succeq 0 & \text{for all } (i,j) \in N_x. \end{array} \qquad (5.70)$$

Here the single $(n+2)$-dimensional cone is replaced by $|N_x|$ smaller 4-dimensional matrix cones, and also in this case each of these cones is a principal submatrix of Z. In case of inexact distances, it can be relaxed to:

$$
\text{(ESDPOP)}\qquad
\begin{aligned}
&\min e^T\big(W\circ(U+V)\big)e \quad \text{such that}\\
&\quad Z_{(1,2),(1,2)}=I_2,\\
&\quad W\circ\big(\mathscr{K}_A(Z)-U+V\big)=W\circ\bar{D},\\
&\quad Z_{(1,2,i,j)(1,2,i,j)}\succeq 0 \quad \text{for all } (i,j): W_{ij}>0,\\
&\quad U,V\geq 0.
\end{aligned}
\tag{5.71}
$$

Given a problem P, let us denote by F^P the set of solutions of problem P. Then the following relation exists between the three relaxations (5.70), (5.68) and (5.40) (that we denote by **SDP**):

$$F^{\text{SDP}}\subset F^{\text{NSDP}}\subset F^{\text{ESDP}}.$$

However, problem (5.40) has $(n-m+2)^2$ variables and $|N_x|+|N_a|$ constraints, while problem (5.68) has at most $4+2(n-m)+\sum_i|N_i|^2$ variables and $|N_x|+|N_a|$ constraints and problem (5.70) has $4+3(n-m)+|N_x|$ variables and also $|N_x|+|N_a|$ constraints. Therefore, problems (5.68) and (5.70) can be solved much faster, since in general $4+2(n-m)+\sum_i|N_i|^2$ and $4+3(n-m)+|N_x|$ are smaller than $(n-m+2)^2$. Furthermore, the two relaxations (5.68) and (5.70), although weaker than (5.40), preserve some interesting theoretical properties of relaxation (5.40). Indeed, relaxation (5.68) is proved to be equivalent to relaxation (5.40) under the chordal condition, i.e., if every cycle of length greater than three has a chord.

As for relaxation (5.70), in [102] the authors prove that the trace criterion (5.43) to measure the localization accuracy is still valid, but only for its max rank solution (that can be easily identified by using a path-following interior-point method). In fact, if the max rank solution of (5.70) satisfies

$$\overline{Y}_{ii}-\|\bar{x}_i\|^2=0,\qquad i\in\{1,\ldots,n-m\},\tag{5.72}$$

then the i-th column of X is the true location of the i-th sensor, and it is invariant over all solutions Z of (5.70). Furthermore, although weaker than relaxation (5.40), relaxation (5.70) is stronger than the **SOCP** relaxation introduced in [99]. To solve problem (5.70) it turns out that solving the dual is much faster than solving directly problem (5.70), and a primal solution can be easily derived via the complementarity conditions (see [102]). Furthermore, the quality of the solution of problem (5.70) does not depend on the quantity and location of anchors, as for example in SPASELOC.

The two edge-based relaxations (5.70) and (5.71) are again considered in [86], and the authors prove that, given a solution of problem (5.70), the trace criterion

(5.72) is also necessary for the sensor i to be correctly localized by an interior solution. This desirable property does not hold anymore if the distances are inexact, i.e. for relaxation (5.71) (a counterexample is provided). In order to recover this interesting property, the authors define a *robust* version of relaxation (5.71) that maintains this efficient characterization of correctly localized sensors for a certain analytic center solution, provided that the noise in the distances is sufficiently small. In particular, they assume that the squared distances are of the form

$$\bar{D}_{ij} = \|p_i - p_j\|^2 + \delta_{ij}, \qquad \text{for all } (i,j) \in A$$

where $|\delta_{ij}| < \rho_{ij}$, and the threshold values ρ are known. In order to find the analytic center solution the authors define the problem

$$\begin{aligned} \min - \sum_{(i,j)\in N_x} \ln\det(Z_{(1,2,i,j)(1,2,i,j)}) - \sum_{i=1}^{m} \ln\det\begin{pmatrix} I & x_i \\ x_i^T & y_{ii} \end{pmatrix} \\ \text{such that } \left|W \circ \left(\mathscr{K}_A(Z) - \bar{D}\right)\right| \le W \circ \rho, \qquad Z_{(1,2),(1,2)} = I_2, \end{aligned} \tag{5.73}$$

where ρ is the matrix containing the values ρ_{ij}. In this way, the true solution becomes feasible for this relaxation. The authors show that for a certain analytic center solution $(\bar{Y}, \bar{X})$ of this relaxation the trace criterion (5.72) is necessary and sufficient for the correct localization of the sensor i, and that the position error for sensor i is $O(\sqrt{\bar{Y}_{ii} - \|\bar{x}_i\|^2})$. Furthermore, they introduce a coordinate gradient descent method to minimize a log-barrier penalty function in order to find such analytic center solution. This method is much faster than applying an interior point method to problem (5.71), gives a comparable accuracy, and is highly parallelizable, a feature that can be exploited for applications where the localization is required in real time.

6. Summary and outlook

We have shown in this survey that **FPDG** is an elegant problem with many applications and solution techniques. In particular, many instances of **FPDG** such as **GRL**, **GRD**, and **SNL**, are NP-hard problems that can be handled elegantly within the **EDM** framework, and **SDP** can be used to efficiently find solutions for many classes of these problems.

We focused particularly on the **SNL** problem. Many algorithms that are specific for **SNL** use a distributed approach, **SDP** and **SOCP** relaxations, and **SDP** (parallel) solvers. The efficiency for these algorithms has improved from solving instances with about hundreds of nodes in seconds to instances with thousands with an accuracy of several decimals, see the software at the URLs:

http://www.math.nus.edu.sg/~mattohkc/SNLSDP.html
http://www.stanford.edu/~yyye/Col.html
http://www.convexoptimization.com/dattorro/sensor_network_localization.html

Recently, a different approach that does not rely on a **SDP** solver is given in [73], where, in the exact distance case, problems of order $n = 100{,}000$ are solved efficiently to high accuracy.

Motivated by the many important applications, we have seen a rapid development and improvement in both the theory and algorithms for **FPDG** problems. Exploiting sparsity and parallelization has just begun. We can expect new efficient algorithms for larger classes of problems, and that are able to deal with high level of noise.

References

[1] A. Y. Alfakih, On the uniqueness of generic bar-and-joint frameworks. Technical report, University of Windsor, Windsor, Ontario, 2007.

[2] A. Y. Alfakih, A. Khandani, and H. Wolkowicz, Solving Euclidean distance matrix completion problems via semidefinite programming. *Comput. Optim. Appl.* **12** (1999), 13–30. Zbl 1040.90537 MR 1704098

[3] S. Al-Homidan, Hybrid methods for optimization problems with positive semidefinite matrix constraints. PhD thesis, University of Dundee, 1993.

[4] S. Al-Homidan and R. Fletcher, Hybrid methods for finding the nearest Euclidean distance matrix. In *Recent advances in nonsmooth optimization*, World Sci. Publ., Singapore 1995, 1–17. Zbl 0928.65077 MR 1459991

[5] S. Al-Homidan and H. Wolkowicz, Approximate and exact completion problems for Euclidean distance matrices using semidefinite programming. *Linear Algebra Appl.* **406** (2005), 109–141. Zbl 1081.15011 MR 2156431

[6] L. Asimow and B. Roth, The rigidity of graphs. *Trans. Amer. Math. Soc.* **245** (1978), 279–289. Zbl 0392.05026 MR 511410

[7] L. Asimow and B. Roth, The rigidity of graphs. II. *J. Math. Anal. Appl.* **68** (1979), 171–190. Zbl 0441.05046 MR 531431

[8] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. O. Anderson, and P. N. Belhumeur, A theory of network localization. *IEEE Trans. Mobile Comput.* **12** (2006), 1663–1678.

[9] M. Bakonyi and C. R. Johnson, The Euclidean distance matrix completion problem. *SIAM J. Matrix Anal. Appl.* **16** (1995), 646–654. Zbl 0823.15012 MR 1321802

[10] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye. Semidefinite programming based algorithms for sensor network localization. *ACM Trans. Sensor Network* **2** (2006), 188–220.

[11] P. Biswas, T.-C. Liang, K.-C. Toh, Y. Ye, and T.-C. Wang, Semidefinite programming approaches for sensor network localization with noisy distance measurements *IEEE Trans. Autom. Sci. Eng.* **3** (2006), 360–371.

[12] P. Biswas, K.-C. Toh, and Y. Ye, A distributed SDP approach for large-scale noisy anchor-free graph reailzation with applications to molecular conformation. *SIAM J. Sci. Comput.* **30** (2008), 1251–1277. Zbl 1161.49028 MR 2398864

[13] P. Biswas and Y. Ye, Semidefinite programming for ad hoc wireless sensor network localization. In *IPSN '04*, Proc. 3rd internat. symp. information processing in sensor networks, Association for Computing Machinery, New York 2004, 46–54.

[14] P. Biswas and Y. Ye, A distributed method for solving semidefinite programs arising from ad hoc wireless sensor network localization. In *Multiscale optimization methods and applications*, Nonconvex Optim. Appl. 82, Springer, New York 2006, 69–84. Zbl 1100.90029 MR 2191577

[15] Å. Björck, *Numerical methods for least squares problems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996. Zbl 0847.65023 MR 1386889

[16] J. Blitzer, K. Q. Weinberger, L. K. Saul, and F. C. N. Pereira, Hierarchical distributed representations for statistical language modeling. In *Advances in Neural Information Processing Systems 17*, MIT Press, Cambridge, Mass., 2005, 185–192.

[17] L. M. Blumenthal, *Theory and applications of distance geometry*. Second edition, Chelsea Publishing Co., New York 1970. Zbl 0208.24801 MR 0268781

[18] A. M. Burrows, J. T. Richtsmeier, M. P. Mooney, T. D. Smith, H. W. Losken, and M. I. Siegel, Three-dimensional analysis of craniofacial form in a familial rabbit model of nonsyndromic coronal suture synostosis using Euclidean distance matrix analysis. *The Cleft Palate-Craniofacial Journal* **36** (1999), 196–206.

[19] E. Candes and M. Wakin, An introduction to compressive sensing. *IEEE Signal Process. Mag.* 25 (2008), no. 2, 21–30.

[20] M. W. Carter, H. H. Jin, M. A. Saunders, and Y. Ye, SpaseLoc: an adaptive subproblem algorithm for scalable wireless sensor network localization. *SIAM J. Optim.* **17** (2006), 1102–1128. Zbl 1136.90321 MR 2274505

[21] A. Cassioli, Global optimization of highly multimodal problems. PhD thesis, Universita di Firenze, Firenze 2008.

[22] K. Chakrabarty and S. S. Iyengar, *Scalable infrastructure for distributed sensor networks*. Springer, London 2005.

[23] R. Connelly, Rigidity and energy. *Invent. Math.* **66** (1982), 11–33. Zbl 0485.52001 MR 652643

[24] R. Connelly, On generic global rigidity. In *Applied geometry and discrete mathematics*, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 4, Amer. Math. Soc., Providence, RI, 1991, 147–155. Zbl 0736.05069 MR 1116345

[25] R. Connelly, Rigidity. In *Handbook of convex geometry*. Vol. A, North-Holland, Amsterdam 1993, 223–271. Zbl 0788.52001 MR 1242981

[26] R. Connelly, Tensegrity frameworks: Why are they stable? In *Rigidity theory and applications*, Kluwer Academic/Plenum Publishers, New York, 1999, 47–54.

[27] R. Connelly, Generic global rigidity. *Discrete Comput. Geom.* **33** (2005), 549–563. Zbl 1072.52016 MR 2132290

[28] R. Connelly and M. Sloughter, Realizability of graphs. Technical report, Dept. of Math., Cornell University, Ithaca, NY, 2004.

[29] J. A. Costa, N. Patwari, and A. O. Hero, III, Distributed weighted-multidimensional scaling for node localization in sensor networks. *ACM Trans. Sensor Network* **2** (2006), 39–64.

[30] G. M. Crippen, Chemical distance geometry: Current realization and future projection. *J. Math. Chem.* **6** (1991), 307–324.

[31] G. M. Crippen and T. F. Havel, *Distance geometry and molecular conformation*. Chemometrics Series 15, Research Studies Press Ltd., Chichester 1988. Zbl 1066.51500 MR 0975025

[32] B. C. Csáji, J. Küng, J. Palkoska and R. Wagner. On the automation of similarity information maintenance in flexible query answering systems. In *Database and expert systems applications* (DEXA 2004), Lect. Notes Comput. Sci. 3180, Springer-Verlag, Berlin 2004, 130–140.

[33] J. Dattorro, *Convex optimization & Euclidean distance geometry*. Meboo Publishing USA, Palo Alto, CA, 2005.

[34] J. Dattorro, Equality relating Euclidean distance cone to positive semidefinite cone. *Linear Algebra Appl.* **428** (2008), 2597–2600. Zbl 1142.15014 MR 2416574

[35] Y. Ding, N. Krislock, J. Qian, and H. Wolkowicz, Sensor network localization, Euclidean distance matrix completions, and graph realization. In *MELT '08: Proceedings of the first ACM international workshop on mobile entity localization and tracking in GPS-less environments*, Association for Computing Machinery, New York 2008, 129–134.

[36] Y. Ding, N. Krislock, J. Qian, and H. Wolkowicz, Sensor network localization, Euclidean distance matrix completions, and graph realization. *Optim. Eng.* **11** (2010), 45–66. MR 2601732

[37] L. Doherty, K. S. J. Pister, and L. El Ghaoui, Convex position estimation in wireless sensor networks. In *Proceedings IEEE INFOCOM 2001*, Vol. 3, IEEE, Piscataway, NJ, 2001, 1655–1663.

[38] Q. Dong and Z. Wu, A geometric build-up algorithm for solving the molecular distance geometry problem with sparse distance data. *J. Global Optim.* **26** (2003), 321–333. Zbl 1032.92036 MR 1977953

[39] A. W. M. Dress and T. F. Havel, The fundamental theory of distance geometry. In *Raisonnement géométrique assisté par ordinateur* (Sophia-Antipolis, 1987), Vol. I, INRIA Rocquencourt, Le Chesnay 1987, 127–169. MR 0921058

[40] T. Eren, D. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. O. Anderson, and P. N. Belhumeur, Rigidity, computation, and randomization of network localization. *Proceedings IEEE INFOCOM 2004*, Vol. 4, IEEE, Piscataway, NJ, 2004, 2673–2684.

[41] R. W. Farebrother, Three theorems with applications to Euclidean distance matrices. *Linear Algebra Appl.* **95** (1987), 11–16. Zbl 0627.15008 MR 907390

[42] V. F. Ferrario, C. Sforza, A. Miani, Jr, and G. Serrao, A three-dimensional evaluation of human facial asymmetry. *J. Anat.* **186** (1995), 103–110.

[43] M. Fukuda, M. Kojima, K. Murota, and K. Nakata, Exploiting sparsity in semidefinite programming via matrix completion. I: General framework. *SIAM J. Optim.* **11** (2000/01), 647–674 (electronic). Zbl 1010.90053 MR 1814036

[44] H. N. Gabow and H. H. Westermann, Forests, frames, and games: algorithms for matroid sums and applications. *Algorithmica* **7** (1992), 465–497. Zbl 0771.05026 MR 1154585

[45] D. Gale, Neighboring vertices on a convex polyhedron. In *Linear inequalities and related systems*, Ann. of Math. Stud. 38, Princeton University Press, Princeton, N.J., 1956, 255–263. Zbl 0072.37805 MR 0085552

[46] M. R. Garey and D. S. Johnson, *Computers and intractability*. W. H. Freeman and Co., San Francisco 1979. Zbl 0411.68039 MR 0519066

[47] H. Gluck, Almost all simply connected closed surfaces are rigid. In *Geometric topology* (Proc. Conf., Park City, Utah, 1974), Lecture Notes in Math. 438, Springer, Berlin 1975, 225–239. Zbl 0315.50002 MR 0400239

[48] W. Glunt, T. L. Hayden, S. Hong, and J. Wells, An alternating projection algorithm for computing the nearest Euclidean distance matrix. *SIAM J. Matrix Anal. Appl.* **11** (1990), 589–600. Zbl 0728.65034 MR 1066161

[49] S. J. Gortler, A. D. Healy, and D. P. Thurston, Characterizing generic global rigidity. *Amer. J. Math.* **132** (2010), 897–939. Zbl 1202.52020 MR 2663644

[50] S. J. Gortler and D. P. Thurston, Characterizing the universal rigidity of generic frameworks. Preprint 2009. arXiv:1001.0172

[51] J. C. Gower, Euclidean distance geometry. *Math. Sci.* **7** (1982), 1–14. Zbl 0492.51017 MR 642165

[52] J. C. Gower, Properties of Euclidean and non-Euclidean distance matrices. *Linear Algebra Appl.* **67** (1985), 81–97. Zbl 0569.15016 MR 787367

[53] H. Grassmann, *Extension theory*. History of Mathematics 19, American Mathematical Society, Providence, RI, 2000. Zbl 0953.01025 MR 1747519

[54] J. Graver, B. Servatius, and H. Servatius, *Combinatorial rigidity*. Grad. Stud. Math. 2, American Mathematical Society, Providence, RI, 1993. Zbl 0788.05001 MR 1251062

[55] M. Grötschel, L. Lovász, and A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica* **1** (1981), 169–197. Zbl 0492.90056 MR 625550

[56] T. F. Havel, Metric matrix embedding in protein structure calculations, NMR spectra analysis, and relaxation theory. *Magn. Reson. Chem.* **41** (2003), S37–S50.

[57] T. L. Hayden, J. Wells, W. M. Liu, and P. Tarazaga, The cone of distance matrices. *Linear Algebra Appl.* **144** (1991), 153–169. Zbl 0718.15011 MR 1081882

[58] B. Hendrickson, The molecule problem: determining conformation from pairwise distances. PhD thesis, Cornell University, Ithaca, NY, 1990.

[59] B. Hendrickson, Conditions for unique graph realizations. *SIAM J. Comput.* **21** (1992), 65–84. Zbl 0756.05047 MR 1148818

[60] B. Hendrickson, The molecule problem: exploiting structure in global optimization. *SIAM J. Optim.* **5** (1995), 835–857. Zbl 0844.05093 MR 1358807

[61] H.-X. Huang, Z.-A. Liang, and P. M. Pardalos, Some properties for the Euclidean distance matrix and positive semidefinite matrix completion problems. *J. Global Optim.* **25** (2003), 3–21. Zbl 1037.15015 MR 1969424

[62] B. Jackson and T. Jordán, Connected rigidity matroids and unique realizations of graphs. *J. Combin. Theory Ser. B* **94** (2005), 1–29. Zbl 1076.05021 MR 2130278

[63] B. Jackson, T. Jordán, and Z. Szabadka, Globally linked pairs of vertices in equivalent realizations of graphs. *Discrete Comput. Geom.* **35** (2006), 493–512. Zbl 1088.05056 MR 2202116

[64] H. Jin, Scalable sensor localization algorithms for wireless sensor networks. PhD thesis, University of Toronto, Toronto, Ontario, 2005.

[65] C. R. Johnson, B. Kroschel, and H. Wolkowicz, An interior-point method for approximate positive semidefinite completions. *Comput. Optim. Appl.* **9** (1998), 175–190. Zbl 0907.90207 MR 1601892

[66] C. R. Johnson and P. Tarazaga, Connections between the real positive semidefinite and distance matrix completion problems. *Linear Algebra Appl.* **223/224** (1995), 375–391. Zbl 0827.15032 MR 1340702

[67] D. S. Kim, Sensor network localization based on natural phenomena. PhD thesis, MIT, Cambridge 2006.

[68] S. Kim, M. Kojima, and H. Waki, Exploiting sparsity in SDP relaxation for sensor network localization. *SIAM J. Optim.* **20** (2009), 192–215. Zbl 1190.65096 MR 2496899

[69] S. Kim, M. Kojima, H. Waki, and M. Yamashita, SFSDP: a Sparse version of full semidefinite programming relaxation for sensor network localization problems. Technical Report B-457, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo 2009.

[70] N. Krislock, Efficient algorithms for large-scale Euclidean distance matrix problems with applications to wireless sensor network localization and molecular conformation. PhD thesis, University of Waterloo, Waterloo, Ontario, 2010.

[71] N. Krislock, V. Piccialli, and H. Wolkowicz, Robust semidefinite programming approaches for sensor network localization with anchors. Research Report CORR 12, Dept. Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, 2006. http://orion.uwaterloo.ca/~hwolkowi/henry/reports/ABSTRACTS.html

[72] N. Krislock, F. Rendl, and H. Wolkowicz, Noisy sensor network localization using semidefinite representations and facial reduction. Research Report CORR 2010-01, University of Waterloo, Waterloo, Ontario, 2010. http://orion.uwaterloo.ca/~hwolkowi/henry/reports/ABSTRACTS.html

[73] N. Krislock and H. Wolkowicz, Explicit sensor network localization using semidefinite representations and facial reductions. *SIAM J. Optim.* **20** (2010), 2679–2708. MR 2678410

[74] G. Laman, On graphs and rigidity of plane skeletal structures. *J. Engrg. Math.* **4** (1970), 331–340. Zbl 0213.51903 MR 0269535

[75] J. B. Lasserre, Global optimization with polynomials and the problem of moments. *SIAM J. Optim.* **11** (2001), 796–817. Zbl 1010.90061 MR 1814045

[76] M. Laurent, Cuts, matrix completions and graph rigidity. *Math. Programming* **79** (1997), 255–283. Zbl 0887.90174 MR 1464770

[77] M. Laurent, A tour d'horizon on positive semidefinite and Euclidean distance matrix completion problems. In *Topics in semidefinite and interior-point methods*, Fields Inst. Commun. 18, Amer. Math. Soc., Providence, RI, 1998, 51–76. Zbl 0906.05074 MR 1607310

[78] L. Lovász and Y. Yemini, On generic rigidity in the plane. *SIAM J. Algebraic Discrete Methods* **3** (1982), 91–98. MR 644960

[79] J. J. Moré and Z. Wu, ε-optimal solutions to distance geometry problems via global continuation. In *Global minimization of nonconvex energy functions: molecular conformation and protein folding* (New Brunswick, NJ, 1995), DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 23, Amer. Math. Soc., Providence, RI, 1996, 151–168. Zbl 0857.49023 MR 1369128

[80] J. J. Moré and Z. Wu, Global continuation for distance geometry problems. *SIAM J. Optim.* **7** (1997), 814–836. Zbl 0891.90168 MR 1462067

[81] J. J. Moré and Z. Wu, Distance geometry optimization for protein structures. *J. Global Optim.* **15** (1999), 219–234. Zbl 0944.92012 MR 1737437

[82] K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima, and K. Murota, Exploiting sparsity in semidefinite programming via matrix completion. II. Implementation and numerical results. *Math. Programming* **95** (2003), 303–327. Zbl 1030.90081 MR 1976483

[83] S. Nawaz, Anchor free localization for ad-hoc wireless sensor networks. PhD thesis, University of New South Wales, Sydney 2008.

[84] J. Nie, Sum of squares method for sensor network localization. *Comput. Optim. Appl.* **43** (2009), 151–179. Zbl 1170.90510 MR 2506248

[85] P. M. Pardalos, D. Shalloway, and G. Xue (eds.), *Global minimization of nonconvex energy functions: molecular conformation and protein folding*. DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 23, Amer. Math. Soc., Providence, RI, 1996. Zbl 0831.00024 MR 1369122

[86] T. K. Pong and P. Tseng, (Robust) edge-based semidefinite programming relaxation of sensor network localization. Technical Report Jan-09, University of Washington, Seattle, WA, 2009; *Math. Programming*, DOI 10.1007/s10107-009-0338-x

[87] M. V. Ramana, L. Tunçel, and H. Wolkowicz, Strong duality for semidefinite programming. *SIAM J. Optim.* **7** (1997), 641–662. Zbl 0891.90129 MR 1462059

[88] B. Recht, M. Fazel, and P. A. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.* **52** (2010), 471–501. Zbl 1198.90321 MR 2680543

[89] K. Römer, Time synchronization and localization in sensor networks. PhD thesis, No. 16106, ETH Zurich, Zurich 2005.

[90] C. Savarese, J. Rabaey, and J. Beutel, Locationing in distributed ad-hoc wireless sensor networks. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Proceedings, IEEE, Piscataway. NJ, 2001, 2037–2040.

[91] A. Savvides, C.-C. Han, and M. B. Strivastava, Dynamic fine-grained localization in ad-hoc networks of sensors. In *MobiCom 2001: Proceedings of the 7th annual international conference on mobile computing and networking*, Association for Computing Machinery, New York 2001, 166–179.

[92] J. B. Saxe, Embeddability of weighted graphs in k-space is strongly NP-hard. In *Proceedings Seventeenth Annual Allerton Conference on Communication, Control, and Computing*, University of Illinois, Department of Electrical Engineering, Urbana-Champaign, Illinois, 1979.

[93] 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. of Math.* (2) **36** (1935), 724–732. JFM 61.0435.04 Zbl 0012.30703 MR 1503248

[94] A. M.-C. So and Y. Ye, Theory of semidefinite programming for sensor network localization. *Math. Programming* **109** (2007), 367–384. Zbl 05131081 MR 2295148

[95] W. S. Torgerson, Multidimensional scaling. I. Theory and method. *Psychometrika* **17** (1952), 401–419. Zbl 0049.37603 MR 0054219

[96] M. W. Trosset, Computing distances between convex sets and subsets of the positive semidefinite matrices. Technical report, Rice University, Houston, Texas, 1997.

[97] M. W. Trosset, Applications of multidimensional scaling to molecular conformation. Computing Science and Statistics 29 (1998), 148–152.

[98] M. W. Trosset, Distance matrix completion by numerical optimization. *Comput. Optim. Appl.* **17** (2000), 11–22. Zbl 0964.65047 MR 1791595

[99] P. Tseng, Second-order cone programming relaxation of sensor network localization. *SIAM J. Optim.* **18** (2007), 156–185. Zbl 1176.90454 MR 2299679

[100] S. Urabl, Cooperative localization in wireless sensor networks. Master's thesis, University of Klagenfurt, Klagenfurt, Austria, 2009.

[101] H. Waki, S. Kim, M. Kojima, and M. Muramatsu, Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity. *SIAM J. Optim.* **17** (2006), 218–242. Zbl 1109.65058 MR 2219151

[102] Z. Wang, S. Zheng, Y. Ye, and S. Boyd, Further relaxations of the semidefinite programming approach to sensor network localization. *SIAM J. Optim.* **19** (2008), 655–673. Zbl 1173.90498 MR 2425034

[103] K. Q. Weinberger, F. Sha, and L. K. Saul, Learning a kernel matrix for nonlinear dimensionality reduction. In *ICML '04 Proceedings of the twenty-first international conference on Machine learning*, Association for Computing Machinery, New York 2004, 106.

[104] W. Whiteley, Matroids and rigid structures. In *Matroid applications*, Encyclopedia Math. Appl. 40, Cambridge Univ. Press, Cambridge 1992, 1–53. Zbl 0768.05025 MR 1165538

[105] H. Wolkowicz, R. Saigal, and L. Vandenberghe (eds.), *Handbook of semidefinite programming*. International Series in Operations Research & Management Science 27. Kluwer Academic Publishers, Norwell, MA, 2000.

[106] D. Wu and Z. Wu, An updated geometric build-up algorithm for solving the molecular distance geometry problems with sparse distance data. *J. Global Optim.* **37** (2007), 661–673. Zbl 1119.92032 MR 2302995

[107] J.-M. Yoon, Y. Gad, and Z. Wu, Mathematical modeling of protein structure using distance geometry. Technical report, University of Houston, Houston 2000.

[108] F. Z. Zhang, On the best Euclidean fit to a distance matrix. *J. Beijing Normal Univ.* (Nat. Sci.) **1987** (1987), No. 4, 21–24. Zbl 0655.15023 MR 939410

[109] Z. Zou, R. H. Byrd, and R. B. Schnabel, A stochastic/perturbation global optimization algorithm for distance geometry problems. Technical report, Dept. of Computer Science, University of Colorado, Boulder, Co., 1996.

Received November 2, 2009; revised June 17, 2010

A. Y. Alfakih, Department of Mathematics & Statistics, University of Windsor, 401 Sunset Avenue, Windsor ON, N9B 3P4, Canada
E-mail: alfakih@uwindsor.ca

M. F. Anjos, Department of Mathematical and Industrial Engineering, Ecole Polytechnique de Montreal, C.P. 6079, succ. Centre-ville, Montréal, Quebec, Canada H3C 3A7
E-mail: miguel-f.anjos@polymtl.ca

V. Piccialli, Dipartimento di Ingegneria dell'Impresa, Università degli Studi di Roma ‘‘Tor Vergata’’, Via del Politecnico 1, 00133 Roma, Italy
E-mail: piccialli@disp.uniroma2.it

H. Wolkowicz, Department of Combinatorics & Optimization, University of Waterloo, 200 University Avenue West, Waterloo ON, N2L 3G1, Canada
E-mail: hwolkowicz@uwaterloo.ca