Euclidean Distance Matrices, Semidefinite Programming, and Sensor Network Localization

Abdo Y. Alfakih [*] Miguel F. Anjos [†] Veronica Piccialli [‡]
Henry Wolkowicz [§]

November 2, 2009

University of Waterloo
Department of Combinatorics and Optimization
Waterloo, Ontario N2L 3G1, Canada
Research Report CORR 2009-05

Key Words: Euclidean distance matrix completions, sensor network localization, fundamental problem of distance geometry, semidefinite programming.
AMS Subject Classification:

Abstract

The fundamental problem of distance geometry, ***FPDG***, involves the characterization and study of sets of points based only on given values of (some 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, ***EDM***, play an important role in ***FPDG***. They use the squared distances and provide elegant and powerful convex relaxations for ***FPDG***. These ***EDM*** problems are closely related to graph realization, ***GRL***; and graph rigidity, ***GRD***, plays an important role. Moreover, by relaxing the embedding dimension restriction, ***EDM*** problems

[*] Department of Mathematics & Statistics, University of Windsor. Research supported by Natural Sciences Engineering Research Council of Canada.

[†] Department of Management Sciences, University of Waterloo, and Universität zu Köln, Institut für Informatik, Pohligstrasse 1, 50969 Köln, Germany. Research supported by Natural Sciences Engineering Research Council of Canada, MITACS, and the Humboldt Foundation.

[‡] Dipartimento di Ingegneria dell'Impresa Università degli Studi di Roma "Tor Vergata" Via del Politecnico, 1 00133 Rome, Italy.

[§] Research supported by Natural Sciences Engineering Research Council Canada, MITACS, and AFOSR.

can be approximated efficiently using semidefinite programming, ***SDP*** . Throughout this survey we emphasize the interplay between: ***FPDG*** , ***EDM*** , ***GRL*** , ***GRD*** , and ***SDP*** . In addition, we illustrate our concepts on one instance of ***FPDG*** , the Sensor Network Localization Problem, ***SNL*** .

Contents

1 Introduction

The fundamental problem of distance geometry (***FPDG***) involves the characterization and study of sets of points, $p_1, \ldots, p_n \in \mathbb{R}^r$ based only on given values for (some of) the distances between pairs of points. More precisely, given only (partial, approximate) distance information $\bar{d}_{ij} \approx \|p_i - p_j\|, ij \in E$, [1] between pairs of points, we need to determine whether we can realize 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, 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 problem since they provide an elegant and strong relaxation for ***FPDG*** . The ***EDM*** consists of the *squared* Euclidean distances between points, $D_{ij} = \|p_i - p_j\|^2, i, j = 1, \ldots, n$. Using the squared rather than ordinary distances, and further relaxing the embedding dimension r, means that completing a partial ***EDM*** is a convex problem. Moreover, a global solution 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., [84, 21] for details on compressed sensing.)

A special instance of ***FPDG*** is the Sensor Network Localization problem (***SNL***). For ***SNL*** , the n points $p_i, i = 1, \ldots, n$, 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 sensors are (approximately) known if and only if the sensors are within a given radio range, R. The ***SNL*** has recently emerged as an important research topic. In this survey we concentrate on the ***SNL*** problem and its connections with ***EDM*** , graph realization (***GRL***), graph rigidity (***GRD***), and ***SDP*** .

Our goal in this survey 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

We work with points (real vectors) $p_1, \ldots, p_n \in \mathbb{R}^r$, where r is the *embedding dimension* of the problem. We let $P^T = \begin{bmatrix} p_1, \ldots, p_n \end{bmatrix} \in \mathcal{M}^{r \times n}$ denote the matrix with columns formed from the set of points. For ***SNL*** , $P = \begin{bmatrix} A \\ X \end{bmatrix}$, where the rows $p_i^T = a_i^T, i = 1, \ldots m$, of $A \in \mathcal{M}^{mr}$ are the positions of the m anchor nodes, and the rows $x_i^T = p_{m+i}^T, i = 1, \ldots, n-m$, 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 $1, 2, \ldots, n$ with edge set E. Typically, for ***FPDG*** the distances $\|x_i - x_j\|, i, j \in E$, 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

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

$\|A\|_F$. More generally, $\langle A, B\rangle = \operatorname{trace} A^TB$ denotes the inner product of two compatible, general, real matrices A, B, and $\|A\|_F = \sqrt{\operatorname{trace} A^TA}$ 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 $p_1, \ldots, p_n$. We let e_i denote the i-th unit vector, e denote the vector of ones, both of appropriate dimension, and $E = ee^T$; $\mathcal{R}(\mathcal{L}), \mathcal{N}(\mathcal{L})$ denotes the range space and nullspace of the linear transformation $\mathcal{L}$, respectively; $\mathcal{L}^*$ denotes the adjoint of $\mathcal{L}$, i.e., $\langle \mathcal{L}(x), y\rangle = \langle x, \mathcal{L}^*(y)\rangle, \forall x, y$; $\mathcal{L}^\dagger$, 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. 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 follow the notation in e.g., [70]: 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

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

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

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

The set $K \subset \mathbb{R}^n$ is a *convex cone* if $\mathbb{R}^n_+(K) \subseteq K, K + K \subseteq K$. $\operatorname{cone}(S)$ 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

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

We write $F \lhd K$ to denote $F \trianglelefteq K, F \neq K$. If $\{0\} \neq F \lhd K$, then F is a *proper face* of K. For $S \subseteq K$, we let $\operatorname{face}(S)$ denote the smallest face of K that contains S.

For a set $S \subset \mathbb{R}^n$, let $S^* := \{\phi \in \mathbb{R}^n : \langle \phi, S\rangle \subseteq \mathbb{R}_+\}$ denote the *polar cone of* S. That $\mathcal{S}^n_+ = {\mathcal{S}^n_+}^*$ is well known, i.e., the ***SDP*** cone is self-polar. 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 results in problems in duality. Here $F^c = \mathcal{S}^n_+ \cap F^\perp$ denotes the *conjugate face* of F.

Lemma 2.1 ([101],[83]) *Suppose that F is a proper face of $\mathcal{S}^n_+$, i.e., $\{0\} \neq F \lhd \mathcal{S}^n_+$. Then:*

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

$$\mathcal{S}^n_+ + \operatorname{span} F^c \text{ is not closed.} \tag{2.3}$$

■

Further notation is introduced as needed, and summarized in the index beginning at page 47.

3 FPDG and EDM

Distance geometry involves the characterization and study of sets of points based only on given values of (some of) the distances between pairs of the points. The origins of the algebra for distance geometry can be traced back to 1896 and the work of Grassmann [52] and continued in the modern era in e.g. [51, 32, 39]. One of the methods used to study ***FPDG*** is to view the problem using the squared distances, i.e., using a Euclidean Distance Matrix, ***EDM***. This allows the application of powerful tools from convex analysis and linear algebra and, more specifically, from Semidefinite Programming, ***SDP***. (This is the approach we emphasize in this survey.)

Theoretical properties of EDMs can be found in e.g., [10, 42, 19, 50, 56, 65, 73, 90]. 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 Laurent [73], and more recently in the book [33]. A discussion on the difficulty in finding efficient algorithms for ***EDMC*** appears in [95]. There are many algorithms that find approximate completions; e.g., [95, 94, 93] presents 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 e.g., [48, 104, 2]. (The harder global model without *squared* distances but with intervals for the distances, is used in e.g., [76, 77, 105].)

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 *predistance matrix* or a *dissimilarity matrix*. In addition, if there exist points $p_1, p_2, \ldots, p_n \in \mathbb{R}^r$ such that

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

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* (***EDMC***) 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, see [64, 4],

$$\begin{array}{cc} \min & \|W \circ (\bar{D} - D)\| \\ \text{subject to} & D \in \mathcal{E}. \end{array} \tag{3.5}$$

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, e.g. [17]

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

3.1 Distance Geometry, EDM , and SDP

Let $P^T = \begin{bmatrix} p_1 & p_2 & \dots & p_n \end{bmatrix} \in \mathcal{M}^{rn}$ 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

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

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

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

where offDiag (D) replaces the diagonal of D with zeros; see (2.1). The linear operators $\mathcal{K}, \mathcal{T}$ are one-one and onto between the *centered* and *hollow* subspaces of $\mathcal{S}^n$. In the classical literature, the linear operator $\mathcal{T}$ is only defined on the subspace $\mathcal{S}_H$. We extend it to all of $\mathcal{S}^n$ with the addition of the operator (projection) offDiag . This means that we now have a simple explicit expression for the Moore-Penrose generalized inverse $\mathcal{K}^\dagger = \mathcal{T}$. See (2.1) and Prop. 3.1 below.

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

Lemma 3.1 ([3]) *Let $X \in \mathcal{S}^{n-1}$ and partition*

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

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

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

■

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

Proposition 3.1 *([3]) The operators $\mathcal{K}, \mathcal{T}$ satisfy*

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

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

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

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, $0 \leq D \in \mathcal{S}_H$, is a ***EDM*** if and only if D is negative semidefinite on $\{e\}^\perp$, the orthogonal complement of e; see e.g. [90, 50, 56, 92]. We now collect this with other characterizations; see e.g. [4, 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

$$\mathcal{K}_V(X) \quad := \quad \mathcal{K}(VXV^T). \tag{3.15}$$

The adjoint of K_V is

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

Let

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

Lemma 3.2 ([4])

$$\mathcal{K}_V(\mathcal{S}^{n-1}) = \mathcal{S}_H,$$
$$\mathcal{T}_V(\mathcal{S}_H) = \mathcal{S}^{n-1},$$

and $\mathcal{K}_V$ and $\mathcal{T}_V$ are inverses of each other on these two spaces. ■

Remark 3.1 *To obtain a one-one mapping between $D \in \mathcal{E}^n$ and $X \in \mathcal{S}^n_+$, one usually adds the centering constraint $Xe = 0$. However, this means that X is restricted to a face of $\mathcal{S}^n_+$ and is singular; and therefore, the Slater constraint qualification (strict feasibility) fails for a **SDP** formulation that uses $\mathcal{K}$. Lemma 3.2 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-one mapping that also has strictly feasible points, i.e., there exists $\hat{X} \in \mathcal{S}^{n-1}_{++}$ such that $\mathcal{K}_V(\hat{X}) = \hat{D} \in \mathcal{E}^n$ and $\mathcal{T}_V(\hat{D}) = \hat{X}$. This is essential for interior-point methods and for 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 3.1 *The following characterizations of $D \in \mathcal{E}^n$ hold.*

1. $D \in \mathcal{S}_H \cap \mathcal{M}^n_+ \cap \{D \in \mathcal{S}^n : v^T e = 0 \implies v^T D v \leq 0\}$
2. $D = \mathcal{K}(B)$, *for some* $B \succeq 0$, *with* $Be = 0$, $B \in \mathcal{S}^n$
3. $D = \mathcal{K}_V(X)$, *for some* $X \in \mathcal{S}^{n-1}_+$
4. $D = \mathcal{L}(X) := \begin{bmatrix} 0 & \operatorname{diag}(X)^T \\ \operatorname{diag}(X) & \mathcal{D}_e(X) - 2X \end{bmatrix}$, *for some* $X \in \mathcal{S}^{n-1}_+$
5. $D = \begin{bmatrix} 0 & \operatorname{diag}(X)^T + (s_X e^T - 2x_r^T) \\ \operatorname{diag}(X) + (s_X e - 2x_r) & \mathcal{D}_e(X) - 2X \end{bmatrix}$, *for some* $X \in \mathcal{S}^{n-1}_+$, *where* $s_X := e^T X e,\ x_r := Xe$
6. $\mathcal{E}^n = \mathcal{K}\left(\mathcal{S}^n_+\right) = \mathcal{K}_V\left(\mathcal{S}^{n-1}_+\right), \quad \mathcal{T}_V\left(\mathcal{E}^n\right) = \mathcal{S}^{n-1}_+$
7. $\mathcal{E}^n = \mathcal{S}_H \cap \left(\mathcal{S}^\perp_C - \mathcal{S}^n_+\right) = \mathcal{S}_H \cap \left(\mathcal{R}(\mathcal{D}_e) - \mathcal{S}^n_+\right)$

■

Proof.

1. Item 1 is the classical characterization of $\mathcal{E}^n$. Proofs are given in e.g. [90, 50, 56, 92]. 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 [4]. Also, it follows from the definition of V and Item 2.

4. Item 4 is given in [1, 3].

5. Item 5 is proved in [3]. It also follows from Item 4 since

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

6. Item 6 is proved in [4] 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 cone $(E) \lhd \mathcal{S}^n_+$ and $\{E\}^\perp = \mathcal{S}_C$. From Lemma 2.1 and Proposition 3.1, 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

$$\begin{array}{rcll} \mathcal{E}^n & = & -(\mathcal{S}_C \cap \mathcal{S}^n_+)^* \cap \mathcal{S}_H, & \text{from Item 1} \\ & = & (\mathcal{S}_C^\perp - \mathcal{S}^n_+) \cap \mathcal{S}_H & \\ & = & (\mathcal{R}(\mathcal{D}_e) - \mathcal{S}^n_+) \cap \mathcal{S}_H, & \text{from Proposition 3.1.} \end{array}$$

■

We have emphasized several times that we are using *squared* distances. The advantages are that we get a convex relaxation if we use $\boldsymbol{EDM}$ and relax the rank. 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 $p_1, \ldots, p_n \in \mathbb{R}^r$ that satisfy

$$\|p_i - p_j\|^2 = D_{ij}, \forall ij \in E.$$

In practice, the distances are only known approximately, e.g. upper and lower bounds are given

$$l_{ij} \leq \|p_i - p_j\| \leq u_{ij}, \forall ij \in E.$$

See e.g. [75, 78], where the distances (not squared) are used. If the rank constraint is not relaxed, then it is well known that the $\boldsymbol{FPDG}$ 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 dimensions is fixed, then this is an NP-hard problem in general, see e.g., [60] for complexity issues related to ***EDMC*** . This formulation can be relaxed using the characterizations in Theorem 3.1 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,

$$\begin{array}{cc} \min & \|W \circ (\bar{D} - \mathcal{K}(Y))\|_F^2 \\ \text{subject to} & Y \in \mathcal{S}_+^n \,, \end{array} \tag{3.18}$$

where we have chosen the Frobenius norm in the objective function. Since $\operatorname{int} \mathcal{E}^k = \emptyset$ and $\mathcal{K}$ maps one-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 [4]. Additional equality or upper and lower bound constraints (in $\bar{D}_{UB}$, and $\bar{D}_{LB}$, respectively) can be added using additional weight matrices W_E, W_{UB} and W_{LB}, respectively.

$$\begin{array}{lrcl} \min & & \|W \circ (\bar{D} - \mathcal{K}_V(Y))\|_F^2 & \\ \text{subject to} & & & \\ & & W_E \circ \mathcal{K}_V(Y) & = \; W_E \circ \bar{D} \\ & W_{LB} \circ \bar{D}_{LB} & \leq \; W_{LB} \circ \mathcal{K}_V(Y) & \\ & & W_{UB} \circ \mathcal{K}_V(Y) & \leq \; W_{UB} \circ \bar{D}_{UB} \\ & & Y \in \mathcal{S}_+^{k-1} \,. & \end{array} \tag{3.19}$$

Here $\mathcal{K}_V$ is defined in (3.15), and $B \leq C$ denotes $C - B \geq 0$, elementwise. Though we have a convex relaxation of ***EDMC*** , the approximation is generally poor if the optimal solution has a large rank, e.g., see the estimates in [4, Lemma 2]. Reducing the rank is an NP-hard problem and related to compressed sensing, e.g., [84, 21].

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 distance geometry problems and, in particular, ***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 the function of the protein, 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 3.1. 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 [32]. There has since been huge progress in this area, see e.g. [81, 31, 59, 4, 103, 55] 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 [43] 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 [20], or the normal levels of facial asymmetry in humans [43].

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 $\boldsymbol{EDM}$ -based approach to similarity search was proposed recently in [11]. 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 $\boldsymbol{EDM}$ 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 $\boldsymbol{EDM}$. 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 $\boldsymbol{EDMs}$, it is actually possible to do much better. Blitzer et al. [18] 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;

2. vectors representing semantically dissimilar words should be well separated.

The idea in [18] is to pursue both of these objectives via the following $\boldsymbol{SDP}$:

$$\begin{array}{rl} \max & \sum_{ij} D_{ij} \\ \text{subject to} & \mathcal{T}_V(D) \succeq 0 \\ & D_{ij} = \|p_i - p_j\|^2, \text{ for all similar vector pairs } p_i, p_j, \end{array} \tag{3.20}$$

where $\mathcal{T}_V$ 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 [99].

We briefly mention the application of $\boldsymbol{EDM}$ to graph realization, $\boldsymbol{GRL}$. Given a simple graph G with vertices $1, 2, \ldots, n$ and non-negative edge weights $\{D_{ij} : ij \in E\}$, we call a

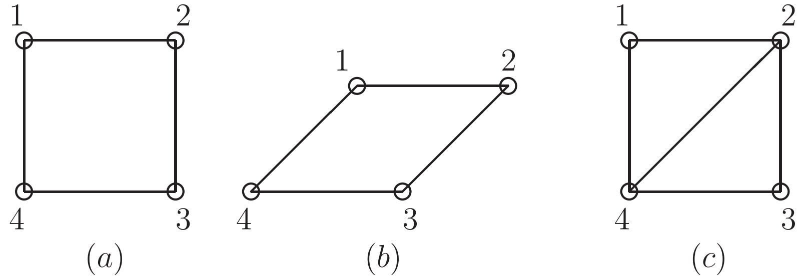

Figure 4.1: An example of three bar frameworks in $\mathbb{R}^2$. Frameworks (a) and (b) are equivalent and flexible; while framework (c) is rigid.

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 [89] and Aspnes et al. [9]. However, some graph families admit polynomial-time algorithms [12, 13, 14, 22, 22]. Also, Connelly and Sloughter [29] 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 the citations in 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* [41] 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. 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 4.1 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 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 $p_1, \ldots, p_n$ are algebraically independent over the integers. That is, $G(p)$ is generic if there does not exist a polynomial f of the components of the p_i with integer coefficients such that

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

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 $\epsilon > 0$, there does not exist any framework $G(q)$ in $\mathbb{R}^r$, which is equivalent to $G(p)$, such that $\|q_i - p_i\| \leq \epsilon$ for all $i = 1, \ldots, n$. 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:

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

Any $\bar{p} = (\bar{p}_1, \ldots, \bar{p}_n)$ 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* [27, 25, 32, 53, 100].

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

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

The converse of the previous Theorem 4.1 is false. However, Asimow and Roth [7] showed that the notions of rigidity and infinitesimal rigidity coincide for generic bar frameworks.

It is well known [47, 8] 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 $\mathbb{R}^r$ 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 the (i,j)th row of R is given by

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

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 (Asimow and Roth [7]) *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. Next we consider the problem of combinatorial characterization of generic bar frameworks.

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 (Laman [71] Lovász and Yemini [74]) *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, 54, 72]. 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 4.1 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 [28, 41, 61, 62]. Hendrickson [58, 59] 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 [26] to be false for $r \geq 3$.

Jackson and Jordán [61] proved that Hendrickson's conjecture is true for $r = 2$.

Theorem 4.4 (Jackson and Jordán [61], Hendrickson [58]) *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

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

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

$$s_{ij} = \begin{cases} -\omega_{ij} & \text{if } (i,j) \in E \\ 0 & \text{if } (i,j) \not\in E \\ \sum_{k:(i,k)\in E} \omega_{ik} & \text{if } i = j. \end{cases}$$

S is called the *stress matrix* associated with ω. Connelly [28] 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 (Connelly [28]) *Let $G(p)$ be a given generic bar framework $G(p)$ 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.*

Theorem 4.6 (Connelly [28], Gortler *et al* [49]) *Let $G(p)$ be a given generic framework $G(p)$ 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 $S = n-1-r$.* ■

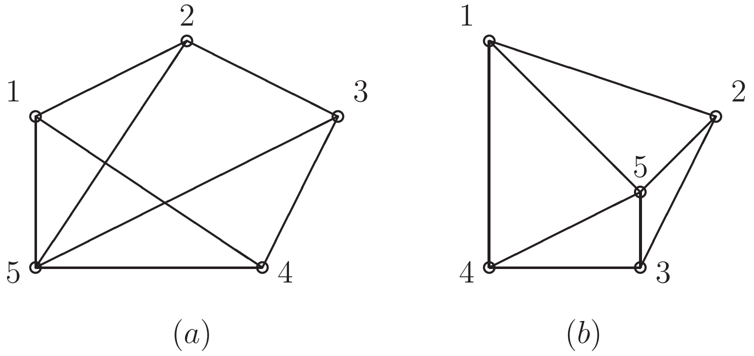

Figure 4.2: An example of two frameworks in $\mathbb{R}^2$. The framework in (a) is universally rigid while the framework in (b) is globally rigid but not universally rigid.

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 4.2 is globally rigid in $\mathbb{R}^2$ but it is not universally rigid.

Alfakih [6] 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 page 17, below.) 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

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

Recall that $p_1, \dots, p_n$ are not contained in a proper hyperplane in $\mathbb{R}^r$, i.e., the affine space spanned by $p_1, \dots, p_n$ 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 *[Alfakih [6]]* [2] *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 the following is a sufficient condition for $G(p)$ to be universally rigid:*

$$\exists\ \bar{r} \times \bar{r} \text{ symmetric positive definite matrix } \Psi : {z^i}^T \Psi z^j = 0, \forall (i,j) \notin E, \tag{4.26}$$

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

Conjecture 4.1 (Alfakih [6]) *Let $G(p)$ be a given generic bar framework in $\mathbb{R}^r$ with n vertices for some $r \leq n-2$, and let Z be the Gale matrix for $G(p)$. If $G(p)$ is universally rigid then Condition (4.26) holds.* ■

In [5, Example 3.1] it is shown that this conjecture is false if the framework $G(p)$ is not generic.

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 4.1 (Alfakih [6]) *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*

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

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 4.1 (Connelly [25]) *Let S be the stress matrix associated with an equilibrium stress ω for framework $G(p)$ with n vertices in $\mathbb{R}^r$, then*

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

[2] Theorem 4.7 was also obtained by Connelly in an unpublished manuscript.

■

In light of Lemma 4.1, 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 and Conjecture 4.1 can be stated equivalently as follows:

Theorem 4.8 *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*

$$\exists\ \bar{r} \times \bar{r} \text{ symmetric non-singular matrix } \Psi : {z^i}^T \Psi z^j = 0, \forall (i,j) \notin E, \tag{4.29}$$

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

Theorem 4.9 *Let $G(p)$ be a generic framework with n vertices in $\mathbb{R}^r$. Then $G(p)$ is universally rigid if there exists a positive semi-definite stress matrix S associated with an equilibrium stress ω for $G(p)$ such that* $\operatorname{rank} S = \bar{r} = n-1-r$. ■

Conjecture 4.2 *Let $G(p)$ be a given generic framework in $\mathbb{R}^r$ with n vertices. If $G(p)$ is universally rigid then there exists a positive semi-definite stress matrix S associated with an equilibrium stress ω for $G(p)$ such that* $\operatorname{rank} S = \bar{r} = n-1-r$. ■

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 $\mathcal{K}, \mathcal{T}$, e.g., Proposition 3.1. 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, e.g. in 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 dust e.g. farmland or chemical plant explosions, etc.

> A quote: "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 many publications, e.g., International Journal of Sensor Networks; recent related theses and books include: [57, 85, 33, 24, 63, 66, 23, 79, 97]. Research groups include CENS at UCLA and Berkeley WEBS. The algorithmic side has advanced quickly. From solving problems with $n = 50$ sensors with low accuracy, current codes can quickly solve problems with $100,000$s of sensors to high accuracy:

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

The algorithms for ***SNL*** often use minor modifications that identify anchors with sensors. In fact, see [69, 35, 36, 70], a set of anchors simply corresponds to a given fixed clique for the graph of the ***EDM*** problem. 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 [70] 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 e.g. linear, semidefinite, conic, or polynomial; and, on the other hand they define *distributed*, rather than centralized, approaches to handle the large sizes 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

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

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, $\boldsymbol{LP}$, constraints: two to bound the angle; and another one to bound the distance. Any combination of the $\boldsymbol{SDP}$ and $\boldsymbol{LP}$ 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 $\boldsymbol{LP}$ constraints. The first set of constraints (5.30) can be solved using a second order cone programming solver, the other set uses an $\boldsymbol{LP}$ 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 $\boldsymbol{SNL}$ include [63, 22, 86, 88]. This idea has been exploited and further developed by Ye and his coauthors in [16, 12, 13, 14, 91, 98]. Their approach is termed the Biswas-Ye ($\boldsymbol{B-Y}$) SDP relaxation and is used as well in e.g., [82, 67, 68]. 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 [70], see Section 5.2.3.

5.1 Biswas-Ye SDP Relaxation, EDMC, and Facial Reduction

The $\boldsymbol{B-Y}$ $\boldsymbol{SDP}$ relaxation of $\boldsymbol{SNL}$ (see the discussion in Section 5 above and (5.35) below) is used in many algorithms for solving $\boldsymbol{SNL}$ problems. Therefore, it is of interest to understand its relationship with the classical relaxations based on $\boldsymbol{EDMC}$. The $\boldsymbol{B-Y}$ relaxation can be derived directly from the definitions, e.g., [15]. Alternatively, we can use the approach in [69, 35, 70] and derive this relaxation from the $\boldsymbol{EDM}$ framework. In fact, we now show that the $\boldsymbol{B-Y}$ relaxation can also be obtained as a *restricted* second step in facial reduction for the $\boldsymbol{EDM}$ relaxation, following on the one for centering in Remark 3.1. This second step is based on the fact that the anchors form a clique in the graph of the $\boldsymbol{SNL}$ (corresponding to a principal submatrix in the $\boldsymbol{EDM}$ 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 5.1 and Remark 5.1, below, provide the details as well as a comparison between the $\boldsymbol{B-Y}$ relaxation and $\boldsymbol{EDMC}$.

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) $\boldsymbol{EDM}$ $\bar{D}$ 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 $\boldsymbol{SNL}$. We first give the basic result for facial reduction for $\boldsymbol{EDMC}$.

Theorem 5.1 ([35, 70]) *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*

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

■

Theorem 5.1 shows that if we know the distances for a clique of cardinalty k with embedding dimension t, then we can reduce the size of the matrix variable in the $\boldsymbol{SDP}$ representation of the $\boldsymbol{EDM}$ from n to $n-k+t$. Now suppose that we are given an $\boldsymbol{SNL}$ problem, i.e. we are given the position of the anchors $a_j, j = 1,\dots,m$, and a partial $\boldsymbol{EDM}$ $\bar{D}$, 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 $x_1,\dots,x_{n-m} \in \mathbb{R}^r$ such that

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

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

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

Recall the description of the $\boldsymbol{SNL}$ problem in Section 2. The matrix P of nodes is partitioned as $P = \begin{bmatrix} A \\ X \end{bmatrix}$, where the position of the anchors $p_i = a_i, i = 1,\dots m$, are the columns of $A^T \in \mathcal{M}^{rm}$; and the unknown positions of the sensors $p_{m+i} = x_i, i = 1,\dots,m-n$, 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 [15], problem (5.32) is modelled using the equivalent (5.34) and is formulated as the following ***SDP*** feasibility problem: find a symmetric matrix $Z \in \mathcal{S}^{n-m+r}$ such that

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

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

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

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 $\bar{P} = \begin{bmatrix} v_1 & \dots & v_r \end{bmatrix} \operatorname{Diag}(\lambda)$. 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

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

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

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

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 5.1 *Define the nonnegative weight matrix* $0 \leq W_E \in \mathcal{M}^n$ *by*

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

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*

$$\mathcal{F}^{EDM}_{U_A} := \left\{ Z : \begin{array}{rcl} & W_E \circ \mathcal{K}_{U_A}(Z) & = & W_E \circ D_E \\ W_{LB} \circ D_{LB} & \leq & W_{LB} \circ \mathcal{K}_{U_A}(Z) & \\ & W_{UB} \circ \mathcal{K}_{U_A}(Z) & \leq & W_{UB} \circ D_{UB} \\ & Z \in \mathcal{S}^{n-m+r}_+ & & \end{array} \right\} \tag{5.39}$$

and

$$\mathcal{F}^{BY}_{A} := \left\{ Z : \begin{array}{rcl} & W_E \circ \mathcal{K}_{A}(Z) & = & W_E \circ D_E \\ W_{LB} \circ D_{LB} & \leq & W_{LB} \circ \mathcal{K}_{A}(Z) & \\ & W_{UB} \circ \mathcal{K}_{A}(Z) & \leq & W_{UB} \circ D_{UB} \\ & Z = \begin{bmatrix} I & X^T \\ X & Y \end{bmatrix} \in \mathcal{S}^{n-m+r}_+ & & \end{array} \right\} \tag{5.40}$$

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

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

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

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

Remark 5.1 *Lemma* (5.1) *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 e.g., the discussion in Section 5.1.2. Therefore, if we replace the objective functions in Lemma 5.1 with the convex function* $\operatorname{trace}(Z_Y - Z_X Z_X^T)$, *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 [35] 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 e.g., [40, 79]. In contrast to using the ***EDMC*** approach outlined in Section 5.1 and Lemma 5.1, localizability is based on finding the location of a sensor using neighbouring 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 [91], 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 $\begin{pmatrix} a_k \\ \mathbf{0} \end{pmatrix} \in \mathbb{R}^h$, $k = 1, \ldots, m$.)

If the network is connected, the authors in [91] 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 [91]. 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 [91] 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 very useful, is not stable under perturbation. For this reason the notion of *strong localizability* is introduced in [91]. 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 [91] assume that problem (5.32) is feasible, i.e that all the distances are exact. However in practice both distances and lower and upper bounds are noisy, so that (5.32) (or (5.33)) may be infeasible. For this reason, in [15] an appropriate objective function is used to modify the relaxation (5.33):

$$\begin{array}{rl} \min & \operatorname{trace}\left(J(W \circ (C^+ + C^-))\right) + \operatorname{trace}\left(J(W_{LB} \circ B^-)\right) + \operatorname{trace}\left(J(W_{UB} \circ B^+)\right) \\ \text{s.t.} & W \circ (\mathcal{K}_A(Z) - C^+ + C^-) = W \circ \bar{D} \\ & W_{LB} \circ (\mathcal{K}_A(Z) + B^-) \geq W_{LB} \circ D_{LB} \\ & W_{UB} \circ (\mathcal{K}_A(Z) - B^+) \leq W_{UB} \circ D_{UB} \\ & Z = \begin{bmatrix} I & X^T \\ X & Y \end{bmatrix} \succeq 0 \\ & B^+, B^-, C^+, C^- \geq 0 \end{array} \qquad (5.42)$$

If 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 (5.42) has a unique solution that is proven to localize the sensors exactly, see [15]. In the general case where the distances are noisy, a probabilistic analysis is carried out in [15], 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, given the solution

$$\bar{Z} = \begin{pmatrix} I & \bar{X}^T \\ \bar{X} & \bar{Y} \end{pmatrix}$$

of (5.42), the quantity

$$\bar{Y} - \bar{X}\bar{X}^T$$

represents the covariance matrix of the random variable $\tilde{x}_j$, $j = 1, \dots, n$, and therefore the quantity

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

is a measure of the quality of the distances, while the individual trace

$$\bar{Y}_{jj} - \|\bar{x}_j\|^2 \qquad (5.43)$$

can be helpful to detect distance measure errors of single sensors.

The case of noisy distances is again considered in [12]. The authors introduce upper and lower bounds on the distances that represent confidence intervals of the measurements. Therefore problem (5.32) is formulated as the problem of finding an X such that:

$$\begin{array}{ll} D^l_{kj} \leq \|a_k - x_j\|^2 \leq D^u_{kj} & \forall (k,j) \in N_a \\ D^l_{ij} \leq \|x_i - x_j\|^2 \leq D^u_{ij} & \forall (i,j) \in N_x \end{array} \qquad (5.44)$$

where $[\underline{\mathrm{D}}_{ij}, \bar{D}_{ij}]$ represents the confidence interval for the squared distance D_{ij}. Its ***SDP*** relaxation is the problem of finding $Z \in \mathcal{S}^{(n+2)}$ such that

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

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 [13], 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

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

The second one corresponds to the sum of squared errors:

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

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), the following ***SDP*** is obtained

$$\begin{array}{ll} \min & \operatorname{trace}\left(J(\Gamma \circ \left|W \circ (\mathcal{K}_A(Z) - \bar{D})\right|)\right) \\ \text{s.t.} & Z = \begin{bmatrix} I & X^T \\ X & Y \end{bmatrix} \succeq 0. \end{array} \tag{5.48}$$

While problem (5.46) is relaxed to

$$\begin{array}{ll} \min & \left\{\operatorname{trace}\left(J(\Gamma \circ \left(W \circ (\mathcal{K}_A(Z) - \bar{D})\right)^2)\right)\right\}^{1/2} \\ \text{s.t.} & Z = \begin{bmatrix} I & X^T \\ X & Y \end{bmatrix} \succeq 0 \end{array} \tag{5.49}$$

where Z as usual is given in (5.35). Error bounds depending on the error in the distances are derived in [13] for both of these formulations, and it is empirically shown that these bounds are quite tight. Furthermore, in [13] a different objective function is considered, where a regularization term is introduced. The effect of this term should be to reduce the problem of crowding. In fact, when the higher 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 an ignored dimension. The idea is then to penalize crowding from the start, by subtracting from the objective function of problem (5.48) the term

$$\lambda\langle I - \mathbf{a}\mathbf{a}^T, Z\rangle, \tag{5.50}$$

where $\mathbf{a} = \left[e/(n+m)\sum_{k=1}^{m} a_k/\sqrt{n+m}\right]$, and $\lambda > 0$ is a regularization parameter. The heuristic choice of λ used in [13] is

$$\lambda^* \;=\; \operatorname{trace}\left(J(\Gamma \circ \left|W \circ (\mathcal{K}_A(Z^*) - \bar{D})\right|)\right)/\left\langle I - \mathbf{a}\mathbf{a}^T, Z^*\right\rangle,$$

where Z^* is the optimal solution of problem (5.48) without the regularization term. Again, 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 problem

$$\min_{X \in \mathcal{M}^{n\times d}} f(X) = \sum_{(i,j)\in N_x} \gamma_{ij}^2 \left(\|x_i - x_j\| - \bar{D}_{ij}\right)^2 + \sum_{(k,j)\in N_a} \gamma_{kj}^2 \left(\|a_k - x_j\| - \bar{D}_{kj}\right)^2. \qquad (5.51)$$

Also in this case each sensor localization is moved along the negative gradient direction of $f(X)$. In presence of high noise a combination of the regularization and gradient method yields a good accuracy in the solution.

5.2 Distributed Algorithms

The bottleneck for the ***SDP*** relaxations has been the large dimension and low accuracy of the problems that can be solved. For this reason a distributed ***SDP*** algorithm was recently developed in [63, 22] and further refined in [16] and [12]. The idea is to partition the anchors in many clusters depending on their physical position, and then each unpositioned sensor is assigned to a cluster whenever it is directly connected to an anchor in the cluster. In this way, a sensor can be assigned to more than one cluster, and some sensors can be 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 cluster is kept below a certain threshold. After solving each cluster the quality of localization of each unknown sensor is evaluated by considering a suitable error measure: in [16] the trace error measure (5.43) is considered, while in [12] a different error measure is introduced:

$$\mathrm{LDME}_j = \frac{\displaystyle\sum_{i\in N_x^j} (\|x_i - x_j\| - \bar{D}_{ij})^2 + \sum_{k\in N_a} (\|a_k - x_j\| - \bar{D}_{kj})^2}{|N_x^j| + |N_a|^j}, \qquad (5.52)$$

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 considered error measure is below a certain threshold, the sensor becomes an anchor and the process is reiterated.

In [16] the ***SDP*** model used for the k-th subproblem is:

$$\begin{array}{ll} \min & \operatorname{trace}\left(J(W_k \circ B)\right) \\ \text{s.t.} & W_k \circ \mathcal{K}_A(Z) = W_k \circ D \\ & \bar{W}_k \circ \mathcal{K}_A(Z) \geq R\bar{W}_k \circ J \\ & Z = \begin{bmatrix} I & X^T \\ X & Y \end{bmatrix} \succeq 0, B \geq 0 \end{array} \qquad (5.53)$$

where W_k has positive weights corresponding to the subset of known distances included in the subproblem, and $\bar{W}_k$ has positive weights corresponding to the subset of unknown distances between sensors and anchors considered in the subproblem.

In solving the $\boldsymbol{SDP}$ model for each cluster, many of the "bounding away" 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 then the violated ones, if any, are added to the model and it is solved again with a "warm start" solution. This strategy considerably speeds up the solution of problem (5.53) since in general the number of iterations needed to get a feasible optimal solution is small. One advantage of this distributed strategy is that the error does not propagate throughout the whole network, but remains in the cluster.

In [12] after the distributed method has produced a localization, the gradient based method is applied to the whole network in order to improve the solution. Different algorithms are implemented, depending on which $\boldsymbol{SDP}$ model is used and whether or not there exists a local gradient based phase. For problems with low noise and low radio ranges, the $\boldsymbol{SDP}$ model (5.58) combined with a gradient based method is better, while for situations where there is more noise the $\boldsymbol{SDP}$ model by itself (5.45) gives a better accuracy. In this approach, the position of the anchors plays 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 the anchors are in the interior to get a good localization it is necessary to have a high connectivity (i.e a high number of anchors or a large radio range for each sensor). On the other hand, since each cluster is built on the basis of the physical positions of the anchors, the approach proposed in [16] and [12] works well only if the anchors are uniformly distributed in the search space.

5.2.1 SPASELOC

To overcome the drawback of poorly positioned anchors, a different distributed algorithm is proposed in [22], called SPASELOC. In particular, in [22], the subsensors and subanchors for each subproblem are chosen dynamically according to some 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 have 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 included subsensors is based first on the number of connected anchors they have and then on the level of connected anchors. The subsensors 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 subsensors 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, increasing the time needed to solve the $\boldsymbol{SDP}$ problem and introducing some redundancy. However, in conditions of high 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 $\boldsymbol{SDP}$ 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 $\widehat{Z}$ is found, and then it is solved again including only the inequality constraints that are violated by $\widehat{Z}$.

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

The strategy of adding violated inequalities increases the solution time and not always gives 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 solution.

It turns out that in general algorithm SPASELOC finds a better localization than the full $\boldsymbol{SDP}$ approach, and this derives from the strategy of building each subproblem considering subsensors that are connected to at least three anchors. This makes often exact the solution of the $\boldsymbol{SDP}$ relaxation of the subproblem. 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, there is no improvement derived from adding more anchors. The SPASELOC algorithm has been extended for solving problems in $\mathbb{R}^3$ in [63]. A related distributed algorithm is presented in [87]. More recently, a distributed approach that exploits the sparsity in the $\boldsymbol{SDP}$ relaxations is given in [67, 68].

5.2.2 Multidimensional Scaling

In [30] a different distributed localization algorithm is proposed, that is based on a weighted version of multidimensional scaling. The multidimensional scaling consists in finding a low dimension representation of a group of objects such that the distances between objects fit as well as possible a set of measured pairwise dissimilarities. When the measured dissimilarities coincide with the exact distances between sensors, classical multidimensional scaling consists in a 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 [30] 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 non increasing global cost and limited communications between sensors. In the paper [30] the considered global function is:

$$\begin{array}{rcl} S & = & 2\sum\limits_{(i,j)\in N_x}\sum\limits_{1\leq t\leq K} w_{ij}^{(t)}(\|x_i - x_j\| - \delta_{ij}^{(t)})^2 + \sum\limits_{(k,i)\in N_a}\sum\limits_{1\leq t\leq K} w_{ik}^{(t)}(\|x_i - a_k\| - \delta_{ki}^{(t)})^2 \\ & & \quad + \sum_{1\leq i\leq n} r_i\|x_i - \bar{x}_i\|, \end{array} \tag{5.54}$$

where the authors assume that for each distance D_{ij} there are K different measurements $\delta_{ij}^{(t)}$. The arbitrary weights $w_{ij}^{(t)}, w_{ik}^{(t)} \geq 0$ classify the accuracy of each measurement. The last term is a penalty term that takes into account prior knowledge about node locations. The stress (5.54) function can be rewritten as

$$S = \sum_{1\leq i\leq n} S_i + c \tag{5.55}$$

where

$$S_i = \sum_{j=1,j\neq i}^{n} \bar{w}_{ij}(\|x_i - x_j\| - \bar{\delta}_{ij})^2 + \sum_{k:(i,k)\in N_a}^{m} 2\bar{w}_{ik}(\|x_i - a_k\| - \bar{\delta}_{ik})^2 + \sum_{i=1}^{n} r_i\|x_i - \bar{x}_i\| \tag{5.56}$$

with $\bar{w}_{ij} = \sum_{t=1}^{K} w_{ij}^{(t)}$ and $\bar{\delta}_{ij} = \sum_{t=1}^{K} w_{ij}^{(t)}\delta_{ij}^{(t)}/\bar{w}_{ij}$. 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 distant from it less than a certain threshold. However, distances contain noise, so that due to the noise in the measurements, some sensors could be not assigned to the 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 $\hat{x}_1, \ldots, \hat{x}_n$. 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 case of function (5.54) is considered in [12], that is the one obtained by setting $K = 1$, $w_{ij}^{(t)} = 1/(2\pi^{\frac{1}{2}}\sigma_{ij})$ and $w_{ik}^{(t)} = 1/(2\pi^{\frac{1}{2}}\sigma_{kj})$, and $r_i = 0$, for all $i = 1, \ldots, n$. The obtained function is

$$\begin{array}{lll} \min & v(X) = \sum\limits_{(k,j)\in N_a} \dfrac{1}{\sigma_{kj}^2}\epsilon_{kj} + \sum\limits_{(i,j)\in N_x} \dfrac{1}{\sigma_{ij}^2}\epsilon_{ij} & \\ \text{s.t.} & (\|a_k - x_j\| - \sqrt{D_{kj}})^2 = \epsilon_{kj} & \forall(k,j)\in N_a \\ & (\|x_i - x_j\| - \sqrt{D_{ij}})^2 = \epsilon_{ij} & \forall(i,j)\in N_x \end{array} \tag{5.57}$$

In [12] problem (5.57) is relaxed into the following ***SDP*** problem:

$$\begin{array}{lrll}
\min & \displaystyle\sum_{(k,j)\in N_a} \frac{1}{\sigma_{kj}^2}\epsilon_{kj} + \sum_{(i,j)\in N_x} \frac{1}{\sigma_{ij}^2}\epsilon_{ij} & & \\
\text{s.t.} & (-\sqrt{\bar{D}_{kj}}\ 1)M_{kj}\begin{pmatrix}-\sqrt{\bar{D}_{kj}}\\ 1\end{pmatrix} = \epsilon_{kj} & \forall(k,j)\in N_a & \\
& (-\sqrt{\bar{D}_{ij}}\ 1)M_{ij}\begin{pmatrix}-\sqrt{\bar{D}_{ij}}\\ 1\end{pmatrix} = \epsilon_{ij} & \forall(i,j)\in N_x & \\
& (a_k^T\ \ -e_j^T)Z\begin{pmatrix}a_k\\ -e_j\end{pmatrix} = v_{kj} & \forall(k,j)\in N_a & (5.58)\\
& (\mathbf{0}^T\ (e_i-e_j)^T)Z\begin{pmatrix}\mathbf{0}\\ (e_i-e_j)\end{pmatrix} = v_{ij}, & \forall(i,j)\in N_x & \\
& M_{kj}\succeq 0 & \forall(k,j)\in N_a & \\
& M_{ij}\succeq 0 & \forall(i,j)\in N_x & \\
& Z\succeq 0, & &
\end{array}$$

where Z is defined as in (5.35) and

$$M_{kj} = \begin{pmatrix}1 & u_{kj}\\ u_{kj} & v_{kj}\end{pmatrix}\ \forall(k,j)\in N_a \text{ and } M_{ij} = \begin{pmatrix}1 & u_{ij}\\ u_{ij} & v_{ij}\end{pmatrix}\ \forall(i,j)\in N_x.$$

If the noise is multiplicative, i.e., the distances are equal to

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

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 (5.58) is

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

In [12] the authors introduce a gradient local search phase that refines the solution obtained by projecting in the two dimensional space the solution of the two SDP relaxation (5.58). The idea is to move each sensor location in the opposite direction of that of the gradient of the sum of error squared function. In particular, the maximum likelihood estimation is the solution of the unconstrained optimization problem

$$\begin{array}{rcl}
X^* \in \arg\min\limits_X f(X) & := & \sum_{(k,j)\in N_a} \frac{1}{\sigma_{kj}^2}(\|a_k - x_j\| - \bar{D}_{kj})^2 \\
& & + \sum_{(i,j)\in N_x} \frac{1}{\sigma_{ij}^2}(\|x_i - x_j\| - \bar{D}_{ij})^2.
\end{array} \quad (5.59)$$

Let the gradient be ∂f_{x_j} for sensor x_j. This gradient can be computed in a distributed way since it relates only to sensors and anchors connected to sensor x_j. The location of sensor x_j is updated in the following way:

$$x_j = x_j - \alpha\partial f_{x_j}^T$$

where α is the step size. This updating rule improves the localization obtained by relaxation (5.58). 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 $\boldsymbol{SDP}$ approach to $\boldsymbol{SNL}$ is presented in [70]. This successful technique uses the $\boldsymbol{EDM}$ model and solves the $\boldsymbol{SNL}$ without using an $\boldsymbol{SDP}$ solver. It efficiently finds high accuracy solutions of large problems.

As in [69, 35, 36], using the positions of the anchors is postponed till after the corresponding $\boldsymbol{EDMC}$ problem is done. A first *elimination phase* finds faces of the $\boldsymbol{SDP}$ cone that correspond to faces of the $\boldsymbol{EDM}$ cone that contain cliques in the $\boldsymbol{SNL}$ 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 are postponed. Once the appropriate face of proper dimension is found, then the second *substitution phase* solves for all the missing distances in the $\boldsymbol{EDM}$ at once. Then, the third *finalize phase* rotates the anchors to their (approximate) original positions. Extremely large problems can be solved to high accuracy. Current tests with random data (with no noise) solve problems of order $n = 100,000, m = 9$ on a laptop in 5 minutes to 16 decimals accuracy.

The facial reduction approach is closely (dually) related to the geometric build-up, e.g., see the formulas in [10] and the algorithms in [38, 102]. The connection is through the $\boldsymbol{EDMC}$ problem and the factorization $PP^T = B = \mathcal{K}^\dagger(D)$.

5.3 Weaker SNL Formulations

Another more recent stream of research is the one that aims at defining different relaxations, maybe weaker than the ones considered up to this point, but faster to be solved. In [96] a second order cone programming ($\boldsymbol{SOCP}$) relaxation has been introduced. It derives from a different reformulation of the sensor network localization problem. Let $\mathcal{G}$ be the graph having $n+m$ nodes corresponding to the sensors and anchors, and let $\mathcal{A}$ be the set of edges connecting all the nodes (both sensors and anchors) where distance measures are available. Then the SNL problem can be formulated as:

$$\begin{array}{ll} \min_{x_1,\dots,x_n,y_{ij}} & \displaystyle\sum_{(i,j)\in\mathcal{A}} y_{ij} \\ \text{s.t.} & y_{ij} = \|x_i - x_j\|^2, \quad \forall (i,j) \in \mathcal{A} \end{array}$$

that can be relaxed in

$$\begin{array}{ll} \min_{x_1,\dots,x_n,y_{ij}} & \displaystyle\sum_{(i,j)\in\mathcal{A}} y_{ij} \\ \text{s.t.} & y_{ij} \geq \|x_i - x_j\|^2, \quad \forall (i,j) \in \mathcal{A} \end{array} \tag{5.60}$$

that is an $\boldsymbol{SOCP}$. This relaxation is always weaker than the SDP relaxation (5.48) (where all the weights are equal to one). As for the $\boldsymbol{SDP}$ 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 $\boldsymbol{SOCP}$ (5.60), there exists a unique set $\mathcal{B} \subseteq \mathcal{A}$ of constraints that are active in all the solutions, namely

$$\|x_i - x_j\| = y_{ij}, \forall \text{ solutions } x_1, \dots, x_n, (y_{ij})_{(i,j)\in\mathcal{A}} \text{ of (5.60) } \Leftrightarrow (i,j) \in \mathcal{B}. \tag{5.61}$$

Any interior solution satisfies (5.61) and satisfies strictly all the other constraints of (5.60). Let

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

In [96] it is proved that all the points with $i \in \mathcal{M}_{\mathcal{B}}$ belong to the convex hull of the points for which the distances are exact, namely satisfy

$$x_i \in \text{conv}\{x_j\}_{j\in\mathcal{N}_{\mathcal{B}}(i)}, i \in \mathcal{M}_{\mathcal{B}}. \tag{5.62}$$

Furthermore, each connected component of $G_{\mathcal{B}} = (\mathcal{M}_{\mathcal{B}} \cup \{1,\dots,m\}, \mathcal{B})$ contains at least an anchor and for every $i \in \{1,\dots,n\}$, x_i is invariant over all the solutions of (5.60) if and only if $i \in \mathcal{M}_{\mathcal{B}}$. If a particular solution of (5.60) is considered, that is called the analytic center solution, i.e., the interior solution that maximizes

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

over all the interior solutions, then

$$x_j \in \text{conv}\{x_j\}_{j\in\mathcal{N}_{\mathcal{A}}(i)}.$$

If the distances contain errors, in [96] it is shown that if the distance error are small, then $(x_i)_{i\in\mathcal{M}_{\mathcal{B}}}$ in a solution of (5.60) has small error that grows proportionally to the square root of the distance error. Relaxation (5.60) can be solved faster than the $\boldsymbol{SDP}$ relaxation, and in [96] a smoothing coordinate gradient descent method is proposed that is very effective in solving problem (5.60). The author also suggests the use of problem (5.60) as a problem preprocessor or combined with the $\boldsymbol{SDP}$ method.

In the same stream of research, in [80] a sum of squares (SOS) relaxation is proposed, starting from the polynomial formulation (5.51) (with all $\gamma_{ij} = 1$) of the SNL Problem. The idea in [80] 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 $\boldsymbol{SDP}$. 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 Problem $\boldsymbol{SNL}$ introduced in [80] is

$$\begin{array}{rll} f^*_{sos} := & \max & \gamma \\ & \text{s.t.} & f(X) - \gamma = m_2(X)^T W m_2(X) \\ & & W \succeq 0. \end{array} \tag{5.63}$$

If the distances are exact, the SOS relaxation (5.63) 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:

$$f(X) = \sum_{(i,j)\in N_x} \left\{ (\|x_i - x_j\|^2 - \bar{D}_{ij}) + \frac{1}{|S_j|} \sum_{k:(k,j)\in N_a} (\|x_j - a_k\|^2 - \bar{D}_{kj})^2 \right\}, \tag{5.64}$$

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

$$\begin{array}{rll} f^{**}_{sos} = & \max & \gamma \\ & \text{s.t.} & f(X) - \gamma = \displaystyle\sum_{(i,j)\in N_x} m_2(x_i, x_j)^T W_{ij} m_2(x_i, x_j) \\ & & W_{ij} \succeq 0,\ (i,j) \in N_x. \end{array} \tag{5.65}$$

In this formulation, the size of W_{ij} is equal to $(d+1)(2d+1)$ which is independent from n, and the total number of decision variables is $O(d^4|A|)$, with $d = 2$ if we are in the plane. Also in this case, if the distances are exact, the SOS relaxation (5.63) 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 [98] the $\boldsymbol{SDP}$ approach is further relaxed. The authors propose two new $\boldsymbol{SDP}$ 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:

$$(NSDP) \quad \begin{array}{ll} \min & \langle \mathbf{0}, Z \rangle \\ \text{s.t.} & Z_{(1,2),(1,2)} = I_2 \\ & W \circ \mathcal{K}_A(Z) = W \circ D \\ & Z^i = Z_{(1,2,i,N_i)(1,2,i,N_i)} \succeq 0 \quad \forall i \end{array} \tag{5.66}$$

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 (5.66) can be relaxed for inexact distances in:

$$(NSDPOP) \qquad \begin{array}{ll} \min & \mathrm{trace}\,(J(W \circ (U+V))) \\ \text{s.t.} & Z_{(1,2),(1,2)} = I_2 \\ & W \circ (\mathcal{K}_A(Z) - U + V) = W \circ \bar{D} \\ & Z^i = Z_{(1,2,i,N_i)(1,2,i,N_i)} \succeq 0, \quad \forall i \\ & U, V \geq 0. \end{array} \tag{5.67}$$

The second relaxation is an edge-based relaxation, as in [80]:

$$(ESDP) \qquad \begin{array}{ll} \min & \langle \mathbf{0}, Z \rangle \\ \text{s.t.} & Z_{(1,2),(1,2)} = I_2 \\ & W \circ \mathcal{K}_A(Z) = W \circ \bar{D} \\ & Z_{(1,2,i,j)(1,2,i,j)} \succeq 0 \qquad \forall (i,j) \in N_x. \end{array} \tag{5.68}$$

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 into

$$(ESDPOP) \qquad \begin{array}{ll} \min & \mathrm{trace}\,(J(W \circ (U+V))) \\ \text{s.t.} & Z_{(1,2),(1,2)} = I_2 \\ & W \circ (\mathcal{K}_A(Z) - U + V) = W \circ \bar{D} \\ & Z_{(1,2,i,j)(1,2,i,j)} \succeq 0, \quad \forall (i,j) : W_{ij} > 0 \\ & U, V \geq 0. \end{array} \tag{5.69}$$

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 (5.68), (5.66) and (5.40) (that we denote by $\boldsymbol{SDP}$):

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

However, problem (5.40) has $(n+2)^2$ variables and $|N_x| + |N_a|$ constraints, while problem (5.66) has at most $4 + 2n + \sum_i |N_i|^2$ variables and $|N_x| + |N_a|$ constraints and problem (5.68) has $4 + 3n + |N_x|$ variables and also $|N_x| + |N_a|$ constraints. Therefore, problems (5.66) and (5.68) can be solved much faster, since in general $4 + 2n + \sum_i |N_i|^2$ and $4 + 3n + |N_x|$ are smaller than $(n+2)^2$. Furthermore, the two relaxations (5.66) and (5.68), although weaker than (5.40), preserve some interesting theoretical properties of relaxation (5.40). Indeed, relaxation (5.66) 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 (5.68), in [98] 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 (5.68) satisfies

$$\bar{Y}_{ii} - \|\bar{x}_i\|^2 = 0, \quad i \in \{1, \ldots, n\}, \tag{5.70}$$

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 (5.68). Furthermore, although weaker than relaxation (5.40), relaxation (5.68) is stronger than the ***SOCP*** relaxation introduced in [96]. To solve problem (5.68) in [98], it turns out that solving the dual is much faster than solving directly problem (5.68), and a primal solution can be easily derived via the complementarity conditions. Furthermore, the quality of the solution of problem (5.68) does not depend on the quantity and location of anchors, as for example in SPASELOC.

The two edge based relaxations (5.68) and (5.69) are again considered in [82], and the authors prove that, given a solution of problem (5.68), the trace criterion (5.70) 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 (5.69) (a counterexample is provided). In order to recover this interesting property, the authors define a "robust" version of relaxation (5.69) that mantains 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

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

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

$$\begin{array}{ll} \min & -\sum_{(i,j)\in N_x} \ln\det(Z_{(1,2,i,j)(1,2,i,j)}) - \sum_{i=1}^{m} \ln\det\begin{pmatrix} I & x_i \\ x_i^T & y_{ii} \end{pmatrix} \\ \text{s.t.} & |W \circ (\mathcal{K}_A(Z) - \bar{D})| \leq W \circ \rho \\ & Z_{(1,2),(1,2)} = I_2 \end{array} \qquad (5.71)$$

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 (5.70) 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 (5.69), gives a comparable accuracy, and it is highly parallelizable, 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 $\boldsymbol{SNL}$ problem. Many algorithms that are specific for $\boldsymbol{SNL}$ use a distributed approach, $\boldsymbol{SDP}$ and $\boldsymbol{SOCP}$ relaxations, and $\boldsymbol{SDP}$ (parallel) solvers. The efficiency for these algorithms has improved from solving instances with about $n = 100$s of nodes in seconds to instances with $n = 1000$s to an accuracy of several decimals, see e.g. the software at URLs:
www.math.nus.edu.sg/~mattohkc/SNLSDP.html
www.stanford.edu/~yyye/Col.html
www.convexoptimization.com/dattorro/sensor_network_localization.html Recently, a different approach that does not rely on a $\boldsymbol{SDP}$ solver is given in [70], where 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 $\boldsymbol{FPDG}$ problems. Exploiting sparsity and parallelization has just begun. We can expect new efficient algorithms for larger classes of problems.

References

[1] S. AL-HOMIDAN. *Hybrid methods for optimization problems with positive semidefinite matrix constraints*. PhD thesis, University of Dundee, 1993.

[2] S. AL-HOMIDAN and R. FLETCHER. Hybrid methods for finding the nearest Euclidean distance matrix. In *Recent advances in nonsmooth optimization*, pages 1–17. World Sci. Publishing, River Edge, NJ, 1995.

[3] S. AL-HOMIDAN and H. WOLKOWICZ. Approximate and exact completion problems for Euclidean distance matrices using semidefinite programming. *Linear Algebra Appl.*, 406:109–141, 2005.

[4] A. ALFAKIH, A. KHANDANI, and H. WOLKOWICZ. Solving Euclidean distance matrix completion problems via semidefinite programming. *Comput. Optim. Appl.*, 12(1-3):13–30, 1999. Computational optimization—a tribute to Olvi Mangasarian, Part I.

[5] A.Y. ALFAKIH. On dimensional rigidity of bar-and-joint frameworks. *Discrete Appl. Math.*, 155(10):1244–1253, 2007.

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

[7] L. ASIMOW and B. ROTH. The rigidity of graphs. *Trans. Amer. Math. Soc.*, 245:279–289, 1978.

[8] L. ASIMOW and B. ROTH. The rigidity of graphs. II. *J. Math. Anal. Appl.*, 68(1):171–190, 1979.

[9] J. ASPNES, T. EREN, D.K. GOLDENBERG, A.S. MORSE, W. WHITELEY, Y.R. YANG, B.D.O. ANDERSON, and P.N. BELHUMEUR. Semidefinite programming approaches for sensor network localization with noisy distance measurements. *IEEE Transactions on Automation Science and Engineering*, 3(4):360–371, October 2006.

[10] M. BAKONYI and C.R. JOHNSON. The Euclidean distance matrix completion problem. *SIAM J. Matrix Anal. Appl.*, 16(2):646–654, 1995.

[11] C.C. BALÁZS, K. JOSEF, JÜRGEN P., and ROLAND W. On the automation of similarity information maintenance in flexible query answering systems. In *DEXA*, pages 130–140, 2004.

[12] P. BISWAS, T.-C. LIAN, T.-C. WANG, and Y. YE. Semidefinite programming based algorithms for sensor network localization. *ACM Trans. Sen. Netw.*, 2(2):188–220, 2006.

[13] P. BISWAS, T.-C. LIANG, Y. YE, K-C. TOH, and T.-C. WANG. Semidefinite programming approaches for sensor network localization with noisy distance measurements. *IEEE Transactions onAutomation Science and Engineering*, 3(4):360–371, October 2006.

[14] P. BISWAS, K-C. TOH, and Y. YE. A distributed sdp approach for large-scale noisy anchor-free graph realization with applications to molecular conformation. *SIAM J. Sci. Comput.*, 30(3):1251–1277, 2008.

[15] P. BISWAS and Y. YE. Semidefinite programming for ad hoc wireless sensor network localization. In *IPSN '04: Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 46–54, New York, NY, USA, 2004. ACM.

[16] 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*, volume 82 of *Nonconvex Optim. Appl.*, pages 69–84. Springer, New York, 2006.

[17] A. BJORCK. *Numerical Methods for Least Squares Problems.* SIAM, Philadelphia, 1996.

[18] J. BLITZER, K.Q. WEINBERGER, L.K. SAUL, and F.C.N. PEREIRA. Hierarchical Distributed Representations for Statistical Language Modeling. *Advances in Neural Information Processing Systems*, 17:185–192, 2005. A Bradford Book.

[19] L.M. BLUMENTHAL. *Theory and applications of distance geometry.* Chelsea Publishing Co., New York, second edition, 1970.

[20] 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(3):196–206, 1999.

[21] E. CANDES and M. WAKIN. An introduction to compressive sensing. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.

[22] 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(4):1102–1128, 2006.

[23] A. CASSIOLI. *Global optimization of highly multimodal problems.* PhD thesis, Universita di Firenze, Dipartimento di sistemi e informatica, Via di S.Marta 3, 50139 Firenze, Italy, 2008.

[24] K. CHAKRABARTY and S.S. IYENGAR. Springer, London, 2005.

[25] R. CONNELLY. Rigidity and energy. *Invent. Math.*, 66(1):11–33, 1982.

[26] R. CONNELLY. On generic global rigidity. In *Applied geometry and discrete mathematics*, volume 4 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 147–155. Amer. Math. Soc., Providence, RI, 1991.

[27] R. CONNELLY. Rigidity. In *Handbook of convex geometry, Vol. A, B*, pages 223–271. North-Holland, Amsterdam, 1993.

[28] R. CONNELLY. Generic global rigidity. *Discrete Comput. Geom.*, 33(4):549–563, 2005.

[29] R. CONNELLY and M. SLOUGHTER. Realizability of graphs. Technical report, Dept. of Math., Cornell University, Ithaca, NY, 2004.

[30] J.A. COSTA, N. PATWARI, and III A.O. HERO. Distributed weighted-multidimensional scaling for node localization in sensor networks. *ACM Trans. Sen. Netw.*, 2(1):39–64, 2006.

[31] G.M. CRIPPEN. Chemical distance geometry: Current realization and future projection. *Journal of Mathematical Chemistry*, 6(1):307–324, 1991.

[32] G.M. CRIPPEN and T.F. HAVEL. *Distance geometry and molecular conformation*, volume 15 of *Chemometrics Series.* Research Studies Press Ltd., Chichester, 1988.

[33] J. DATTORRO. *Convex Optimization & Euclidean Distance Geometry.* Meboo Publishing, USA, 2005.

[34] J. DATTORRO. Equality relating Euclidean distance cone to positive semidefinite cone. *Linear Algebra Appl.*, 428(11-12):2597–2600, 2008.

[35] Y. DING, N. KRISLOCK, J. QIAN, and H. WOLKOWICZ. Sensor network localization, Euclidean distance matrix completions, and graph realization. *Optimization and Engineering*, to appear(CORR 2006-23, to appear), 2006.

[36] Y. DING, N. KRISLOCK, J. QIAN, and H. WOLKOWICZ. Sensor network localization, Euclidean distance matrix completions, and graph realization. In *Proceedings of MELT08, San Francisco*, pages 129–134, 2008.

[37] L. DOHERTY, K.S.J. PISTER, and EL. Convex position estimation in wireless sensor networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1655–1663, 2001.

[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(3):321–333, 2003.

[39] A.W.M. DRESS and T.F. HAVEL. The fundamental theory of distance geometry. In *Computer aided geometric reasoning, Vol. I, II (Sophia-Antipolis, 1987)*, pages 127–169. INRIA, Rocquencourt, 1987.

[40] T. EREN, D. GOLDENBERG, W. WHITLEY, Y. YANG, A. MORSE, B. ANDERSON, and P. BELHEUMER. Rigidity, computation, and randomization of network localization. In *Proc. IEEE Conf. Computer Comm. (Infocom '04)*, 2004.

[41] 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 in network localization, 2004. IEEE INFOCOM.

[42] R.W. FAREBROTHER. Three theorems with applications to Euclidean distance matrices. *Linear Algebra Appl.*, 95:11–16, 1987.

[43] V.F. FERRARIO, C. SFORZA, A. MIANI JR, and G. SERRAO. A three-dimensional evaluation of human facial asymmetry. *J. Anat.*, 186:103–110, 1995.

[44] H.N. GABOW and H.H. WESTERMANN. Forests, frames, and games: algorithms for matroid sums and applications. *Algorithmica*, 7(5-6):465–497, 1992.

[45] D. GALE. Neighboring vertices on a convex polyhedron. In *Linear inequalities and related system*, volume 38 of *Annals of Mathematics Studies*, pages 255–263. Princeton University Press, Princeton, NJ, 1956.

[46] M.R. GAREY and D.S. JOHNSON. *Computers and Intractability: A guide to the theory of NP–completeness.* Freeman, San Francisco, 1979.

[47] H. GLUCK. Almost all simply connected closed surfaces are rigid. In *Geometric Topology (Proc. Conf., Park City, Utah, 1974)*, volume 438 of *Lecture Notes in Mathematics*, pages 225–239. Springer-Verlag, Berlin, 1975.

[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(4):589–600, 1990.

[49] S.J. GORTLER, A.D. HEALY, and D.P. THURSTON. Characterizing generic global rigidity, 2007. arXiv/0710.0926.

[50] J. C. GOWER. Properties of Euclidean and non-Euclidean distance matrices. *Linear Algebra Appl.*, 67:81–97, 1985.

[51] J.C. GOWER. Euclidean distance geometry. *Math. Sci.*, 7(1):1–14, 1982.

[52] H. GRASSMANN. *Extension theory*, volume 19 of *History of Mathematics*. American Mathematical Society, Providence, RI, 2000. Translated from the 1896 German original and with a foreword, editorial notes and supplementary notes by Lloyd C. Kannenberg.

[53] J. GRAVER, B. SERVATIUS, and H. SERVATIUS. *Combinatorial rigidity*. American Mathematical Society, Grad. Stud. Math, Vol 2, 1993.

[54] M. GRÖTSCHEL, L. LOVÁSZ, and A. SCHRIJVER. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981.

[55] T.F. HAVEL. Metric matrix embedding in protein structure calculations, nmr spectra analysis, and relaxation theory, 2003.

[56] T.L. HAYDEN, J. WELLS, W-M. LIU, and P. TARAZAGA. The cone of distance matrices. *Linear Algebra Appl.*, 144:153–169, 1991.

[57] B. HENDRICKSON. *The Molecule Problem: Determining Conformation from Pairwise Distances*. PhD thesis, Cornell University, 1990.

[58] B. HENDRICKSON. Conditions for unique graph realizations. *SIAM J. Comput.*, 21:65–84, 1992.

[59] B. HENDRICKSON. The molecule problem: exploiting structure in global optimization. *SIAM J. Optim.*, 5(4):835–857, 1995.

[60] 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(1):3–21, 2003. Dedicated to Professor J. B. Rosen on his 80th birthday.

[61] B. JACKSON and T. JORDÁN. Connected rigidity matroids and unique realizations of graphs. *J. Combin. Theory Ser. B*, 94(1):1–29, 2005.

[62] B. JACKSON, T. JORDÁN, and Z. SZABADKA. Globally linked pairs of vertices in equivalent realizations of graphs. *Discrete Comput. Geom.*, 35:493–512, 2006.

[63] H. JIN. *Scalable Sensor Localization Algorithms for Wireless Sensor Networks*. PhD thesis, Toronto University, Toronto, Ontario, Canada, 2005.

[64] C.R. JOHNSON, B. KROSCHEL, and H. WOLKOWICZ. An interior-point method for approximate positive semidefinite completions. *Comput. Optim. Appl.*, 9(2):175–190, 1998.

[65] C.R. JOHNSON and P. TARAZAGA. Connections between the real positive semidefinite and distance matrix completion problems. *Linear Algebra Appl.*, 223/224:375–391, 1995. Special issue honoring Miroslav Fiedler and Vlastimil Pták.

[66] D.S. KIM. *Sensor Network Localization Based on Natural Phenomena*. PhD thesis, Dept, Electr. Eng. and Comp. Sc., MIT, 2006.

[67] S. KIM, M. KOJIMA, and H. WAKI. Exploiting sparsity in SDP relaxation for sensor network localization. *SIAM J. Optim.*, 20(1):192–215, 2009.

[68] S. KIM, M. KOJIMA, H. WAKI, and M. YAMASHITA. A sparse version of full semidefinite programming relaxation for sensor network localization problems. Technical Report B-457, Department of Mathematical and Computing Sciences Tokyo Institute of Technology, Oh-Okayama, Meguro, Tokyo 152-8552, 2009.

[69] N. KRISLOCK, V. PICCIALLI, and H. WOLKOWICZ. Robust semidefinite programming approaches for sensor network localization with anchors. Technical Report CORR 2006-12, University of Waterloo, Waterloo, Ontario, 2006. URL:orion.uwaterloo.ca/~hwolkowi/henry/reports/ABSTRACTS.html#sensorKPW.

[70] N. KRISLOCK and H. WOLKOWICZ. Explicit sensor network localization using semidefinite representations and clique reductions. Technical Report CORR 2009-04, University of Waterloo, Waterloo, Ontario, 2009. Available at URL: www.optimization-online.org/DB_HTML/2009/05/2297.html.

[71] G. LAMAN. On graphs and rigidity of plane skeletal structures. *Journal of Engineering mathematics*, 4:331–340, 1970.

[72] M. LAURENT. Cuts, matrix completions and graph rigidity. *Math. Programming*, 79:255–284, 1997.

[73] M. LAURENT. A tour d'horizon on positive semidefinite and Euclidean distance matrix completion problems. In *Topics in semidefinite and interior-point methods (Toronto, ON, 1996)*, volume 18 of *Fields Inst. Commun.*, pages 51–76. Amer. Math. Soc., Providence, RI, 1998.

[74] L. LOVÁSZ and Y. YEMINI. On generic rigidity in the plane. *SIAM J. Alg. Disc. Meth.*, 3:91–98, 1982.

[75] 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)*, pages 151–168. Amer. Math. Soc., Providence, RI, 1996.

[76] J.J. MORÉ and Z. WU. Global continuation for distance geometry problems. *SIAM J. Optim.*, 7(3):814–836, 1997.

[77] J.J. MORÉ and Z. WU. Distance geometry optimization for protein structures. *J. Global Optim.*, 15(3):219–234, 1999.

[78] J.J. MORÉ and Z. WU. Distance geometry optimization for protein structures. *J. Global Optim.*, 15(3):219–234, 1999.

[79] S. NAWAZ. *Anchor free localization for ad-hoc wireless sensor networks.* PhD thesis, University of New South Wales, 2008.

[80] J. NIE. Sum of squares method for sensor network localization. *Comput. Optim. Appl.*, 43:151–179, 2009.

[81] P.M. PARDALOS, D. SHALLOWAY, and G. XUE, editors. *Global minimization of nonconvex energy functions: molecular conformation and protein folding*, volume 23 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science.* American Mathematical Society, Providence, RI, 1996. Papers from the DIMACS Workshop held as part of the DIMACS Special Year on Mathematical Support for Molecular Biology at Rutgers University, New Brunswick, New Jersey, March 20–21, 1995.

[82] 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.

[83] M.V. RAMANA, L. TUNCEL, and H. WOLKOWICZ. Strong duality for semidefinite programming. *SIAM J. Optim.*, 7(3):641–662, 1997.

[84] B. RECHT, M. FAZEL, and P. PARRILO. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. Technical report, Caltech, Pasadena, California, 2007.

[85] K. ROMER. *Time Synchronization and Localization in Sensor Networks.* PhD thesis, ETH Zurich, 2005.

[86] C. SAVARESE, J. RABAEY, , and J. BEUTEL. Locationing in distributed ad-hoc wireless sensor networks. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 2037–2040, 2001.

[87] A. SAVVIDES, C.-C. HAN, and M.B. STRIVASTAVA. Dynamic fine-grained localization in ad-hoc networks of sensors. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179, New York, NY, USA, 2001. ACM.

[88] A. SAVVIDES, C.C. HAN, and M.B. SRIVASTAVA. Dynamic fine grained localization in ad-hoc sensor networks. In *Proceedings of the Fifth International Conference on Mobile Computing and Networking (Mobicom 2001)*, pages 166–179, 2001.

[89] J.B. SAXE. Embeddability of weighted graphs in k-space is strongly NP-hard. In *Seventeenth Annual Allerton Conference on Communication, Control, and Computing, Proceedings of the Conference held in Monticello, Ill., October 10–12, 1979*, pages xiv+1036, Urbana, 1979. University of Illinois Department of Electrical Engineering. Proceedings of the International School of Physics "Enrico Fermi", LXX*.

[90] I.J. SCHOENBERG. Remarks to Maurice Frechet's article: Sur la definition axiomatique d'une classe d'espaces vectoriels distancies applicables vectoriellement sur l'espace de Hilbert. *Ann. Math.*, 36:724–732, 1935.

[91] A.M. SO and Y. YE. Theory of semidefinite programming for sensor network localization. *Math. Program.*, 109(2):367–384, 2007.

[92] W.S. TORGERSON. Multidimensional scaling. I. Theory and method. *Psychometrika*, 17:401–419, 1952.

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

[94] M.W. TROSSET. Applications of multidimensional scaling to molecular conformation. *Computing Science and Statistics*, 29:148–152, 1998.

[95] M.W. TROSSET. Distance matrix completion by numerical optimization. *Comput. Optim. Appl.*, 17(1):11–22, 2000.

[96] P. TSENG. Second-order cone programming relaxation of sensor network localization. *SIAM J. on Optimization*, 18(1):156–185, 2007.

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

[98] Z. WANG, S. ZHENG, S. BOYD, and Y. YE. Further relaxations of the semidefinite programming approach to sensor network localization. *SIAM J. Optim.*, 19(2):655–673, 2008.

[99] 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*, page 106, New York, NY, USA, 2004. ACM.

[100] W. WHITELEY. Matroids and rigid structures. In *Matroid applications*, volume 40 of *Encyclopedia Math. Appl.*, pages 1–53. Cambridge Univ. Press, Cambridge, 1992.

[101] H. WOLKOWICZ, R. SAIGAL, and L. VANDENBERGHE, editors. *Handbook of semidefinite programming*. International Series in Operations Research & Management Science, 27. Kluwer Academic Publishers, Boston, MA, 2000. Theory, algorithms, and applications.

[102] 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(4):661–673, 2007.

[103] J-M. YOON, Y. GAD, and Z. WU. Mathematical modeling of protein structure using distance geometry. Technical report, University of Houston, 2000.

[104] F. Z. ZHANG. On the best Euclidean fit to a distance matrix. *Beijing Shifan Daxue Xuebao*, 4:21–24, 1987.

[105] 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.

Index