Euclidean Distance Matrices and Applications

Nathan Krislock[1] and Henry Wolkowicz[2]

[1] University of Waterloo, Department of Combinatorics and Optimization, Waterloo, Ontario N2L 3G1, Canada, ngbkrislock@uwaterloo.ca
[2] University of Waterloo, Department of Combinatorics and Optimization, Waterloo, Ontario N2L 3G1, Canada, hwolkowicz@uwaterloo.ca

1 Introduction

Over the past decade, Euclidean distance matrices, or EDMs, have been receiving increased attention for two main reasons. The first reason is that the many applications of EDMs, such as molecular conformation in bioinformatics, dimensionality reduction in machine learning and statistics, and especially the problem of wireless sensor network localization, have all become very active areas of research. The second reason for this increased interest is the close connection between EDMs and semidefinite matrices. Our recent ability to solve semidefinite programs, SDPs, efficiently means we can now also solve many problems involving EDMs efficiently.

1.1 Background

Distance geometry and Euclidean distance matrices

Two foundational papers in the area of Euclidean distance matrices are [105] and [120]. The topic was further developed with the series of papers [63, 64, 65], followed by [43, 54]. For papers on the Euclidean distance matrix completion problem and the related semidefinite completion problem, see the classic paper on semidefinite completion [67], and follow-up papers [19] and [78]; also see [88] on the topic of the complexity of these completion problems. More on the topic of uniqueness of Euclidean distance matrix completions can be found in the papers [8, 9]. The cone of Euclidean distance matrices and its geometry is described in, for example, [11, 59, 71, 111, 112]. Using semidefinite optimization to solve Euclidean distance matrix problems is studied in [2, 4]. Further theoretical results are given in [10, 13]. Books and survey papers containing a treatment of Euclidean distance matrices include, for example, [31, 44, 87], and most recently [3]. The topic of rank minimization for Euclidean distance matrix problems is discussed in, for example, [34, 35, 55, 56, 99, 100].

Graph realization and graph rigidity

The complexity of graph realization in a fixed dimension was determined to be NP-hard by [103, 119]. For studies on graph rigidity, see, for example, [6, 7, 12, 23, 24, 39, 73, 74, 76], and the references therein. Graph rigidity for sensor network localization is studied as graph rigidity with some nodes being grounded or anchored; see, for example, [53, 109]. Semidefinite optimization techniques have also been applied to graph realization and graph rigidity problems; see, for example, [20, 107, 108].

Sensor network localization

While semidefinite relaxations were discussed earlier in [4] for Euclidean distance matrix problems, the first semidefinite relaxations specialized for the sensor network localization problem were proposed by [47]. The paper by [28] followed with what is now called the Biswas-Ye semidefinite relaxation of sensor network localization problem. As problems with only a few hundred sensors could be solved directly using the Biswas-Ye relaxation, [77] and [36] proposed the scalable SpaseLoc semidefinite-based build-up algorithm which solves small subproblems using the Biswas-Ye semidefinite relaxation, locating a few sensors at a time. In the follow-up paper [26] propose regularization and refinement techniques for handling noisy problems using the Biswas-Ye semidefinite relaxation. In order to handle larger problems, a distributed method was proposed in [29] which clusters points together in small groups, solves the smaller subproblems, then stitches the clusters together; see also the PhD thesis [25]. To allow the solution of larger problems, [114] proposed further relaxations of the sensor network localization problem in which they do not insist that the full n-by-n matrix of the Biswas-Ye relaxation be positive semidefinite, but rather only that certain submatrices of this matrix be positive semidefinite; the most successful of these further relaxations is the so-called edge-based SDP relaxation, or ESDP. A noise-aware robust version of the ESDP relaxation called ρ-ESDP is proposed by [97] and which is solved with their Log-barrier Penalty Coordinate Gradient Descent (LPCGD) method. Using techniques from [57], it is shown in [80] how to form an equivalent sparse version of the full Biswas-Ye relaxation, called SFSDP. The sparsity in the SFSDP formulation can then be exploited by a semidefinite optimization solver, allowing the solution of noisy instances of the sensor network localization problem with up to 18000 sensors and 2000 anchors to high accuracy in under ten minutes; see [81]. Most recently, it was shown in [84] how to use facial reduction to solve a semidefinite relaxation of the sensor network localization problem; the resulting algorithm is able to solve noiseless problems with up to 100,000 sensors and 4 anchors to high accuracy in under six minutes on a laptop computer. The connection between the sensor network localization problem and the Euclidean distance matrix problem is described in [3]. In particular, the connection uses facial reduction based on the clique of anchors.

Other relaxations have also been studied; [113] considers a second-order cone (SOC) relaxation of the sensor network localization problem, while [96] studies the sum-of-squares (SOS) relaxation of this problem.

For more applied approaches and general heuristics, see, for example, [32, 33, 37, 40, 91, 92, 95, 102, 110, 118]. A older survey paper on wireless sensor networks is [1]; for a recent book on wireless ad hoc and sensor networks, see [90].

The complexity of the sensor network localization problem is discussed in [16, 17]. References for the single sensor localization problem are, for example, [21, 22].

Molecular conformation

An early algorithmic treatment of molecular conformation is [69] in which they give their bound embedding algorithm EMBED. This paper was then followed by the book [42]; a review paper [41] provides an update three years after the publication of this book. A personal historical perspective is given in [70].

Other algorithmic developments followed, including: a divide-and-conquer algorithm called ABBIE based on identifying rigid substructures [72]; an alternating projection approach [58]; a global smoothing continuation code called DGSOL [93, 94]; a geometric build-up algorithm [48, 49, 116, 117]; an extended recursive geometric build-up algorithm [50]; a difference of convex functions (d.c.) optimization algorithm [15]; a method based on rank-reducing perturbations of the distance matrix that maintain desired structures [52]; an algorithm for solving a distance matrix based, large-scale, bound constrained, non-convex optimization problem called StrainMin [68].

Recently, semidefinite optimization approaches to the molecular conformation problem have been studied in [25, 27, 89].

1.2 Outline

We begin in Section 2 by discussing some preliminaries and introducing notation. In Section 3, we explain the close connection to semidefinite matrices and the many recent results arising from this special relationship. In Section 5, we will look at some popular applications, and we especially focus on the problem of sensor network localization (SNL).

2 Preliminaries

We let $\mathcal{S}^n$ be the space of $n \times n$ real symmetric matrices. A *Euclidean distance matrix* (EDM) is a matrix D for which

$$\exists p_1,\dots,p_n \in \mathbb{R}^r, \text{ such that } D_{ij} = \|p_i - p_j\|_2^2, \quad \forall i,j = 1,\dots,n. \tag{1}$$

The set of Euclidean distance matrices is denoted $\mathcal{E}^n$. If D is an EDM, then the smallest integer r for which condition (1) is possible is called the *embedding dimension* of D, and is denoted $\operatorname{embdim}(D)$.

2.1 Further Notation

The adjoint of a linear transformation T is denoted T^* and satisfies $\langle Tx, y\rangle = \langle x, T^*y\rangle, \forall x, y$. For a given matrix M and vector v, we let $\operatorname{diag}(M)$ denote the vector formed from the diagonal of M. The adjoint $\operatorname{Diag}(v) = \operatorname{diag}^*(v)$ is the diagonal matrix formed with v as its diagonal. For a given symmetric matrix B, we let $B[\alpha]$ denote the principal submatrix formed using the rows/columns from the index set α.

The cone of symmetric positive semidefinite (resp. definite) matrices is denoted by $\mathcal{S}^n_+$ (resp. $\mathcal{S}^n_{++}$). The cone $\mathcal{S}^n_+$ is closed and convex and induces the Löwner partial order:

$$A \succeq B \text{ (resp. } A \succ B), \quad \text{ if } A - B \in \mathcal{S}^n_+ \text{ (resp. } A - B \in \mathcal{S}^n_{++}).$$

A convex cone $F \subseteq K$ is a *face of the cone* K, denoted $F \trianglelefteq K$, if

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

If $F \trianglelefteq K$, but is not equal to K, we write $F \lhd K$. 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. If $F \trianglelefteq K$, the *conjugate face* of F is $F^c := F^\perp \cap K^*$, where $K^* := \{x : \langle x, y\rangle \geq 0, \forall y \in K\}$ is the *dual cone* of the cone K; in fact it is easy to show that $F^c \trianglelefteq K^*$ and that if $\phi \in F^c$, then $F \trianglelefteq K \cap \{\phi\}^\perp$ (see, for example, [82]). A face $F \trianglelefteq K$ is an *exposed face* if it is the intersection of K with a hyperplane. It is well known that $\mathcal{S}^n_+$ is *facially exposed*: every face $F \trianglelefteq \mathcal{S}^n_+$ is exposed. For more on faces of convex cones, the interested reader is encouraged to refer to [98, 101, 104].

3 Euclidean distance matrices and semidefinite matrices

The connection between EDMs and semidefinite matrices is well known. This has been studied at length in, e.g., [78, 86, 87]. There is a natural relationship between the sets $\mathcal{S}^n_+$ and $\mathcal{E}^n$. Suppose that $D \in \mathcal{E}^n$ is realized by the points $p_1,\dots,p_n \in \mathbb{R}^r$. Let

$$P := \begin{bmatrix} p_1^T \\ \vdots \\ p_n^T \end{bmatrix} \in \mathbb{R}^{n\times r} \quad \text{and} \quad Y := PP^T = \left(p_i^T p_j\right)_{i,j=1}^n.$$

The matrix $Y = PP^T$ is known as the *Gram matrix* of the points $p_1, \ldots, p_n$. Then, $\forall i, j \in \{1, \ldots, n\}$, we have

$$\begin{aligned} D_{ij} &= \|p_i - p_j\|_2^2 \\ &= p_i^T p_i + p_j^T p_j - 2p_i^T p_j \\ &= Y_{ii} + Y_{jj} - 2Y_{ij}. \end{aligned}$$

Therefore, $D = \mathcal{K}(Y)$, where $\mathcal{K} \colon \mathcal{S}^n \to \mathcal{S}^n$ is the linear operator[3] defined as

$$\mathcal{K}(Y)_{ij} := Y_{ii} + Y_{jj} - 2Y_{ij}, \quad \text{for } i, j = 1, \ldots, n.$$

Equivalently, we can define $\mathcal{K}$ by

$$\mathcal{K}(Y) := \operatorname{diag}(Y)e^T + e \operatorname{diag}(Y)^T - 2Y, \tag{2}$$

where $e \in \mathbb{R}^n$ is the vector of all ones. From this simple observation, we can see that $\mathcal{K}$ maps the cone of semidefinite matrices, $\mathcal{S}^n_+$, onto $\mathcal{E}^n$. That is, $\mathcal{K}(\mathcal{S}^n_+) = \mathcal{E}^n$. In addition, since $\mathcal{S}^n_+$ is a convex cone, we immediately get that $\mathcal{E}^n$ is a convex cone.

Mappings between EDM and SDP

There are several linear transformations that map between $\mathcal{E}^n$ and the cone of semidefinite matrices. We first present some useful properties of $\mathcal{K}$ in (2) and related transformations and their adjoints follow; see, e.g., [84]. For $Y \in \mathcal{S}^n$ we let $\mathcal{D}_e(Y) := \operatorname{diag}(Y)e^T + e \operatorname{diag}(Y)^T$; by abuse of notation, we also let $\mathcal{D}_e(y) := ye^T + ey^T$, for $y \in \mathbb{R}^n$. Then, our main operator of interest is

$$\mathcal{K}(Y) := \mathcal{D}_e(Y) - 2Y.$$

The adjoints are

$$\mathcal{D}_e^*(D) = 2 \operatorname{Diag}(De), \qquad \mathcal{K}^*(D) = 2(\operatorname{Diag}(De) - D).$$

We also have an explicit representation for the Moore-Penrose generalized inverse:

$$\boxed{\mathcal{K}^\dagger(D) = -\frac{1}{2} J \operatorname{offDiag}(D) J}$$

where $J := I - \frac{1}{n}ee^T$, $\operatorname{offDiag}(D) := D - \operatorname{Diag}(\operatorname{diag}(D))$. In addition,

$$\begin{aligned} \mathcal{S}^n_H &:= \{D \in \mathcal{S}^n : \operatorname{diag}(D) = 0\} & \qquad \mathcal{K}\mathcal{K}^\dagger(D) &= \operatorname{offDiag}(D) \\ \mathcal{S}^n_C &:= \{Y \in \mathcal{S}^n : Ye = 0\} & \qquad \mathcal{K}^\dagger\mathcal{K}(Y) &= JYJ \end{aligned}$$

[3] Early appearances of this linear operator are in [105, 120]. However, the use of the notation $\mathcal{K}$ for this linear operator dates back to [43] wherein κ was used due to the fact that the formula for $\mathcal{K}$ is basically the *cosine law* ($c^2 = a^2 + b^2 - 2ab\cos(\gamma)$). Later on, in [78], K was used to denote this linear operator.

$\mathcal{S}_H^n$ is called the *hollow subspace*, while $\mathcal{S}_C^n$ is called the *centered subspace*.

$$\operatorname{range}(\mathcal{K}) = \mathcal{S}_H^n \qquad \operatorname{null}(\mathcal{K}^\dagger) = \operatorname{range}(\operatorname{Diag})$$
$$\operatorname{range}(\mathcal{K}^\dagger) = \mathcal{S}_C^n \qquad \operatorname{null}(\mathcal{K}) = \operatorname{range}(\mathcal{D}_e)$$

$$\mathcal{K}(\mathcal{S}_C^n) = \mathcal{S}_H^n \qquad \mathcal{K}^\dagger(\mathcal{S}_H^n) = \mathcal{S}_C^n$$
$$\mathcal{K}(\mathcal{S}_+^n \cap \mathcal{S}_C^n) = \mathcal{E}^n \qquad \mathcal{K}^\dagger(\mathcal{E}^n) = \mathcal{S}_+^n \cap \mathcal{S}_C^n$$

$$\operatorname{embdim}(D) = \operatorname{rank} \mathcal{K}^\dagger(D), \quad \text{for } D \in \mathcal{E}^n$$
$$\|\mathcal{K}\|_F := \max_{0 \neq Y \in \mathcal{S}^n} \frac{\|\mathcal{K}(Y)\|_F}{\|Y\|_F} = 2\sqrt{n}$$

We let $\mathcal{T} := \mathcal{K}^\dagger$. Then $\mathcal{K}$ and $\mathcal{T}$ map between the centered subspace, $\mathcal{S}_C^n$, and the hollow subspace, $\mathcal{S}_H^n$. Since $\operatorname{int}(\mathcal{S}_C^n \cap \mathcal{S}_+^n) = \emptyset$, we can have problems with constraint qualifications and unbounded optimal sets. To avoid this ([2, 4]), we define $\mathcal{K}_V \colon \mathcal{S}^{n-1} \to \mathcal{S}^n$ by

$$\mathcal{K}_V(X) := \mathcal{K}(VXV^T), \tag{3}$$

where $V \in \mathbb{R}^{n \times (n-1)}$ is full column rank and satisfies $V^T e = 0$. Then $\mathcal{K}_V(\mathcal{S}_+^{n-1}) = \mathcal{E}^n$. And, $\mathcal{K}_V(X) = D \in \mathcal{E}^n$ implies that VXV^T is the corresponding Gram matrix.

Alternatively, see [2], we can use $\mathcal{L} \colon \mathcal{S}^{n-1} \to \mathcal{S}^n$

$$\mathcal{L}(X) := \begin{bmatrix} 0 & \operatorname{diag}(X)^T \\ \operatorname{diag}(X) & \mathcal{K}(X) \end{bmatrix}. \tag{4}$$

And, as with $\mathcal{K}_V$, we get $\mathcal{L}(\mathcal{S}_+^{n-1}) = \mathcal{E}^n$.

3.1 Properties of $\mathcal{K}$

We now include further useful properties of the linear map $\mathcal{K}$.

Translational invariance and the null space of $\mathcal{K}$

The null space of $\mathcal{K}$ is closely related to the translational invariance of distances between a set of points. Suppose that $P \in \mathbb{R}^{n \times r}$ and that

$$\hat{P} = P + ev^T, \quad \text{for some } v \in \mathbb{R}^r;$$

that is, $\hat{P}$ is the matrix formed by translating every row of P by the vector v. Clearly, P and $\hat{P}$ generate the same Euclidean distance matrix, so we have $\mathcal{K}(PP^T) = \mathcal{K}(\hat{P}\hat{P}^T)$. Note that

$$\begin{aligned} \hat{P}\hat{P}^T &= PP^T + Pve^T + ev^TP + ev^Tve^T \\ &= PP^T + \mathcal{D}_e(y), \end{aligned}$$

where $y := Pv + \frac{v^Tv}{2}e$. Thus, we have

$$0 = \mathcal{K}(\hat{P}\hat{P}^T - PP^T) = \mathcal{K}(\mathcal{D}_e(y)),$$

so $\mathcal{D}_e(y) \in \operatorname{null}(\mathcal{K})$. Thus, if there are no *anchors* among the points, then we do not have to worry about translations of the set of points.

Rotational invariance of the Gram matrix

Suppose that $P \in \mathbb{R}^{n\times r}$ and that

$$\hat{P} = PQ, \quad \text{for some } Q \in \mathbb{R}^{r\times r} \text{ orthogonal;}$$

that is, $\hat{P}$ is the matrix formed by rotating/reflecting each row of P by the same orthogonal transformation. Again, we clearly have that P and $\hat{P}$ generate the same Euclidean distance matrix, but we can say more. If Y is the Gram matrix of P and $\hat{Y}$ is the Gram matrix of $\hat{P}$, then

$$\hat{Y} = \hat{P}\hat{P}^T = PQQ^TP^T = PP^T = Y.$$

Therefore, we have that the Gram matrix is invariant under orthogonal transformations of the points. Thus, when using a semidefinite matrix Y to represent a Euclidean distance matrix D with $D = \mathcal{K}(Y)$, orthogonal transformations will not affect Y nor D.

The maps $\mathcal{K}$ and $\mathcal{K}^\dagger$ as bijections

Consider $\mathcal{K}$ and $\mathcal{K}^\dagger$ restricted to the subspaces $\mathcal{S}^n_C$ and $\mathcal{S}^n_H$, respectively. Then, the above expressions for the ranges implies that the map $\mathcal{K}\colon \mathcal{S}^n_C \to \mathcal{S}^n_H$ is a bijection and $\mathcal{K}^\dagger\colon \mathcal{S}^n_H \to \mathcal{S}^n_C$ is its inverse.

If we consider $\mathcal{K}$ and $\mathcal{K}^\dagger$ restricted to the convex cones $\mathcal{S}^n_+ \cap \mathcal{S}^n_C$ and $\mathcal{E}^n$, respectively, then the map $\mathcal{K}\colon \mathcal{S}^n_+ \cap \mathcal{S}^n_C \to \mathcal{E}^n$ is a bijection and $\mathcal{K}^\dagger\colon \mathcal{E}^n \to \mathcal{S}^n_+ \cap \mathcal{S}^n_C$ is its inverse. Note that we could use a rotation and replace the face $\mathcal{S}^n_+ \cap \mathcal{S}^n_C \lhd \mathcal{S}^n_+$, as is done above with the $\mathcal{K}_V$ linear transformation defined in equation (3).

Embedding dimension and a theorem of Schoenberg

We now give the following much celebrated theorem of Schoenberg [105] (also found in the later paper by Young and Householder [120]) that provides a method for testing if a matrix is a Euclidean distance matrix and a method determining the embedding dimension of a Euclidean distance matrix; see also [4].

Theorem 3.1 ([105, 120]) *A matrix $D \in \mathcal{S}^n_H$ is a Euclidean distance matrix if and only if $\mathcal{K}^\dagger(D)$ is positive semidefinite. Furthermore, if $D \in \mathcal{E}^n$, then*

$$\operatorname{embdim}(D) = \operatorname{rank} \mathcal{K}^\dagger(D) \le n-1. \quad \square$$

This means that given $D \in \mathcal{E}^n$, we can find the Gram matrix $B = \mathcal{K}^\dagger(D)$ and the full rank factorization $PP^T = B$. Then the points, p_j, given by the rows of P satisfy $p_j \in \mathbb{R}^t$, for $j = 1, \ldots, n$. Moreover, t is necessarily less than n.

4 The Euclidean distance matrix completion problem

Following [19], we say that an n-by-n matrix D is a *partial Euclidean distance matrix* if every entry of D is either "specified" or "unspecified", $\operatorname{diag}(D) = 0$, and every fully specified principal submatrix of D is a EDM. Note that this definition implies that every specified entry of D is nonnegative. In addition, if every fully specified principal submatrix of D has embedding dimension less than or equal to r, then we say that D is a partial EDM in $\mathbb{R}^r$.

Associated with an n-by-n partial EDM D is a *weighted undirected graph* $G = (N, E, \omega)$ with node set $N := \{1, \ldots, n\}$, edge set

$$E := \{ij : i \neq j, \text{ and } D_{ij} \text{ is specified}\},$$

and edge weights $\omega \in \mathbb{R}^E_+$ with $\omega_{ij} = \sqrt{D_{ij}}$, for all $ij \in E$. We say that H is the 0–1 *adjacency matrix* of G if $H \in \mathcal{S}^n$ with

$$H_{ij} = \begin{cases} 1, & ij \in E \\ 0, & ij \notin E. \end{cases}$$

The *Euclidean distance matrix completion (EDMC) problem* asks to find a completion of a partial Euclidean distance matrix D; that is, if $G = (N, E, \omega)$ is the weighted graph associated with D, the EDMC problem can be posed as

$$\begin{array}{ll} \text{find} & \hat{D} \in \mathcal{E}^n \\ \text{s.t.} & \hat{D}_{ij} = D_{ij}, \ \forall ij \in E. \end{array} \tag{5}$$

Letting $H \in \mathcal{S}^n$ be the 0–1 adjacency matrix of G, the EDMC problem can be stated as

$$\begin{array}{ll} \text{find} & \hat{D} \in \mathcal{E}^n \\ \text{s.t.} & H \circ \hat{D} = H \circ D, \end{array} \tag{6}$$

where "$\circ$" represents the component-wise (or *Hadamard*) matrix product.

Using the linear map $\mathcal{K}$, we can substitute $\hat{D} = \mathcal{K}(Y)$, where $Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C$, in the EDMC problem (6) to obtain the equivalent problem

$$\begin{array}{ll} \text{find} & Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C \\ \text{s.t.} & H \circ \mathcal{K}(Y) = H \circ D. \end{array} \tag{7}$$

4.1 The low-dimensional EDM completion problem

If D is a partial EDM in $\mathbb{R}^r$, one is often interested in finding a Euclidean distance matrix completion of D that has embedding dimension r. The *low-dimensional Euclidean distance matrix completion problem* is

$$\begin{array}{rc} \text{find} & \hat{D} \in \mathcal{E}^n \\ \text{s.t.} & H \circ \hat{D} = H \circ D \\ & \operatorname{embdim}(\hat{D}) = r, \end{array} \tag{8}$$

where H is the 0–1 adjacency matrix of the graph G associated with D.

Using the linear map $\mathcal{K}$, we can state the low-dimensional EDMC problem (8) as the following rank constrained SDP:

$$\begin{array}{rc} \text{find} & Y \in \mathcal{S}^n_+ \\ \text{s.t.} & H \circ \mathcal{K}(Y) = H \circ D \\ & Ye = 0 \\ & \operatorname{rank}(Y) = r. \end{array} \tag{9}$$

Note that the constraint $Ye = 0$ means that there is no positive definite feasible solution. This means that standard interior point methods cannot properly handle this problem without some modification, e.g., the use of K_V given above in equation (3).

4.2 Chordal EDM completions

Let G be a graph and C be a cycle in the graph. We say that C has a *chord* if there are two vertices on C that are connected by an edge which is not contained in C. Note that it is necessary that a cycle with a chord have length more than three. The graph G is called *chordal* if every cycle of the graph with length three or more has a *chord*. In the landmark paper [67], they show the strong result that any partial semidefinite matrix with a chordal graph has a semidefinite completion; moreover, if a graph is not chordal, then there exists a partial semidefinite matrix with that graph, but having no semidefinite completion.

Due to the strong connection between EDMs and semidefinite matrices, it is not surprising that the result of chordal semidefinite completions of [67] extends to the case of chordal EDMC. Indeed, see [19]:

1. any partial Euclidean distance matrix in $\mathbb{R}^r$ with a chordal graph can be completed to a distance matrix in $\mathbb{R}^r$;
2. every nonchordal graph has a partial Euclidean distance matrix that does not admit any distance matrix completions;
3. if the graph G of a partial Euclidean distance matrix D in $\mathbb{R}^r$ is chordal, then the completion of D is unique if and only if

$$\operatorname{rank}\left(\begin{bmatrix} 0 & e^T \\ e & D[S] \end{bmatrix}\right) = r + 2, \qquad \text{for all minimal vertex separators } S \text{ of } G.$$

A set of vertices S in a graph G is a *minimal vertex separator* if removing the vertices S from G separates some vertices u and v in G, and no proper subset of S separates u and v. It is discussed in, for example, [85] how the *maximum cardinality search (MCS)* can be used to test in linear time if a graph G is chordal; moreover, [85] show that MCS can also be used to compute all minimal vertex separators of a chordal graph in linear time. See also [8, 9], for example, for more on the topic of uniqueness of EDMC.

4.3 Corresponding Graph Realization Problems

Suppose that we are given an $n \times n$ partial EDM $\bar{D}$, where only the elements $\bar{D}_{ij}$, for all $ij \in E$, are known. In addition, suppose every fully specified principal submatrix of $\bar{D}$ has an embedding dimension less or equal to r. We let $\mathcal{G} = (N, E, \omega)$ be the corresponding simple weighted graph on the node set $N = \{1, \dots, n\}$ whose edge set E corresponds to the known entries of $\bar{D}$, with edge weights $\bar{D}_{ij} = \omega_{ij}^2$, for all $ij \in E$. The *graph realization* problem consists of finding a mapping $p\colon N \to \mathbb{R}^r$, with $p_i \in \mathbb{R}^r$ for all $i \in N$, such that $\|p_i - p_j\| = \omega_{ij}$, for all $ij \in E$.

We note that a clique $C \subseteq N$ of the graph $\mathcal{G}$ defines a complete subgraph of $\mathcal{G}$ and this corresponds to a known principal submatrix of $\bar{D}$. Cliques play a significant role in a facial reduction algorithm for the SNL problem that we describe in Section 4.8 below.

We note here the deep connection between the Euclidean distance matrix completion problem and the problem of *graph realization*. Let $N := \{1, \dots, n\}$. Given a graph $G = (N, E, \omega)$ with edge weights $\omega \in \mathbb{R}^E_+$, the graph realization problem asks to find a mapping $p\colon N \to \mathbb{R}^r$ such that

$$\|p_i - p_j\| = \omega_{ij}, \qquad \text{for all } ij \in E;$$

in this case, we say that G has an *r-realization*, p. Clearly the graph realization problem is equivalent to the problem of Euclidean distance matrix completion, and the problem of the r-realizability of a weighted graph is equivalent to the low-dimensional Euclidean distance matrix completion problem.

A related problem is that of graph rigidity. Again, let $N := \{1, \dots, n\}$. An unweighted graph $G = (N, E)$ together with a mapping $p\colon N \to \mathbb{R}^r$ is called a *framework* (also *bar framework*) in $\mathbb{R}^r$, and is denoted by (G, p). Frameworks (G, p) in $\mathbb{R}^r$ and (G, q) in $\mathbb{R}^s$ are called *equivalent* if

$$\|p_i - p_j\| = \|q_i - q_j\|, \qquad \text{for all } ij \in E.$$

Furthermore, p and q are called *congruent* if

$$\|p_i - p_j\| = \|q_i - q_j\|, \qquad \text{for all } i, j = 1, \dots, n. \tag{10}$$

Note that condition (10) can be stated as

$$\mathcal{K}(PP^T) = \mathcal{K}(QQ^T),$$

where $P \in \mathbb{R}^{n\times r}$ and $Q \in \mathbb{R}^{n\times s}$ are defined as

$$P := \begin{bmatrix} p_1^T \\ \vdots \\ p_n^T \end{bmatrix} \qquad \text{and} \qquad Q := \begin{bmatrix} q_1^T \\ \vdots \\ q_n^T \end{bmatrix}.$$

Thus, if $P^T e = Q^T e = 0$, then we have $PP^T = QQ^T$, which implies that:

- $P\bar{Q} = \begin{bmatrix} Q \ 0 \end{bmatrix}$, for some orthogonal $\bar{Q} \in \mathbb{R}^r$, if $r \geq s$;
- $\begin{bmatrix} P \ 0 \end{bmatrix} \bar{Q} = Q$, for some orthogonal $\bar{Q} \in \mathbb{R}^s$, if $s \geq r$.

A framework (G, p) in $\mathbb{R}^r$ is called *globally rigid* in $\mathbb{R}^r$ if all equivalent frameworks (G, q) in $\mathbb{R}^r$ satisfy condition (10). Similarly, a framework (G, p) in $\mathbb{R}^r$ is called *universally rigid* in $\mathbb{R}^r$ if, for all $s = 1, \ldots, n-1$, all equivalent frameworks (G, q) in $\mathbb{R}^s$ satisfy condition (10). Note that a framework (G, p) in $\mathbb{R}^r$ corresponds to the pair (H, D), where H is the 0–1 adjacency matrix of G, and D is a Euclidean distance matrix $\operatorname{embdim}(D) \leq r$. Therefore, the framework (G, p) given by (H, D) is globally rigid if

$$H \circ \hat{D} = H \circ D \quad \Rightarrow \quad \hat{D} = D,$$

for all $\hat{D} \in \mathcal{E}^n$ with $\operatorname{embdim}(\hat{D}) \leq r$; equivalently, (G, p) is globally rigid if

$$H \circ \mathcal{K}(Y) = H \circ D \quad \Rightarrow \quad \mathcal{K}(Y) = D,$$

for all $Y \in \mathcal{S}_+^n \cap \mathcal{S}_C^n$ with $\operatorname{rank}(Y) \leq r$. Moreover, the framework (G, p) given by (H, D) is universally rigid if

$$H \circ \hat{D} = H \circ D \quad \Rightarrow \quad \hat{D} = D,$$

for all $\hat{D} \in \mathcal{E}^n$; equivalently, (G, p) is universally rigid if

$$H \circ \mathcal{K}(Y) = H \circ D \quad \Rightarrow \quad \mathcal{K}(Y) = D,$$

for all $Y \in \mathcal{S}_+^n \cap \mathcal{S}_C^n$.

Graph realization and graph rigidity is a vast area of research, so we keep our discussion brief. More information can be found in, for example, [39, 73, 107], the recent survey [3], and the references therein.

4.4 Low-dimensional EDM completion is NP-hard

We now discuss the complexity of the low-dimensional Euclidean distance matrix completion decision problem (8) (equivalently, problem (9)). It was independently discovered by [103] and [119] that the problem of graph embeddability with integer edge weights, in $r = 1$ or $r = 2$ dimensions, is NP-complete by reduction from the NP-complete problem PARTITION. The PARTITION problem is defined as follows: Given a set of S of n integers, determine if there is a partition of S into two sets $S_1, S_2 \subseteq S$ such that the sum of the integers in S_1 equals the sum of the integers in S_2. PARTITION is one of Karp's 21 NP-complete problems in his landmark paper [79].

Theorem 4.1 ([103, Theorem 3.2]) *The problem of* 1*-embeddability of graphs with integer weights is NP-complete.* □

Furthermore, by showing that for any r, r-embeddability of $\{1,2\}$-weighted graphs is NP-hard, [103] proves that the problem of r-embeddability of integer weighted graphs is strongly NP-hard.

In practice, the so-called *unit disk graphs* are used; these graphs have realization in some Euclidean space satisfying their edge-weights such that the distance between vertices that are not connected is greater than some *radio range* R, and R is greater than all the edge weights. Thus, vertices are connected if and only if they are within radio range. However, see [18], the realizability of unit disk graphs is, in fact, NP-hard (again, by reduction from PARTITION). In the [18] proof, they again work with cycles, which are typically not uniquely realizable. In applications, one is often most interested in unit disk graphs that have a unique realization. However, it is shown in [16, 17] that there is no efficient algorithm for solving the unit disk graph localization problem, even if that graph has a unique realization, unless RP = NP (RP is the class of randomized polynomial time solvable problems). Problems in RP can be solved in polynomial time with high probability using a randomized algorithm. See also [88], for example, for more on the topic of the complexity of Euclidean distance matrix completion and related problems.

Due to these hardness results for the low-dimensional Euclidean distance matrix problem, we turn to convex relaxations which can be solved efficiently, but may not solve our original problem.

4.5 SDP relaxation of the low-dimensional EDM completion problem

The semidefinite relaxation of the low-dimensional EDM completion problem (9) is given by relaxing the hard $\operatorname{rank}(Y) = r$ constraint. Thus, we have the following tractable convex relaxation

$$\begin{array}{rc} \text{find} & Y \in \mathcal{S}^n_+ \\ \text{s.t.} & H \circ \mathcal{K}(Y) = H \circ D \\ & Ye = 0. \end{array} \tag{11}$$

This semidefinite relaxation essentially allows the points to move into $\mathbb{R}^k$, where $k > r$. That is, if a solution Y of problem (11) has $\operatorname{rank}(Y) = k > r$, then we have found a Euclidean distance matrix completion of D with embedding dimension k; this is even possible if D has a completion with embedding dimension r, or even if D has a *unique* completion with embedding dimension r.

We can view this relaxation as a Lagrangian relaxation of the low-dimensional EDM completion problem.

Proposition 4.2 ([83, Prop. 2.48]) *Relaxation* (11) *is the Lagrangian relaxation of Problem* (9). □

Duality of the SDP relaxation

Problem (11) is equivalent to

$$\begin{array}{rl} \text{minimize} & 0 \\ \text{subject to} & H \circ \mathcal{K}(Y) = H \circ D \\ & Ye = 0 \\ & Y \in \mathcal{S}^n_+ . \end{array}$$

The Lagrangian of this problem is given by

$$\begin{aligned} L(Y, \Lambda, v) &= \langle \Lambda, H \circ \mathcal{K}(Y) - H \circ D \rangle + \langle v, Ye \rangle \\ &= \langle \mathcal{K}^*(H \circ \Lambda), Y \rangle - \langle H \circ \Lambda, D \rangle + \left\langle ev^T, Y \right\rangle \\ &= \left\langle \mathcal{K}^*(H \circ \Lambda) + \frac{1}{2}\left(ev^T + ve^T\right), Y \right\rangle - \langle H \circ \Lambda, D \rangle \\ &= \left\langle \mathcal{K}^*(H \circ \Lambda) + \frac{1}{2}\mathcal{D}_e(v), Y \right\rangle - \langle H \circ \Lambda, D \rangle . \end{aligned}$$

Therefore, the Lagrangian dual problem is

$$\sup_{\Lambda, v} \inf_{Y \in \mathcal{S}^n_+} L(Y, \Lambda, v),$$

which is equivalent to

$$\sup \left\{ - \langle H \circ \Lambda, D \rangle : \mathcal{K}^*(H \circ \Lambda) + \frac{1}{2}\mathcal{D}_e(v) \succeq 0 \right\}.$$

From this dual problem, we obtain the following partial description of the conjugate face of the minimal face of the EDM completion problem (11).

Proposition 4.3 *Let* $F := \text{face}(\mathcal{F})$*, where*

$$\mathcal{F} := \left\{ Y \in \mathcal{S}^n_+ : H \circ \mathcal{K}(Y) = H \circ D, Ye = 0 \right\}.$$

If $\mathcal{F} \neq \emptyset$*, then*

$$\text{face} \left\{ S \in \mathcal{S}^n_+ : S = \mathcal{K}^*(H \circ \Lambda) + \frac{1}{2}\mathcal{D}_e(v), \langle H \circ \Lambda, D \rangle = 0 \right\} \trianglelefteq F^c. \quad \square$$

Using Proposition 4.3, we obtain the following partial description of the minimal face of the EDM completion problem (11).

Corollary 4.4 *If* $\mathcal{F} := \left\{ Y \in \mathcal{S}^n_+ : H \circ \mathcal{K}(Y) = H \circ D, Ye = 0 \right\} \neq \emptyset$ *and there exists* $S \succeq 0$ *such that*

$$S = \mathcal{K}^*(H \circ \Lambda) + \frac{1}{2}\mathcal{D}_e(v) \quad \textit{and} \quad \langle H \circ \Lambda, D \rangle = 0,$$

for some $\Lambda \in \mathcal{S}^n$ *and* $v \in \mathbb{R}^n$*, then*

$$\text{face}(\mathcal{F}) \trianglelefteq \mathcal{S}^n_+ \cap \{S\}^\perp . \quad \square$$

For example, we can apply Corollary 4.4 as follows. Let $\Lambda := 0$ and $v := e$. Then,

$$\mathcal{K}^*(H \circ \Lambda) + \frac{1}{2}\mathcal{D}_e(v) = ee^T \succeq 0, \quad \text{and} \quad \langle H \circ \Lambda, D \rangle = 0.$$

Thus,

$$\operatorname{face}\left\{Y \in \mathcal{S}^n_+ : H \circ \mathcal{K}(Y) = H \circ D, Ye = 0\right\} \trianglelefteq \mathcal{S}^n_+ \cap \left\{ee^T\right\}^\perp = V\mathcal{S}^{n-1}_+ V^T,$$

where $\begin{bmatrix} V & \frac{1}{\sqrt{n}}e \end{bmatrix} \in \mathbb{R}^{n \times n}$ is orthogonal. Note that we have $V^TV = I$ and $VV^T = J$, where J is the orthogonal projector onto $\{e\}^\perp$. Therefore, Problem (11) is equivalent to the reduced problem

$$\begin{array}{ll} \text{find} & Z \in \mathcal{S}^{n-1}_+ \\ \text{s.t.} & H \circ \mathcal{K}_V(Z) = H \circ D, \end{array}$$

where $\mathcal{K}_V : \mathcal{S}^{n-1} \to \mathcal{S}^n$ is defined as in equation (3).

4.6 Rank minimization heuristics for the EDM completion problem

In order to encourage having a solution of the semidefinite relaxation with low rank, the following heuristic has been suggested by [115] and used with great success by [26] on the sensor network localization problem. The idea is that we can try to “flatten” the graph associated with a partial Euclidean distance matrix by pushing the nodes of the graph away from each other as much as possible. This flattening of the graph then corresponds to reducing the rank of the semidefinite solution of the relaxation. Geometrically, this makes a lot of sense, and [115] gives this nice analogy: a loose string on the table can occupy two dimensions, but the same string pulled taut occupies just one dimension.

Therefore, we would like to maximize the objective function

$$\sum_{i,j=1}^n \|p_i - p_j\|^2 = e^T\mathcal{K}(PP^T)e,$$

where

$$P := \begin{bmatrix} p_1^T \\ \vdots \\ p_n^T \end{bmatrix},$$

subject to the distance constraints holding. Moreover, if we include the constraint that $P^Te = 0$, then

$$\begin{aligned}
e^T\mathcal{K}(PP^T)e &= \left\langle ee^T, \mathcal{K}(PP^T)\right\rangle \\
&= \left\langle \mathcal{K}^*(ee^T), PP^T\right\rangle \\
&= \left\langle 2(\mathrm{Diag}(ee^Te) - ee^T), PP^T\right\rangle \\
&= \left\langle 2(nI - ee^T), PP^T\right\rangle \\
&= \left\langle 2nI, PP^T\right\rangle - \left\langle ee^T, PP^T\right\rangle \\
&= 2n \cdot \mathrm{trace}(PP^T).
\end{aligned}$$

Note that

$$\mathrm{trace}(PP^T) = \sum_{i=1}^n \|p_i\|^2,$$

so pushing the nodes away from each other is equivalent to pushing the nodes away from the origin, under the assumption that the points are centred at the origin. Normalizing this objective function by dividing by the constant $2n$, substituting $Y = PP^T$, and relaxing the rank constraint on Y, we obtain the following *regularized* semidefinite relaxation of the low-dimensional Euclidean distance matrix completion problem:

$$\begin{array}{ll}
\text{maximize} & \mathrm{trace}(Y) \\
\text{subject to} & H \circ \mathcal{K}(Y) = H \circ D \\
& Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C.
\end{array} \tag{12}$$

It is very interesting to compare this heuristic with the *nuclear norm* rank minimization heuristic that has received much attention lately. This nuclear norm heuristic has had much success and obtained many practical results for computing minimum-rank solutions of linear matrix equations (for example, finding the exact completion of low-rank matrices); see, for example, [14, 34, 35, 55, 100] and the recent review paper [99].

The *nuclear norm* of a matrix $X \in \mathbb{R}^{m\times n}$ is given by

$$\|X\|_* := \sum_{i=1}^k \sigma_i(X),$$

where $\sigma_i(X)$ is the i^{th} largest singular value of X, and $\mathrm{rank}(X) = k$. The nuclear norm of a symmetric matrix $Y \in \mathcal{S}^n$ is then given by

$$\|Y\|_* = \sum_{i=1}^n |\lambda_i(Y)|.$$

Furthermore, for $Y \in \mathcal{S}^n_+$, we have $\|Y\|_* = \mathrm{trace}(Y)$.

Since we are interested in solving the *rank minimization* problem,

$$\begin{array}{ll}
\text{minimize} & \mathrm{rank}(Y) \\
\text{subject to} & H \circ \mathcal{K}(Y) = H \circ D \\
& Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C,
\end{array} \tag{13}$$

the nuclear norm heuristic gives us the problem

$$\begin{array}{rc} \text{minimize} & \operatorname{trace}(Y) \\ \text{subject to} & H \circ \mathcal{K}(Y) = H \circ D \\ & Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C. \end{array} \tag{14}$$

However, this is geometrically interpreted as trying to bring all the nodes of the graph as close to the origin as possible. Intuition tells us that this approach would produce a solution with a high embedding dimension.

Another rank minimization heuristic that has been considered recently in [56] is the log-det maximization heuristic. There they successfully computed solutions to Euclidean distance matrix problems with very low embedding dimension via this heuristic.

4.7 Nearest EDM Problem

For many applications, we are given an approximate (or partial) EDM $\bar{D}$ and we need to find the *nearest EDM*. Some of the elements of $\bar{D}$ may be exact. Therefore, we can model this problem as the norm minimization problem,

$$\begin{array}{ll} \min & \|W \circ (K(B) - \bar{D})\| \\ \text{s.t.} & K(B)_{ij} = \bar{D}_{ij}, \forall ij \in E, \\ & B \succeq 0, \end{array} \tag{15}$$

where W represents a weight matrix to reflect the accuracy of the data, and E is a subset of the pairs of nodes corresponding to exact data. This model often includes upper and lower bounds on the distances; see, e.g., [4].

We have not specified the norm in (15). The Frobenius norm was used in [4], where small problems were solved; see also [5]. The Frobenius norm was also used in [45, 46], where the EDM was specialized to the sensor network localization, SNL, model. All the above approaches had a difficult time solving large problems. The difficulty was both in the size of the problem and the accuracy of the solutions. It was observed in [46] that the Jacobian of the optimality conditions had many zero singular values at optimality. An explanation of this *degeneracy* is discussed below in Section 4.8.

4.8 Facial reduction

As mentioned in Section 4.7 above, solving large scale nearest EDM problems using SDP is difficult due to the size of the resulting SDP and also due to the difficulty in getting accurate solutions. In particular, the empirical tests in [46] led to the observation [83, 84] that the problems are highly, implicitly degenerate. In particular, if we have a clique in the data, $\alpha \subseteq N$, (equivalently, we have a known principal submatrix of the data $\bar{D}$), then we have the following basic result.

Theorem 4.5 *([84, Thm 2.3]) Let $D \in \mathcal{E}^n$, with embedding dimension r. Let $\bar{D} := D[1\!:\!k] \in \mathcal{E}^k$ with embedding dimension t, and $B := \mathcal{K}^\dagger(\bar{D}) = \bar{U}_B S \bar{U}_B^T$, 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 $\begin{bmatrix} V & \frac{U^T e}{\|U^T e\|} \end{bmatrix} \in \mathcal{M}^{n-k+t+1}$ be orthogonal. Then*

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

Theorem 4.5 implies that the Slater constraint qualification (strict feasibility) fails if the known set of distances contains a clique. Moreover, we explicitly find the expression for the face of feasible semidefinite Gram matrices. This means we have reduced the size of the problem from matrices in $\mathcal{S}^n$ to matrices in $\mathcal{S}^{n-k+t}$. We can continue to do this for all disjoint cliques. Note the equivalences between the cliques, the faces, and the subspace representation for the faces.

The following result shows that we can continue to reduce the size of the problem using two intersecting cliques. All we have to do is find the corresponding subspace representations and calculate the intersection of these subspaces.

Theorem 4.6 *([84, Thm 2.7]) Let $D \in \mathcal{E}^n$ with embedding dimension r and, define the sets of positive integers*

$$\begin{gathered} \alpha_1 := 1\!:\!(\bar{k}_1 + \bar{k}_2), \quad \alpha_2 := (\bar{k}_1 + 1)\!:\!(\bar{k}_1 + \bar{k}_2 + \bar{k}_3) \subseteq 1\!:\!n, \\ k_1 := |\alpha_1| = \bar{k}_1 + \bar{k}_2, \quad k_2 := |\alpha_2| = \bar{k}_2 + \bar{k}_3, \\ k := \bar{k}_1 + \bar{k}_2 + \bar{k}_3. \end{gathered}$$

For $i = 1, 2$, let $\bar{D}_i := D[\alpha_i] \in \mathcal{E}^{k_i}$ with embedding dimension t_i, and $B_i := \mathcal{K}^\dagger(\bar{D}_i) = \bar{U}_i S_i \bar{U}_i^T$, where $\bar{U}_i \in \mathcal{M}^{k_i \times t_i}$, $\bar{U}_i^T \bar{U}_i = I_{t_i}$, $S_i \in \mathcal{S}^{t_i}_{++}$, and $U_i := \begin{bmatrix} \bar{U}_i & \frac{1}{\sqrt{k_i}} e \end{bmatrix} \in \mathcal{M}^{k_i \times (t_i+1)}$. Let t and $\bar{U} \in \mathcal{M}^{k\times(t+1)}$ satisfy

$$\mathcal{R}(\bar{U}) = \mathcal{R}\left(\begin{bmatrix} U_1 & 0 \\ 0 & I_{\bar{k}_3} \end{bmatrix} \right) \cap \mathcal{R}\left(\begin{bmatrix} I_{\bar{k}_1} & 0 \\ 0 & U_2 \end{bmatrix} \right), \text{ with } \bar{U}^T \bar{U} = I_{t+1}.$$

Let $U := \begin{bmatrix} \bar{U} & 0 \\ 0 & I_{n-k} \end{bmatrix} \in \mathcal{M}^{n\times(n-k+t+1)}$ and $\begin{bmatrix} V & \frac{U^T e}{\|U^T e\|} \end{bmatrix} \in \mathcal{M}^{n-k+t+1}$ be orthogonal. Then

$$\bigcap_{i=1}^{2} \operatorname{face} \mathcal{K}^\dagger \left(\mathcal{E}^n(\alpha_i, \bar{D}_i) \right) = \left(U \mathcal{S}_+^{n-k+t+1} U^T \right) \cap \mathcal{S}_C = (UV) \mathcal{S}_+^{n-k+t} (UV)^T. \quad \square$$

Moreover, the intersections of subspaces can be found efficiently and accurately.

Lemma 4.7 *([84, Lemma 2.9]) Let*

$$U_1 := \begin{matrix} & r+1 \\ s_1 & \\ k & \end{matrix}\!\!\begin{bmatrix} U_1' \\ U_1'' \end{bmatrix}, \quad U_2 := \begin{matrix} k \\ s_2 \end{matrix}\!\begin{bmatrix} U_2'' \\ U_2' \end{bmatrix}, \quad \hat{U}_1 := \begin{matrix} s_1 \\ k \\ s_2 \end{matrix}\!\begin{bmatrix} U_1' & 0 \\ U_1'' & 0 \\ 0 & I \end{bmatrix}, \quad \hat{U}_2 := \begin{matrix} s_1 \\ k \\ s_2 \end{matrix}\!\begin{bmatrix} I & 0 \\ 0 & U_2'' \\ 0 & U_2' \end{bmatrix}$$

be appropriately blocked with $U_1'', U_2'' \in \mathcal{M}^{k\times(r+1)}$ *full column rank and* $\mathcal{R}(U_1'') = \mathcal{R}(U_2'')$. *Furthermore, let*

$$\bar{U}_1 := \begin{matrix} s_1 \\ k \\ s_2 \end{matrix}\!\begin{bmatrix} U_1' \\ U_1'' \\ U_2'(U_2'')^\dagger U_1'' \end{bmatrix}, \quad \bar{U}_2 := \begin{matrix} s_1 \\ k \\ s_2 \end{matrix}\!\begin{bmatrix} U_1'(U_1'')^\dagger U_2'' \\ U_2'' \\ U_2' \end{bmatrix}.$$

Then $\bar{U}_1$ *and* $\bar{U}_2$ *are full column rank and satisfy*

$$\mathcal{R}(\hat{U}_1) \cap \mathcal{R}(\hat{U}_2) = \mathcal{R}\left(\bar{U}_1\right) = \mathcal{R}\left(\bar{U}_2\right).$$

Moreover, if $e_{r+1} \in \mathbb{R}^{r+1}$ *is the* $(r+1)^{\text{st}}$ *standard unit vector, and* $U_i e_{r+1} = \alpha_i e$, *for some* $\alpha_i \neq 0$, *for* $i = 1, 2$, *then* $\bar{U}_i e_{r+1} = \alpha_i e$, *for* $i = 1, 2$. $\square$

As mentioned above, a clique α corresponds to both a principal submatrix $\bar{D}[\alpha]$, in the data $\bar{D}$; to a face in $\mathcal{S}^n_+$, and to a subspace. In addition, we can use $\mathcal{K}$ to obtain $B = \mathcal{K}^\dagger(\bar{D}[\alpha]) \succeq 0$ to represent the face and then use the factorization $B = PP^T$ to find a point representation. Using this point representation increases the accuracy of the calculations.

4.9 Minimum norm feasibility problem

The (best) least squares feasible solution was considered in [2]. We can replace $\mathcal{K}$ with $\mathcal{L}$ in (4). We get

$$\begin{array}{lll} \min & \|X\|_F^2 & \\ \text{s.t.} & \mathcal{L}(X)_{ij} = \bar{D}_{ij}, & \forall ij \in E \\ & X \succeq 0. & \end{array} \tag{16}$$

The Lagrangian dual is particularly elegant and can be solved efficiently. In many cases, these (large, sparse case) problems can essentially be solved explicitly.

5 Applications

There are many applications of Euclidean distance matrices, including wireless sensor network localization, molecular conformation in chemistry and bioinformatics, and nonlinear dimensionality reduction in statistics and machine learning. For space consideration, we emphasize sensor network localization. And, in particular, we look at methods that are based on the EDM problem and compare these with the Biswas-Ye SDP relaxation.

5.1 Sensor network localization

The *sensor network localization (SNL)* problem is a low-dimensional Euclidean distance matrix completion problem in which the position of a subset of the nodes is specified. Typically, a wireless ad hoc sensor network consists of n sensors in e.g., a geographical area. Each sensor has wireless communication capability and the ability for some signal processing and networking. Applications abound, for example: military; detection and characterization of chemical, biological, radiological, nuclear, and explosive attacks; monitoring environmental changes in plains, forests, oceans, etc.; monitoring vehicle traffic; providing security in public facilities; etc...

We let $x_1, \dots, x_{n-m} \in \mathbb{R}^r$ denote the unknown *sensor* locations; while $a_1 = x_{n-m+1}, \dots, a_m = x_n \in \mathbb{R}^r$ denotes the known positions of the *anchors/beacons*. Define:

$$X := \begin{bmatrix} x_1^T \\ \vdots \\ x_{n-m}^T \end{bmatrix} \in \mathbb{R}^{(n-m)\times r}; \quad A := \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix} \in \mathbb{R}^{m\times r}; \quad P := \begin{bmatrix} X \\ A \end{bmatrix} \in \mathbb{R}^{n\times r}.$$

We partition the n-by-n partial Euclidean distance matrix as

$$D =: \begin{matrix} & \begin{matrix} n-m & m \end{matrix} \\ \begin{matrix} n-m \\ m \end{matrix} & \begin{bmatrix} D_{11} & D_{12} \\ D_{12}^T & D_{22} \end{bmatrix} \end{matrix}.$$

The sensor network localization problem can then be stated as follows.

> Given: $A \in \mathbb{R}^{m\times r}$; $D \in \mathcal{S}^n$ a partial Euclidean distance matrix satisfying $D_{22} = \mathcal{K}(AA^T)$, with corresponding 0–1 adjacency matrix H;

$$\begin{array}{rl} \text{find} & X \in \mathbb{R}^{(n-m)\times r} \\ \text{s.t.} & H \circ \mathcal{K}\left(PP^T\right) = H \circ D \\ & P = \begin{bmatrix} X \\ A \end{bmatrix} \in \mathbb{R}^{n\times r}. \end{array} \tag{17}$$

Before discussing the semidefinite relaxation of the sensor network localization problem, we first discuss the importance of the anchors, and how we may, in fact, ignore the constraint on P that its bottom block must equal the anchor positions A.

Anchors and the Procrustes problem

For the uniqueness of the sensor positions, a key assumption that we need to make is that the affine hull of the anchors $A \in \mathbb{R}^{m\times r}$ is full-dimensional, $\operatorname{aff}\{a_1, \dots, a_m\} = \mathbb{R}^r$. This further implies that $m \geq r+1$, and A is full column rank.

On the other hand, from the following observations, we see that we can ignore the constraint $P_2 = A$ in problem (17), where $P \in \mathbb{R}^{n\times r}$ is partitioned as

$$P =: \begin{matrix} \\ {\scriptstyle n-m} \\ {\scriptstyle m} \end{matrix} \overset{\scriptstyle r}{\begin{bmatrix} P_1 \\ P_2 \end{bmatrix}}. \tag{18}$$

The distance information is D_{22} in sufficient for completing the partial EDM; i.e., if P satisfies $H \circ \mathcal{K}\left(PP^T\right) = H \circ D$ in problem (17), then

$$\mathcal{K}(P_2P_2^T) = \mathcal{K}(AA^T).$$

Assuming, without loss of generality, that $P_2^Te = 0$ and $A^Te = 0$, we have that $P_2P_2^T = AA^T$. As we now see, this implies that there exists an orthogonal $Q \in \mathbb{R}^r$ such that $P_2Q = A$. Since such an orthogonal transformation does not change the distances between points, we have that PQ is a feasible solution of the sensor network localization problem (17), with sensor positions $X := P_1Q$, i.e., we can safely ignore the positions of the anchors until after the missing distances have been found.

The existence of such an orthogonal transformation follows from the classical Procrustes problem. Given $A, B \in \mathbb{R}^{m\times n}$, solve:

$$\begin{array}{ll} \text{minimize} & \|BQ - A\|_F \\ \text{subject to} & Q^TQ = I. \end{array} \tag{19}$$

The general solution to this problem was first given in [106] (see also [62, 66, 75]).

Theorem 5.1 ([106]) *Let $A, B \in \mathbb{R}^{m\times n}$. Then $Q := UV^T$ is an optimal solution of the Procrustes problem* (19)*, where $B^TA = U\Sigma V^T$ is the singular value decomposition of B^TA.* □

From Theorem 5.1, we have the following useful consequence.

Proposition 5.2 ([83, Prop. 3.2]) *Let $A, B \in \mathbb{R}^{m\times n}$. Then $AA^T = BB^T$ if and only if there exists an orthogonal $Q \in \mathbb{R}^{n\times n}$ such that $BQ = A$.* □

Therefore, the method of discarding the constraint $P_2 = A$ discussed above is justified. This suggests a simple approach for solving the sensor network localization problem. First we solve the equivalent low-dimensional Euclidean distance matrix completion problem,

$$\begin{array}{ll} \text{find} & \bar{Y} \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C \\ \text{s.t.} & H \circ \mathcal{K}\left(\bar{Y}\right) = H \circ D \\ & \operatorname{rank}(\bar{Y}) = r. \end{array} \tag{20}$$

Note that without the anchor constraint, we can now assume that our points are centred at the origin, hence the constraint $\bar{Y} \in \mathcal{S}^n_C$. Factoring $\bar{Y} = PP^T$, for some $P \in \mathbb{R}^{n\times r}$, we then translate P so that $P_2^T e = 0$. Similarly, we translate the anchors so that $A^T e = 0$. We then apply an orthogonal transformation to align P_2 with the anchors positions in A by solving a Procrustes problem using the solution technique in Theorem 5.1. Finally, if we translated to centre the anchors, we simply translate everything back accordingly.

However, as problems with rank constraints are often NP-hard, problem (20) may be very hard to solve. In fact, we saw in Section 4.4 that the general problem (20) can be reduced to the NP-complete problem Partition. Therefore, we now turn to investigating semidefinite relaxations of the sensor network localization problem.

Based on our discussion in this section, the relaxation that immediately comes to mind is the semidefinite relaxation of the low-dimensional Euclidean distance matrix completion problem (9), which is given by relaxing the rank constraint in problem (20); that is, we get the relaxation

$$\begin{array}{rl} \text{find} & \bar{Y} \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C \\ \text{s.t.} & H \circ \mathcal{K}\left(\bar{Y}\right) = H \circ D. \end{array} \tag{21}$$

However, as we will see in the next section, it is possible to take advantage of the structure available in the constraints corresponding to the anchor-anchor distances. We will show how this structure allows us to reduce the size of the semidefinite relaxation.

Semidefinite relaxation of the SNL problem

To get a semidefinite relaxation of the sensor network localization problem, we start by writing problem (17) as,

$$\begin{array}{rl} \text{find} & X \in \mathbb{R}^{(n-m)\times r} \\ \text{s.t.} & H \circ \mathcal{K}\left(\bar{Y}\right) = H \circ D \\ & \bar{Y} = \begin{bmatrix} XX^T & XA^T \\ AX^T & AA^T \end{bmatrix}. \end{array} \tag{22}$$

Next we show that the nonlinear second constraint on $\bar{Y}$, the block-matrix constraint, may be replaced by a semidefinite constraint, a linear constraint, and a rank constraint on $\bar{Y}$.

Proposition 5.3 ([46, 84]) *Let $A \in \mathbb{R}^{m\times r}$ have full column rank, and let $\bar{Y} \in \mathcal{S}^n$ be partitioned as*

$$\bar{Y} =: \begin{array}{c} \scriptstyle n-m \\ \scriptstyle m \end{array} \overset{\begin{array}{cc} \scriptstyle n-m & \scriptstyle m \end{array}}{\begin{bmatrix} \bar{Y}_{11} & \bar{Y}_{12} \\ \bar{Y}_{12}^T & \bar{Y}_{22} \end{bmatrix}}.$$

Then the following hold:

1. *If* $\bar{Y}_{22} = AA^T$ *and* $\bar{Y} \succeq 0$*, then there exists* $X \in \mathbb{R}^{(n-m)\times r}$ *such that*
$$\bar{Y}_{12} = XA^T,$$
and X *is given uniquely by* $X = \bar{Y}_{12}A^{\dagger T}$.
2. *There exists* $X \in \mathbb{R}^{(n-m)\times r}$ *such that* $\bar{Y} = \begin{bmatrix} XX^T & XA^T \\ AX^T & AA^T \end{bmatrix}$ *if and only if* $\bar{Y}$ *satisfies*
$$\left\{ \begin{array}{r} \bar{Y} \succeq 0 \\ \bar{Y}_{22} = AA^T \\ \operatorname{rank}(\bar{Y}) = r \end{array} \right\}. \quad \square$$

Now, by Proposition 5.3, we have that problem (17) is equivalent to

$$\begin{array}{ll} \text{find} & \bar{Y} \in \mathcal{S}^n_+ \\ \text{s.t.} & H \circ \mathcal{K}\left(\bar{Y}\right) = H \circ D \\ & \bar{Y}_{22} = AA^T \\ & \operatorname{rank}(\bar{Y}) = r. \end{array} \tag{23}$$

Relaxing the hard rank constraint, we obtain the semidefinite relaxation of the sensor network localization problem:

$$\begin{array}{ll} \text{find} & \bar{Y} \in \mathcal{S}^n_+ \\ \text{s.t.} & H \circ \mathcal{K}\left(\bar{Y}\right) = H \circ D \\ & \bar{Y}_{22} = AA^T. \end{array} \tag{24}$$

As in Proposition 4.2, this relaxation is equivalent to the Lagrangian relaxation of the sensor network localization problem (23). Moreover, this relaxation essentially allows the sensors to move into a higher dimension. To obtain a solution in $\mathbb{R}^r$, we may either project the positions in the higher dimension onto $\mathbb{R}^r$, or we can try a best rank-r approximation approach.

Further transformations of the SDP relaxation of the SNL problem

From Proposition 5.3, we have that $\bar{Y} \succeq 0$ and $\bar{Y}_{22} = AA^T$ implies that

$$\bar{Y} = \begin{bmatrix} Y & XA^T \\ AX^T & AA^T \end{bmatrix}$$

for some $Y \in \mathcal{S}^{n-m}_+$ and $X = \bar{Y}_{12}A^{\dagger T} \in \mathbb{R}^{(n-m)\times r}$. Now we make the key observation that having anchors in a Euclidean distance matrix problem implies that our feasible points are restricted to a face of the semidefinite cone. This is because, if $\bar{Y}$ is feasible for problem (24), then

$$\bar{Y} = \begin{bmatrix} I & 0 \\ 0 & A \end{bmatrix} \begin{bmatrix} Y & X \\ X^T & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & A \end{bmatrix}^T \in U_A \mathcal{S}^{n-m+r}_+ U_A^T, \tag{25}$$

where

$$U_A := \begin{array}{c} \\ {\scriptstyle n-m} \\ {\scriptstyle m} \end{array} \!\!\begin{array}{c} {\scriptstyle n-m \;\; r} \\ \begin{bmatrix} I & 0 \\ 0 & A \end{bmatrix} \end{array} \in \mathbb{R}^{n\times(n-m+r)}. \tag{26}$$

The fact that we must have

$$\begin{array}{c} \\ {\scriptstyle n-m} \\ {\scriptstyle r} \end{array} \!\!\begin{array}{c} {\scriptstyle n-m \;\; r} \\ \begin{bmatrix} Y & X \\ X^T & I \end{bmatrix} \end{array} \in \mathcal{S}_+^{n-m+r}$$

follows from $\bar{Y} \succeq 0$ and the assumption that A has full column rank (hence U_A has full column rank). Therefore, we obtain the following reduced problem:

$$\begin{array}{ll} \text{find} & Z \in \mathcal{S}_+^{n-m+r} \\ \text{s.t.} & H \circ \mathcal{K}\left(U_A Z U_A^T\right) = H \circ D \\ & Z_{22} = I, \end{array} \tag{27}$$

where Z is partitioned as

$$Z =: \begin{array}{c} \\ {\scriptstyle n-m} \\ {\scriptstyle r} \end{array} \!\!\begin{array}{c} {\scriptstyle n-m \quad r} \\ \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{12}^T & Z_{22} \end{bmatrix} \end{array}. \tag{28}$$

Since $Y - XX^T$ is the *Schur complement* of the matrix

$$\begin{bmatrix} Y & X \\ X^T & I \end{bmatrix}$$

with respect to the positive definite identity block, we have that

$$Y \succeq XX^T \quad \Leftrightarrow \quad \begin{bmatrix} Y & X \\ X^T & I \end{bmatrix} \succeq 0.$$

Therefore, we have a choice of a larger linear semidefinite constraint, or a smaller quadratic semidefinite constraint. See [38] for a theoretical discussion on the barriers associated with these two representations; see [46] for a numerical comparison.

The Biswas-Ye formulation

We now present the Biswas-Ye formulation [25, 26, 27, 28, 29] of the semidefinite relaxation of the sensor network localization problem. First we let $Y := XX^T$ be the Gram matrix of the rows of the matrix $X \in \mathbb{R}^{(n-m)\times r}$. Letting $e_{ij} \in \mathbb{R}^{n-m}$ be the vector with 1 in the i^{th} position, -1 in the j^{th} position, and zero elsewhere, we have

$$\begin{aligned}\|x_i - x_j\|^2 &= x_i^T x_i + x_j^T x_j - 2x_i^T x_j \\ &= e_{ij}^T X X^T e_{ij} \\ &= \left\langle e_{ij} e_{ij}^T, Y \right\rangle,\end{aligned}$$

for all $i, j = 1, \ldots, n-m$. Furthermore, letting $e_i \in \mathbb{R}^{n-m}$ be the vector with 1 in the i^{th} position, and zero elsewhere. Then

$$\begin{aligned}\|x_i - a_j\|^2 &= x_i^T x_i + a_j^T a_j - 2x_i^T a_j \\ &= \begin{bmatrix} e_i \\ a_j \end{bmatrix}^T \begin{bmatrix} X \\ I \end{bmatrix} \begin{bmatrix} X \\ I \end{bmatrix}^T \begin{bmatrix} e_i \\ a_j \end{bmatrix} \\ &= \left\langle \begin{bmatrix} e_i \\ a_j \end{bmatrix} \begin{bmatrix} e_i \\ a_j \end{bmatrix}^T, \begin{bmatrix} Y & X \\ X^T & I \end{bmatrix} \right\rangle,\end{aligned}$$

for all $i = 1, \ldots, n-m$, and $j = n-m+1, \ldots, n$, where we are now considering the anchors a_j to be indexed by $j \in \{n-m+1, \ldots, n\}$. Let $G = (N, E)$ be the graph corresponding to the partial Euclidean distance matrix D. Let

$$E_x := \{ij \in E : 1 \leq i < j \leq n-m\}$$

be the set of edges between sensors, and let

$$E_a := \{ij \in E : 1 \leq i \leq n-m, \ n-m+1 \leq j \leq n\}$$

be the set of edges between sensors and anchors. The sensor network localization problem can then be stated as:

$$\begin{array}{ll} \text{find } X \in \mathbb{R}^{(n-m)\times r} & \\ \text{s.t. } \left\langle \begin{bmatrix} e_{ij} \\ 0 \end{bmatrix} \begin{bmatrix} e_{ij} \\ 0 \end{bmatrix}^T, Z \right\rangle = D_{ij}, \quad \forall ij \in E_x & \\ \quad\ \ \left\langle \begin{bmatrix} e_i \\ a_j \end{bmatrix} \begin{bmatrix} e_i \\ a_j \end{bmatrix}^T, Z \right\rangle = D_{ij}, \quad \forall ij \in E_a & \\ \quad\ \ Z = \begin{bmatrix} Y & X \\ X^T & I \end{bmatrix} & \\ \quad\ \ Y = XX^T. & \end{array} \tag{29}$$

The Biswas-Ye semidefinite relaxation of the sensor network localization problem is then formed by relaxing the hard constraint $Y = XX^T$ to the convex constraint $Y \succeq XX^T$. As mentioned above, $Y \succeq XX^T$ is equivalent to

$$\begin{bmatrix} Y & X \\ X^T & I \end{bmatrix} \succeq 0. \tag{30}$$

Therefore, the Biswas-Ye relaxation is the *linear* semidefinite optimization problem

$$
\begin{array}{ll}
\text{find } Z \in \mathcal{S}_+^{n-m+r} & \\
\text{s.t. } \left\langle \begin{bmatrix} e_{ij} \\ 0 \end{bmatrix} \begin{bmatrix} e_{ij} \\ 0 \end{bmatrix}^T, \; Z \right\rangle = D_{ij}, \quad \forall ij \in E_x \\
\qquad \left\langle \begin{bmatrix} e_i \\ a_j \end{bmatrix} \begin{bmatrix} e_i \\ a_j \end{bmatrix}^T, \; Z \right\rangle = D_{ij}, \quad \forall ij \in E_a \\
\qquad Z_{22} = I,
\end{array}
\tag{31}
$$

where Z is partitioned as in equation (28). Clearly, we have that the Biswas-Ye formulation (31) is equivalent to the Euclidean distance matrix formulation (27). This is, in fact, identical to the relaxation in (25). The advantage of this point of view is that the approximations of the sensor positions are taken from the matrix X in (30) directly. However, it is not necessarily the case that these approximations are better than using the P obtained from a best rank-r approximation of Y in (25). (See also the section on page 26, below.) In fact, the empirical evidence in [46] indicate the opposite is true. Thus, this further emphasizes the fact that the anchors can be ignored.

Unique localizability

The sensor network localization problem (17)/(29) is called *uniquely localizable* if there is a unique solution $X \in \mathbb{R}^{(n-m)\times r}$ for problem (17)/(29) and if $\bar{X} \in \mathbb{R}^{(n-m)\times h}$ is a solution to the problem with anchors $\bar{A} := \begin{bmatrix} A & 0 \end{bmatrix} \in \mathbb{R}^{m\times h}$, then $\bar{X} = \begin{bmatrix} X & 0 \end{bmatrix}$. The following theorem from [109] shows that the semidefinite relaxation is tight if and only if the sensor network localization problem is uniquely localizable.

Theorem 5.4 ([109, Theorem 2]) *Let $A \in \mathbb{R}^{m\times r}$ such that $\begin{bmatrix} A & e \end{bmatrix}$ has full column rank. Let D be an n-by-n partial Euclidean distance matrix satisfying $D_{22} = \mathcal{K}(AA^T)$, with corresponding graph G and* 0–1 *adjacency matrix H. If G is connected, the following are equivalent.*

1. *The sensor network localization problem* (17)/(29) *is uniquely localizable.*
2. *The max-rank solution of the relaxation* (27)/(31) *has rank r.*
3. *The solution matrix Z of the relaxation* (27)/(31) *satisfies $Y = XX^T$, where*
$$Z = \begin{bmatrix} Y & X \\ X^T & I \end{bmatrix}. \; \square$$

Therefore, Theorem 5.4 implies that we can solve *uniquely localizable* instances of the sensor network localization problem in polynomial time by solving the semidefinite relaxation (27)/(31). However, it is important to point out an instance of the sensor network localization problem (17)/(29) which has a unique solution in $\mathbb{R}^r$ need not be uniquely localizable. This is especially important to point out in light of the complexity result in [17] and [16] in which it is proved that there is no efficient algorithm to solve instances of the sensor network localization problem having a unique solution in $\mathbb{R}^r$, unless $RP = NP$. This means we have two types of sensor network localization problem instances that have a unique solution in $\mathbb{R}^r$: (i) uniquely localizable, having no non-congruent solution in a higher dimension; (ii) not uniquely localizable, having a non-congruent solution in a higher dimension. Type (i) instances can be solved in polynomial time. Type (ii) cannot be solved in polynomial time, unless RP = NP.

Obtaining sensor positions from the semidefinite relaxation

Often it can be difficult to obtain a low rank solution from the semidefinite relaxation of a combinatorial optimization problem. An example of a successful semidefinite rounding technique is the impressive result in [60] for the Max-Cut problem; however, this is not always possible. For the sensor network localization problem we must obtain sensor positions from a solution of the semidefinite relaxation.

In the case of the Biswas-Ye formulation (31), or equivalently the Euclidean distance matrix formulation (27), the sensor positions $X \in \mathbb{R}^{(n-m)\times r}$ are obtained from a solution $Z \in \mathcal{S}_+^{n-m+r}$ by letting $X := Z_{12}$, where Z is partitioned as in equation (28). By Proposition 5.3, under the constraints that $\bar{Y} \succeq 0$ and $\bar{Y}_{22} = AA^T$, we may also compute the sensor positions as $X := \bar{Y}_{12}A^{\dagger T}$. Clearly, these are equivalent methods for computing the sensor positions.

Just after discussing the Procrustes problem (19), another method for computing the sensor positions was discussed for the uniquely localizable case when $\operatorname{rank}(\bar{Y}) = r$. Now suppose that $\operatorname{rank}(\bar{Y}) > r$. In this case we find a best rank-r approximation of $\bar{Y}$. For this, we turn to the classical result of Eckart and Young [51].

Theorem 5.5 ([30, Theorem 1.2.3]) *Let $A \in \mathbb{R}^{m\times n}$ and $k := \operatorname{rank}(A)$. Let*

$$A = U\Sigma V^T = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

be the singular value decomposition of A. Then the unique optimal solution of

$$\min\left\{\|A - X\|_F : \operatorname{rank}(X) = r\right\}$$

is given by

$$X := \sum_{i=1}^{r} \sigma_i u_i v_i^T,$$

with $\|A - X\|_F^2 = \sum_{i=r+1}^{k} \sigma_i^2$. □

Note that $\bar{Y} \succeq 0$, so the eigenvalue decomposition $\bar{Y} = UDU^T$ is also the singular value decomposition of $\bar{Y}$, and is less expensive to compute.

Comparing two methods

Let $A \in \mathbb{R}^{m\times r}$ and D be an n-by-n partial Euclidean distance matrix with

$$D = \begin{bmatrix} D_{11} & D_{12} \\ D_{12}^T & \mathcal{K}(AA^T) \end{bmatrix}.$$

Let H be the 0–1 adjacency matrix corresponding to D. Suppose D has a completion $\bar{D} \in \mathcal{E}^n$ having $\operatorname{embdim}(\bar{D}) = r$. Suppose Z is a feasible solution of the semidefinite relaxation (31) (or equivalently, the Euclidean distance matrix formulation (27)).

Using a path-following interior-point method to find Z can result in a solution with high rank. Indeed, [61] show that for semidefinite optimization problems having strict complementarity, the central path converges to the analytic centre of the optimal solution set (that is, the optimal solution with maximum determinant).

Let $\bar{Y} := U_A Z U_A^T$, where U_A is as defined in equation (26). Suppose that $k := \operatorname{rank}(\bar{Y}) > r$. Then $\mathcal{K}(\bar{Y})$ is a Euclidean distance matrix completion of the partial Euclidean distance matrix D, with $\operatorname{embdim}\mathcal{K}(\bar{Y}) = k$. Moreover, $\bar{Y} \in \mathcal{S}_+^n$ and $\bar{Y}_{22} = AA^T$, so by Proposition 5.3, we have that

$$\bar{Y} = \begin{bmatrix} Y & XA^T \\ AX^T & AA^T \end{bmatrix},$$

where $Y := Z_{11} = \bar{Y}_{11}$ and $X := Z_{12} = \bar{Y}_{12} A^{\dagger T}$. Let $\bar{Y}_r \in \mathcal{S}_+^n$ be the nearest rank-r matrix to $\bar{Y} \in \mathcal{S}_+^n$, in the sense of Theorem 5.5. Let $\bar{P} \in \mathbb{R}^{n\times k}$ and $\bar{P}_r \in \mathbb{R}^{n\times r}$ such that

$$\bar{Y} = \bar{P}\bar{P}^T \qquad \text{and} \qquad \bar{Y}_r = \bar{P}_r \bar{P}_r^T.$$

Let $\bar{P}$ and $\bar{P}_r$ be partitioned as

$$\bar{P} =: \begin{matrix} \\ {\scriptstyle n-m} \\ {\scriptstyle m} \end{matrix} \begin{matrix} {\scriptstyle r} \quad {\scriptstyle k-r} \\ \begin{bmatrix} \bar{P}_{11} & \bar{P}_{12} \\ \bar{P}_{12}^T & \bar{P}_{22} \end{bmatrix} \end{matrix} \qquad \text{and} \qquad \bar{P}_r =: \begin{matrix} \\ {\scriptstyle n-m} \\ {\scriptstyle m} \end{matrix} \begin{matrix} {\scriptstyle r} \\ \begin{bmatrix} \bar{P}_r' \\ \bar{P}_r'' \end{bmatrix} \end{matrix}.$$

For the first method, we simply use X for the sensor positions. Since $\begin{bmatrix} \bar{P}_{12}^T & \bar{P}_{22} \end{bmatrix} \begin{bmatrix} \bar{P}_{12}^T & \bar{P}_{22} \end{bmatrix}^T = \bar{Y}_{22} = AA^T = \begin{bmatrix} A & 0 \end{bmatrix} \begin{bmatrix} A & 0 \end{bmatrix}^T$, there exists an orthogonal matrix $\bar{Q} \in \mathbb{R}^{k\times k}$ such that $\begin{bmatrix} \bar{P}_{12}^T & \bar{P}_{22} \end{bmatrix} \bar{Q} = \begin{bmatrix} A & 0 \end{bmatrix}$. Let $\hat{P} := \bar{P}\bar{Q}$ and define the partition

$$\hat{P} =: \begin{matrix} \\ {\scriptstyle n-m} \\ {\scriptstyle m} \end{matrix} \begin{matrix} {\scriptstyle r} \quad {\scriptstyle k-r} \\ \begin{bmatrix} \hat{P}_{11} & \hat{P}_{12} \\ A & 0 \end{bmatrix} \end{matrix}.$$

Therefore, we see that the semidefinite relaxation of the sensor network localization problem has allowed the sensors to move into the higher dimension of $\mathbb{R}^k$ instead of fixing the sensors to the space $\mathbb{R}^r$. Now we have that

$$\bar{Y} = \hat{P}\hat{P}^T = \begin{bmatrix} \hat{P}_{11}\hat{P}_{11}^T + \hat{P}_{12}\hat{P}_{12}^T & \hat{P}_{11}A^T \\ A\hat{P}_{11}^T & AA^T \end{bmatrix},$$

implying that $\hat{P}_{11} = X$, and $Y = XX^T + \hat{P}_{12}\hat{P}_{12}^T$, so $Y - XX^T = \hat{P}_{12}\hat{P}_{12}^T \succeq 0$. Therefore, we have that

$$\left\| \bar{Y} - \begin{bmatrix} X \\ A \end{bmatrix} \begin{bmatrix} X \\ A \end{bmatrix}^T \right\|_F = \left\| \hat{P}_{12}\hat{P}_{12}^T \right\|_F = \left\| Y - XX^T \right\|_F.$$

For the second method, we use $\bar{X} := \bar{P}_r'' Q_r$ for the sensor positions, where $Q_r \in \mathbb{R}^{r\times r}$ is an orthogonal matrix that minimizes $\left\| \bar{P}_r'' Q_r - A \right\|_F$. Furthermore, we let $\bar{A} := \bar{P}_r'' Q_r$. Therefore, we have the following relationship between the two different approaches for computing sensor positions:

$$\left\| \bar{Y} - \begin{bmatrix} \bar{X} \\ \bar{A} \end{bmatrix} \begin{bmatrix} \bar{X} \\ \bar{A} \end{bmatrix}^T \right\|_F = \|\bar{Y} - \bar{P}_r\bar{P}_r^T\|_F \leq \left\| \bar{Y} - \begin{bmatrix} X \\ A \end{bmatrix} \begin{bmatrix} X \\ A \end{bmatrix}^T \right\|_F.$$

Moreover, computational experiments given at the end of this section show that we typically have

$$\left\| \bar{Y} - \begin{bmatrix} \bar{X} \\ A \end{bmatrix} \begin{bmatrix} \bar{X} \\ A \end{bmatrix}^T \right\|_F \approx \left\| \bar{Y} - \begin{bmatrix} \bar{X} \\ \bar{A} \end{bmatrix} \begin{bmatrix} \bar{X} \\ \bar{A} \end{bmatrix}^T \right\|_F.$$

Note that we have no guarantee that X from the first method, or $\bar{X}$ from the second method, satisfy the distance constraints. Moreover, we have the following bounds on the approximation of the Euclidean distance matrix $\mathcal{K}\left(\bar{Y}\right)$:

$$\begin{aligned} \left\| \mathcal{K}\left(\bar{Y}\right) - \mathcal{K}\left(\begin{bmatrix} X \\ A \end{bmatrix} \begin{bmatrix} X \\ A \end{bmatrix}^T \right) \right\|_F &\leq \|\mathcal{K}\|_F \left\| \bar{Y} - \begin{bmatrix} X \\ A \end{bmatrix} \begin{bmatrix} X \\ A \end{bmatrix}^T \right\|_F \\ &= 2\sqrt{n} \left\| Y - XX^T \right\|_F; \qquad (32) \\ \left\| \mathcal{K}\left(\bar{Y}\right) - \mathcal{K}\left(\begin{bmatrix} \bar{X} \\ \bar{A} \end{bmatrix} \begin{bmatrix} \bar{X} \\ \bar{A} \end{bmatrix}^T \right) \right\|_F &\leq \|\mathcal{K}\|_F \left\| \bar{Y} - \begin{bmatrix} \bar{X} \\ \bar{A} \end{bmatrix} \begin{bmatrix} \bar{X} \\ \bar{A} \end{bmatrix}^T \right\|_F \\ &= 2\sqrt{n} \left(\sum_{i=r+1}^{k} \lambda_i^2(\bar{Y}) \right)^{1/2}. \qquad (33) \end{aligned}$$

Clearly, the upper bound (33) for the second method is lower than upper bound (32) for the first method, but this need not imply that the second method gives a better r-dimensional approximation of the k-dimensional Euclidean distance matrix $\mathcal{K}(\bar{Y})$. Indeed, numerical tests were run in [46] to compare these two methods and often found better results using $\bar{X}$ than when using X, but this was not always the case.

In Figure 1 we have given the results of a simple numerical test conducted to investigate the differences between Method 1 ($P = [X; A]$), Method 2 ($P = [\bar{X}; \bar{A}]$), and Method 3 ($P = [\bar{X}; A]$). In all three cases, we compared the error in the approximation of the Gram matrix $\bar{Y}$ ($\|\bar{Y} - PP^T\|_F$), and the error in the approximation of the Euclidean distance matrix $\mathcal{K}(\bar{Y})$ ($\|\mathcal{K}(\bar{Y}) - \mathcal{K}(PP^T)\|_F$). In this test, we use 90 sensors and 10 anchors in dimension $r = 2$. We had 10 of the sensors inaccurately placed in dimension $k = 20$, to various degrees. We see that Method 2 and Method 3 are almost identical and may improve the approximation error when some sensors are inaccurately placed.

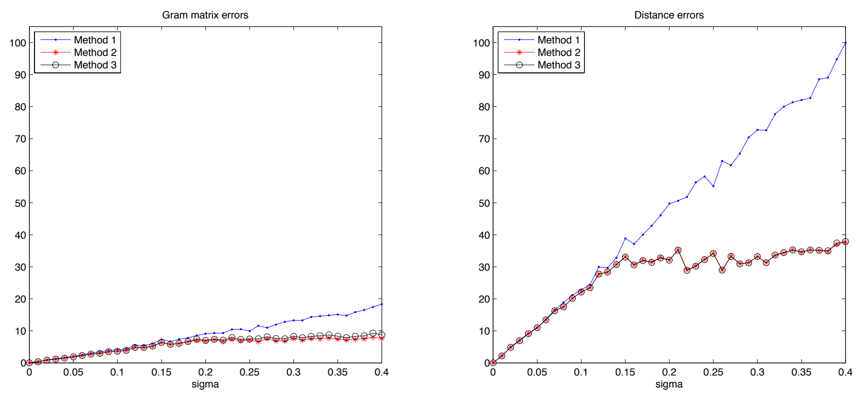

Fig. 1. Method 1 ($P = [X; A]$), Method 2 ($P = [\bar{X}; \bar{A}]$), and Method 3 ($P = [\bar{X}; A]$). The Gram matrix error is $\|\bar{Y} - PP^T\|_F$ and the distance error is $\|\mathcal{K}(\bar{Y}) - \mathcal{K}(PP^T)\|_F$, both normalized so that the maximum error is 100. We use $n = 100$ and $m = 10$ (90 sensors and 10 anchors) in dimension $r = 2$; ten of the sensors are inaccurately placed in dimension $k = 20$, to various degrees based on $Y - XX^T = \sigma \hat{P}_{12} \hat{P}_{12}^T$ for a randomly generated matrix $\hat{P}_{12} \in \mathbb{R}^{90 \times 18}$ with exactly 10 nonzero rows.

References

1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, *Wireless sensor networks: a survey*, Computer networks, 38 (2002), pp. 393–422.
2. S. Al-Homidan and H. Wolkowicz, *Approximate and exact completion problems for Euclidean distance matrices using semidefinite programming*, Linear Algebra Appl., 406 (2005), pp. 109–141.
3. A. Alfakih, M. Anjos, V. Piccialli, and H. Wolkowicz, *Euclidean distance matrices, semidefinite programming, and sensor network localization*, Portug. Math., (to appear).
4. A. Alfakih, A. Khandani, and H. Wolkowicz, *Solving Euclidean distance matrix completion problems via semidefinite programming*, Comput. Optim. Appl., 12 (1999), pp. 13–30. A tribute to Olvi Mangasarian.

5. A. Alfakih and H. Wolkowicz, *Matrix completion problems*, in Handbook of semidefinite programming, vol. 27 of Internat. Ser. Oper. Res. Management Sci., Kluwer Acad. Publ., Boston, MA, 2000, pp. 533–545.
6. A. Y. Alfakih, *Graph rigidity via Euclidean distance matrices*, Linear Algebra Appl., 310 (2000), pp. 149–165.
7. ———, *On rigidity and realizability of weighted graphs*, Linear Algebra Appl., 325 (2001), pp. 57–70.
8. ———, *On the uniqueness of Euclidean distance matrix completions*, Linear Algebra Appl., 370 (2003), pp. 1–14.
9. ———, *On the uniqueness of Euclidean distance matrix completions: the case of points in general position*, Linear Algebra Appl., 397 (2005), pp. 265–277.
10. ———, *On the nullspace, the rangespace and the characteristic polynomial of Euclidean distance matrices*, Linear Algebra Appl., 416 (2006), pp. 348–354.
11. ———, *A remark on the faces of the cone of Euclidean distance matrices*, Linear Algebra Appl., 414 (2006), pp. 266–270.
12. ———, *On dimensional rigidity of bar-and-joint frameworks*, Discrete Appl. Math., 155 (2007), pp. 1244–1253.
13. A. Y. Alfakih and H. Wolkowicz, *Two theorems on Euclidean distance matrices and Gale transform*, Linear Algebra Appl., 340 (2002), pp. 149–154.
14. B. P. W. Ames and S. A. Vavasis, *Nuclear norm minimization for the planted clique and biclique problems*, tech. rep., University of Waterloo, 2009.
15. L. T. H. An and P. D. Tao, *Large-scale molecular optimization from distance matrices by a D.C. optimization approach*, SIAM Journal on Optimization, 14 (2003), pp. 77–114.
16. J. Aspnes, T. Eren, D. K. Goldenberg, A. Morse, W. Whiteley, Y. R. Yang, B. D. O. Anderson, and P. N. Belhumeur, *A theory of network localization*, IEEE Transactions on Mobile Computing, 5 (2006), pp. 1663–1678.
17. J. Aspnes, D. Goldenberg, and Y. R. Yang, *On the computational complexity of sensor network localization*, Lecture Notes in Computer Science, 3121 (2004), pp. 32–44.
18. M. Bădoiu, E. D. Demaine, M. Hajiaghayi, and P. Indyk, *Low-dimensional embedding with extra information*, Discrete Comput. Geom., 36 (2006), pp. 609–632.
19. M. Bakonyi and C. Johnson, *The Euclidean distance matrix completion problem*, SIAM Journal on Matrix Analysis and Applications, 16 (1995), pp. 646–654.
20. A. Barvinok, *Problems of distance geometry and convex properties of quadratic maps*, Discrete and Computational Geometry, 13 (1995), pp. 189–202.
21. A. Beck, P. Stoica, and J. Li, *Exact and approximate solutions of source localization problems*, Signal Processing, IEEE Transactions on, 56 (2008), pp. 1770–1778.
22. A. Beck, M. Teboulle, and Z. Chikishev, *Iterative minimization schemes for solving the single source localization problem*, SIAM Journal on Optimization, 19 (2008), pp. 1397–1416.
23. M. Belk, *Realizability of graphs in three dimensions*, Discrete Comput. Geom., 37 (2007), pp. 139–162.
24. M. Belk and R. Connelly, *Realizability of graphs*, Discrete Comput. Geom., 37 (2007), pp. 125–137.

25. P. Biswas, *Semidefinite programming approaches to distance geometry problems*, PhD thesis, Stanford University, 2007.
26. P. Biswas, T.-C. Liang, K.-C. Toh, , Y. Ye, and T.-C. Wang, *Semidefinite programming approaches for sensor network localization with noisy distance measurements*, IEEE Transactions on Automation Science and Engineering, 3 (2006), pp. 360–371.
27. P. Biswas, K.-C. Toh, and Y. Ye, *A distributed SDP approach for large-scale noisy anchor-free graph realization with applications to molecular conformation*, SIAM Journal on Scientific Computing, 30 (2008), pp. 1251–1277.
28. P. Biswas and Y. Ye, *Semidefinite programming for ad hoc wireless sensor network localization*, in Information Processing In Sensor Networks, Proceedings of the third international symposium on Information processing in sensor networks, Berkeley, Calif., 2004, pp. 46–54.
29. ———, *A distributed method for solving semidefinite programs arising from ad hoc wireless sensor network localization*, in Multiscale Optimization Methods and Applications, vol. 82 of Nonconvex Optim. Appl., Springer, 2006, pp. 69–84.
30. Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
31. L. M. Blumenthal, *Theory and applications of distance geometry*, Chelsea Pub. Co, 1970.
32. J. Bruck, J. Gao, and A. Jiang, *Localization and routing in sensor networks by local angle information*, ACM Trans. Sen. Netw., 5 (2009), pp. 1–31.
33. N. Bulusu, J. Heidemann, and D. Estrin, *GPS-less low-cost outdoor localization for very small devices*, Personal Communications, IEEE, 7 (2000), pp. 28–34.
34. E. J. Candès and Y. Plan, *Matrix completion with noise*. Submitted to Proceedings of the IEEE, 2009.
35. E. J. Candès and B. Recht, *Exact matrix completion via convex optimization*, tech. rep., Caltech, 2008.
36. M. W. Carter, H. H. Jin, M. A. Saunders, and Y. Ye, *SpaseLoc: An adaptive subproblem algorithm for scalable wireless sensor network localization*, SIAM Journal on Optimization, 17 (2006), pp. 1102–1128.
37. A. Cassioli, *Solving the sensor network localization problem using an heuristic multistage approach*, tech. rep., Università degli Studi di Firenze, 2009.
38. C. Chua and L. Tunçel, *Invariance and efficiency of convex representations*, Mathematical Programming, 111 (2008), pp. 113–140.
39. R. Connelly, *Generic global rigidity*, Discrete Comput. Geom., 33 (2005), pp. 549–563.
40. J. A. Costa, N. Patwari, and A. O. Hero, III, *Distributed weighted-multidimensional scaling for node localization in sensor networks*, ACM Trans. Sen. Netw., 2 (2006), pp. 39–64.
41. G. M. Crippen, *Chemical distance geometry: Current realization and future projection*, Journal of Mathematical Chemistry, 6 (1991), pp. 307–324.
42. G. M. Crippen and T. F. Havel, *Distance Geometry and Molecular Conformation*, vol. 15 of Chemometrics Series, Research Studies Press Ltd., Chichester, 1988.
43. F. Critchley, *On certain linear mappings between inner-product and squared distance matrices*, Linear Algebra Appl., 105 (1988), pp. 91–107.

44. J. Dattorro, *Convex Optimization & Euclidean Distance Geometry*, Meboo Publishing USA, 2008.
45. Y. Ding, N. Krislock, J. Qian, and H. Wolkowicz, *Sensor network localization, Euclidean distance matrix completions, and graph realization*, in Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-Less Environment, San Francisco, 2008, pp. 129–134.
46. ———, *Sensor network localization, Euclidean distance matrix completions, and graph realization*, Optim. Eng., 11 (2010), pp. 45–66.
47. L. Doherty, K. S. J. Pister, and L. El Ghaoui, *Convex position estimation in wireless sensor networks*, in INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, vol. 3, 2001, pp. 1655–1663 vol.3.
48. Q. Dong and Z. Wu, *A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances*, Journal of Global Optimization, 22 (2002), pp. 365–375.
49. ———, *A geometric build-up algorithm for solving the molecular distance geometry problem with sparse distance data*, Journal of Global Optimization, 26 (2003), pp. 321–333.
50. R. dos Santos Carvalho, C. Lavor, and F. Protti, *Extending the geometric build-up algorithm for the molecular distance geometry problem*, Information Processing Letters, 108 (2008), pp. 234 – 237.
51. C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.
52. I. Z. Emiris and T. G. Nikitopoulos, *Molecular conformation search by distance matrix perturbations*, J. Math. Chem., 37 (2005), pp. 233–253.
53. T. Eren, O. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. O. Anderson, and P. N. Belhumeur, *Rigidity, computation, and randomization in network localization*, in INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 4, 2004, pp. 2673–2684.
54. R. W. Farebrother, *Three theorems with applications to Euclidean distance matrices*, Linear Algebra Appl., 95 (1987), pp. 11–16.
55. M. Fazel, *Matrix Rank Minimization with Applications*, PhD thesis, Stanford University, 2002.
56. M. Fazel, H. Hindi, and S. P. Boyd, *Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices*, in Proceedings of the American Control Conference, 2003, pp. 2156–2162.
57. M. Fukuda, M. Kojima, K. Murota, and K. Nakata, *Exploiting sparsity in semidefinite programming via matrix completion i: General framework*, SIAM Journal on Optimization, 11 (2001), pp. 647–674.
58. W. Glunt, T. Hayden, and M. Raydan, *Molecular conformations from distance matrices*, Journal of Computational Chemistry, 14 (1993), pp. 114–120.
59. W. Glunt, T. L. Hayden, S. Hong, and J. Wells, *An alternating projection algorithm for computing the nearest Euclidean distance matrix*, SIAM Journal on Matrix Analysis and Applications, 11 (1990), pp. 589–600.
60. M. X. Goemans and D. P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. ACM, 42 (1995), pp. 1115–1145.

61. D. Goldfarb and K. Scheinberg, *Interior point trajectories in semidefinite programming*, SIAM Journal on Optimization, 8 (1998), pp. 871–886.
62. G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 3rd ed., 1996.
63. J. C. Gower, *Euclidean distance geometry*, Math. Sci., 7 (1982), pp. 1–14.
64. ———, *Distance matrices and their Euclidean approximation*, in Data analysis and informatics, III (Versailles, 1983), North-Holland, Amsterdam, 1984, pp. 3–21.
65. ———, *Properties of Euclidean and non-Euclidean distance matrices*, Linear Algebra Appl., 67 (1985), pp. 81–97.
66. B. Green, *The orthogonal approximation of an oblique structure in factor analysis*, Psychometrika, 17 (1952), pp. 429–440.
67. R. Grone, C. R. Johnson, E. M. Sá, and H. Wolkowicz, *Positive definite completions of partial Hermitian matrices*, Linear Algebra and its Applications, 58 (1984), pp. 109–124.
68. I. G. Grooms, R. M. Lewis, and M. W. Trosset, *Molecular embedding via a second order dissimilarity parameterized approach*, SIAM Journal on Scientific Computing, 31 (2009), pp. 2733–2756.
69. T. Havel, I. Kuntz, and G. Crippen, *The theory and practice of distance geometry*, Bulletin of Mathematical Biology, 45 (1983), pp. 665–720.
70. T. F. Havel, *Metric matrix embedding in protein structure calculations, NMR spectra analysis, and relaxation theory*, Magnetic Resonance in Chemistry, 41 (2003), pp. S37–S50.
71. T. L. Hayden, J. Wells, W. M. Liu, and P. Tarazaga, *The cone of distance matrices*, Linear Algebra Appl., 144 (1991), pp. 153–169.
72. B. Hendrickson, *The Molecule Problem: Determining Conformation from Pairwise Distances*, PhD thesis, Cornell University, 1990.
73. ———, *Conditions for unique graph realizations*, SIAM Journal on Computing, 21 (1992), pp. 65–84.
74. ———, *The molecule problem: Exploiting structure in global optimization*, SIAM Journal on Optimization, 5 (1995), pp. 835–857.
75. N. J. Higham, *Computing the polar decomposition—with applications*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), pp. 1160–1174.
76. B. Jackson and T. Jordán, *Connected rigidity matroids and unique realizations of graphs*, J. Combin. Theory Ser. B, 94 (2005), pp. 1–29.
77. H. H. Jin, *Scalable Sensor Localization Algorithms for Wireless Sensor Networks*, PhD thesis, University of Toronto, Toronto, Ontario, Canada, 2005.
78. C. Johnson and P. Tarazaga, *Connections between the real positive semidefinite and distance matrix completion problems*, Linear Algebra Appl., 223/224 (1995), pp. 375–391. Special issue honoring Miroslav Fiedler and Vlastimil Pták.
79. R. M. Karp, *Reducibility among combinatorial problems*, in Complexity of computer computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972), Plenum, New York, 1972, pp. 85–103.
80. S. Kim, M. Kojima, and H. Waki, *Exploiting sparsity in SDP relaxation for sensor network localization*, SIAM J. Optim., 20 (2009), pp. 192–215.
81. S. Kim, M. Kojima, H. Waki, and M. Yamashita, *A sparse version of full semidefinite programming relaxation for sensor network localization problems*, Tech. Rep. B-457, Department of Mathematical and Computing Sciences Tokyo Institute of Technology, Oh-Okayama, Meguro, Tokyo 152-8552, 2009.

82. N. Krislock, *Semidefinite Facial Reduction for Low-Rank Euclidean Distance Matrix Completion*, PhD thesis, University of Waterloo, 2010.
83. N. Krislock, *Semidefinite Facial Reduction for Low-Rank Euclidean Distance Matrix Completion*, PhD thesis, University of Waterloo, 2010.
84. N. Krislock and H. Wolkowicz, *Explicit sensor network localization using semidefinite representations and facial reductions*, SIAM Journal on Optimization, 20 (2010), pp. 2679–2708.
85. P. S. Kumar and C. V. Madhavan, *Minimal vertex separators of chordal graphs*, Discrete Appl. Math., 89 (1998), pp. 155–168.
86. M. Laurent, *A connection between positive semidefinite and Euclidean distance matrix completion problems*, Linear Algebra Appl., 273 (1998), pp. 9–22.
87. 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), Amer. Math. Soc., Providence, RI, 1998, pp. 51–76.
88. M. Laurent, *Polynomial instances of the positive semidefinite and Euclidean distance matrix completion problems*, SIAM Journal on Matrix Analysis and Applications, 22 (2001), pp. 874–894.
89. N.-H. Z. Leung and K.-C. Toh, *An SDP-based divide-and-conquer algorithm for large scale noisy anchor-free graph realization*, tech. rep., Department of Mathematics, National University of Singapore, 2008.
90. X.-Y. Li, *Wireless Ad Hoc and Sensor Networks: Theory and Applications*, Cambridge University Press, 2008.
91. S. Megerian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, *Worst and best-case coverage in sensor networks*, Mobile Computing, IEEE Transactions on, 4 (2005), pp. 84–92.
92. D. Moore, J. Leonard, D. Rus, and S. Teller, *Robust distributed network localization with noisy range measurements*, in SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems, New York, NY, USA, 2004, ACM, pp. 50–61.
93. J. J. Moré and Z. Wu, *Global continuation for distance geometry problems*, SIAM Journal on Optimization, 7 (1997), pp. 814–836.
94. J. J. Moré and Z. Wu, *Distance geometry optimization for protein structures*, Journal of Global Optimization, 15 (1999), pp. 219–234.
95. S. Nawaz, *Anchor Free Localization for Ad-hoc Wireless Sensor Networks*, PhD thesis, University of New South Wales, 2008.
96. J. Nie, *Sum of squares method for sensor network localization*, Computational Optimization and Applications, 43 (2009), pp. 151–179.
97. T. Pong and P. Tseng, *(Robust) Edge-based semidefinite programming relaxation of sensor network localization*, Mathematical Programming, (2010).
98. M. V. Ramana, L. Tunçel, and H. Wolkowicz, *Strong duality for semidefinite programming*, SIAM Journal on Optimization, 7 (1997), pp. 641–662.
99. B. Recht, M. Fazel, and P. A. Parrilo, *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, SIAM Review, 52 (2010), pp. 471–501.
100. B. Recht, W. Xu, and B. Hassibi, *Necessary and sufficient conditions for success of the nuclear norm heuristic for rank minimization*, 2008.
101. R. T. Rockafellar, *Convex Analysis*, Princeton University Press, 1970.
102. 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, New York, NY, USA, 2001, ACM, pp. 166–179.

103. J. B. Saxe, *Embeddability of weighted graphs in k-space is strongly NP-hard*, in Proceedings of the 17th Allerton Conference on Communications, Control, and Computing, 1979, pp. 480–489.
104. R. Schneider, *Convex bodies: the Brunn-Minkowski theory*, Cambridge University Press, 1993.
105. I. Schoenberg, *Remarks to Maurice Fréchet's article "Sur la définition axiomatique d'une classe d'espace distanciés vectoriellement applicable sur l'espace de Hilbert"*, Ann. Math., 36 (1935), pp. 724–732.
106. P. Schönemann, *A generalized solution of the orthogonal Procrustes problem*, Psychometrika, 31 (1966), pp. 1–10.
107. A. M.-C. So, *A Semidefinite Programming Approach to the Graph Realization Problem: Theory, Applications and Extensions*, PhD thesis, Computer Science Department, Stanford University, 2007.
108. A. M.-C. So and Y. Ye, *A semidefinite programming approach to tensegrity theory and realizability of graphs*, in SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm, New York, NY, USA, 2006, ACM, pp. 766–775.
109. ———, *Theory of semidefinite programming for sensor network localization*, Math. Program., 109 (2007), pp. 367–384.
110. T. Stoyanova, F. Kerasiotis, A. Prayati, and G. Papadopoulos, *Evaluation of impact factors on RSS accuracy for localization and tracking applications in sensor networks*, Telecommunication Systems, 42 (2009), pp. 235–248.
111. P. Tarazaga, *Faces of the cone of Euclidean distance matrices: characterizations, structure and induced geometry*, Linear Algebra Appl., 408 (2005), pp. 1–13.
112. P. Tarazaga, T. L. Hayden, and J. Wells, *Circum-Euclidean distance matrices and faces*, Linear Algebra Appl., 232 (1996), pp. 77–96.
113. P. Tseng, *Second-order cone programming relaxation of sensor network localization*, SIAM J. Optim., 18 (2007), pp. 156–185.
114. Z. Wang, S. Zheng, S. Boyd, and Y. Ye, *Further relaxations of the semidefinite programming approach to sensor network localization*, SIAM J. Optim., 19 (2008), pp. 655–673.
115. 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, New York, NY, USA, 2004, ACM, p. 106.
116. D. Wu and Z. Wu, *An updated geometric build-up algorithm for solving the molecular distance geometry problems with sparse distance data*, Journal of Global Optimization, 37 (2007), pp. 661–673.
117. D. Wu, Z. Wu, and Y. Yuan, *Rigid versus unique determination of protein structures with geometric buildup*, Optimization Letters, 2 (2008), pp. 319–331.
118. Z. Yang, Y. Liu, and X.-Y. Li, *Beyond trilateration: On the localizability of wireless ad-hoc networks*, in INFOCOM 2009, IEEE, 2009, pp. 2392–2400.
119. Y. Yemini, *Some theoretical aspects of position-location problems*, in 20th Annual Symposium on Foundations of Computer Science (San Juan, Puerto Rico, 1979), IEEE, New York, 1979, pp. 1–8.
120. G. Young and A. Householder, *Discussion of a set of points in terms of their mutual distances*, Psychometrika, 3 (1938), pp. 19–22.

Index