NOISY EUCLIDEAN DISTANCE REALIZATION: ROBUST FACIAL REDUCTION AND THE PARETO FRONTIER

D. DRUSVYATSKIY*, N. KRISLOCK†, Y.-L. VORONIN‡, AND H. WOLKOWICZ§

Abstract. We present two algorithms for large-scale low-rank Euclidean distance matrix completion problems, based on semidefinite optimization. Our first method works by relating cliques in the graph of the known distances to faces of the positive semidefinite cone, yielding a combinatorial procedure that is provably robust and parallelizable. Our second algorithm is a first order method for maximizing the trace—a popular low-rank inducing regularizer—in the formulation of the problem with a constrained misfit. Both of the methods output a point configuration that can serve as a high-quality initialization for local optimization techniques. Numerical experiments on large-scale sensor localization problems illustrate the two approaches.

Key words. Euclidean distance matrices, sensor network localization, convex optimization, facial reduction, Frank-Wolfe algorithm, semidefinite programming

AMS subject classifications. 90C22, 90C25 , 52A99

1 Introduction. A pervasive task in distance geometry is the inverse problem: given only local pairwise Euclidean distances among a set of points, recover their locations in space. More precisely, given a weighted undirected graph $G = (V, E, d)$ on a vertex set $\{1, \dots, n\}$ and an integer r, find (if possible) a set of points $x_1, \dots, x_n$ in $\mathbb{R}^r$ satisfying

$$\|x_i - x_j\|^2 = d_{ij}, \quad \text{for all edges } ij \in E,$$

where $\|\cdot\|$ denotes the usual Euclidean norm on $\mathbb{R}^r$. In most applications, the given squared distances d_{ij} are inexact, and one then seeks points $x_1, \dots, x_n$ satisfying the distance constraints only approximately. This problem appears under numerous names in the literature, such as Euclidean Distance Matrix (EDM) completion and graph realization [2,12,26], and is broadly applicable for example in wireless networks, statistics, robotics, protein reconstruction, and dimensionality reduction in data analysis; the recent survey [27] has an extensive list of relevant references. Fixing notation, we will refer to this problem as *EDM completion*, throughout.

The EDM completion problem can be modeled as the nonconvex feasibility problem: find a symmetric $n \times n$ matrix X satisfying

$$\left\{ \begin{array}{ll} X_{ii} + X_{jj} - 2X_{ij} = d_{ij}, & \text{for all } ij \in E, \\ Xe = 0, & \\ \operatorname{rank} X \leq r, & \\ X \succeq 0, & \end{array} \right\} \tag{1.1}$$

where e stands for the vector of all ones. Indeed, if $X = PP^T$ is a maximal rank factorization of such a matrix X, then the rows of P yield a solution to the EDM

*Department of Mathematics, University of Washington, Seattle, WA 98195-4350, USA. Research was partially supported by the AFOSR YIP award FA9550-15-1-0237. math.washington.edu/~ddrusv

†Department of Mathematical Sciences, Northern Illinois University, DeKalb, IL 60115, USA. www.math.niu.edu/~krislock

‡Department of Computer Science, University of Colorado, Boulder, CO 80309-0430, USA. cs.colorado.edu/~yuvo9296

§Department of Combinatorics and Optimization, Waterloo, Ontario N2L 3G1, Canada. Research supported by Natural Sciences Engineering Research Council Canada and a grant from AFOSR. orion.math.uwaterloo.ca/~hwolkowi

completion problem. The constraint $Xe = 0$ simply ensures that the rows of P are centered around the origin. Naturally a convex relaxation is obtained by simply ignoring the rank constraint. The resulting problem is convex (a semidefinite program (SDP) in fact) and so more tractable. For many instances, particularly coming from dense wireless networks, this relaxation is exact, that is the solution of the convex rank-relaxed problem automatically has the desired rank r [32]. Consequently, semidefinite programming techniques have proven to be extremely useful for this problem; see for example [6–10, 23, 28, 32, 40]. For large networks, however, the SDPs involved can become intractable for off-the-shelf methods. Moreover, this difficulty is compounded by the inherent *ill-conditioning* in the SDP relaxation of (1.1)—a key theme of the paper. For example, it is easy to see that each clique in G on more than $r+2$ vertices certifies that the SDP is *not* strictly feasible, provided the true points of the clique were in general position in $\mathbb{R}^r$.

In the current work, we attempt to close the computational gap by proposing a combinatorial algorithm and an efficient first-order method for the EDM completion problem. The starting point is the observation that the cliques in G play a special role in the completion problem. Indeed, from each sufficiently large clique in the graph G, one can determine a face of the positive semidefinite cone containing the entire feasible region of (1.1). This observation immediately motivated the algorithm of [23]. The procedure proceeds by collecting a large number of cliques in the graph and intersecting the corresponding faces two at a time (while possibly growing cliques), each time causing a dimensional decrease in the problem. If the SDP relaxation is exact and the graph is sufficiently dense, the method often terminates with a unique solution without having to invoke an SDP solver. An important caveat of this geometric approach is that near-exactness of the distance measurements is essential for the algorithm to work, both in theory and in practice, for the simple reason that randomly perturbed faces of the positive semidefinite cone typically intersect only at the origin. Remarkably, using dual certificates, we are able to design a method complementary to [23] for the problem (1.1) that under reasonable conditions, is provably robust to noise in the distance measurements, in the sense that the output error is linearly proportional to the noise level. Moreover, in contrast to the algorithm [23], the new method is conceptually easy to parallelize. In the late stages of writing the current paper, we became aware of the related work [30]. There the author proposes a robust algorithm for the EDM completion problem that is in the same spirit as ours, but is stated in the language of rigidity theory. As a byproduct, our current work yields an interpretation of the algorithm [30] in terms of facial reduction iterations and SDP techniques. Moreover, we offer additional improvements via a nonrigid clique union subroutine (Subsection 3.3.1), which we found essential for the success of the algorithm.

In the second part of the paper, we propose a first order method for solving the noisy EDM completion problem. To this end, we consider maximizing the trace—a popular low-rank inducing regularizer [5, 41]—in the formulation of the problem:

$$\begin{aligned} &\text{maximize} && \operatorname{tr} X \\ &\text{subject to} && \sum_{ij\in E} |X_{ii} + X_{jj} - 2X_{ij} - d_{ij}|^2 \le \sigma \\ &&& Xe = 0 \\ &&& X \succeq 0. \end{aligned} \tag{1.2}$$

Here σ is an a priori chosen tolerance reflecting the total noise level. Notice, that this formulation directly contrasts the usual min-trace regularizer in compressed sensing;

nonetheless it is very natural. An easy computation shows that in terms of the factorization $X = PP^T$, the equality $\operatorname{tr}(X) = \frac{1}{2n}\sum_{i,j=1}^{n} \|p_i - p_j\|^2$ holds, where p_i are the rows of P. Thus trace maximization serves to "flatten" the realization of the graph. We note in passing that we advocate using (1.2) instead of perhaps the more usual regularized problem

$$\begin{aligned} \text{minimize} \quad & \sum_{ij\in E} |X_{ii} + X_{jj} - 2X_{ij} - d_{ij}|^2 - \lambda \operatorname{tr} X \\ \text{subject to} \quad & Xe = 0, \quad X \succeq 0. \end{aligned}$$

The reason is that choosing a reasonable value of the trade-off parameter λ can be difficult, whereas an estimate of σ is typically available from a priori known information on the noise level.

As was observed above, for $\sigma = 0$ the problem formulation (1.2) notoriously fails strict feasibility. In particular, for small $\sigma \geq 0$ the feasible region is very thin and the solution to the problem is unstable. As a result, iterative methods that maintain feasibility are likely to exhibit serious difficulties. Keeping this in mind, we propose an *infeasible* first-order method, which is not directly effected by the poor conditioning of the underlying problem.

To this end, consider the following parametric problem, obtained by "flipping" the objective and the quadratic constraint in (1.2):

$$\begin{aligned} v(\tau) := \quad \text{minimize} \quad & \sum_{ij\in E} |X_{ii} + X_{jj} - 2X_{ij} - d_{ij}|^2 \\ \text{subject to} \quad & \operatorname{tr} X = \tau \\ & Xe = 0 \\ & X \succeq 0. \end{aligned}$$

Notice that the problem of evaluating $v(\tau)$ is readily amenable to first order methods, in direct contrast to (1.2). Indeed, the feasible region is geometrically simple. In particular, linear optimization over the region only requires computing a maximal eigenvalue. Hence the evaluation of $v(\tau)$ is well adapted for the *Frank-Wolfe method*, a projection-free first order algorithm. Indeed, the gradient of the objective function is very sparse (as sparse as the edge set E) and therefore optimizing the induced linear functional over the feasible region then becomes a cheap operation. Now, solving (1.2) amounts to finding the largest value of τ satisfying $v(\tau) \leq \sigma$, a problem that can be solved by an approximate Newton method. Analogous root finding strategies can be found, for example, in [3, 37–39]. Using this algorithm, we investigate the apparent superiority of the max-trace regularizer over the min-trace regularizer with respect to both low-rank recovery and efficient computation.

The outline of the paper is as follows. Section 2 collects some preliminaries on the facial structure of the positive semidefinite cone and the SDP relaxation of the EDM completion problem. Section 3 presents the proposed robust facial reduction algorithm and provides some numerical illustrations. Section 4 describes the proposed Pareto search technique with Frank-Wolfe iterations, and presents numerical experiments.

2 Preliminaries.

In this section, we record some preliminaries and formally state the EDM completion problem.

2.1 Geometry of the positive semidefinite cone.

The main tool we use in the current work (even if indirectly) is semidefinite programming (SDP). To this end,

let $\mathcal{S}^n$ denote the Euclidean space of $n \times n$ real symmetric matrices endowed with the trace inner product $\langle A, B\rangle = \operatorname{tr} AB$ and the Frobenius norm $\|A\|_F = \sqrt{\operatorname{tr} A^2}$. The convex cone of $n \times n$ positive semidefinite (PSD) matrices will be denoted by $\mathcal{S}^n_+$. This cone defines a partial ordering: for any $A, B \in \mathcal{S}^n$ the binary relation $A \succeq B$ means $A - B \in \mathcal{S}^n_+$. A convex subset $\mathcal{F}$ of $\mathcal{S}^n_+$ is a *face* of $\mathcal{S}^n_+$ if $\mathcal{F}$ contains any line segment in $\mathcal{S}^n_+$ whose relative interior intersects $\mathcal{F}$, and a face $\mathcal{F}$ of $\mathcal{S}^n_+$ is *proper* if it is neither empty nor all of $\mathcal{S}^n_+$. All faces of $\mathcal{S}^n_+$ have the (primal) form

$$\mathcal{F} = \left\{ U \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} U^T : A \in \mathcal{S}^k_+ \right\}, \tag{2.1}$$

for some $n \times n$ orthogonal matrix U and some integer $k \in \{0, 1, \ldots, n\}$. Any face $\mathcal{F}$ of $\mathcal{S}^n_+$ can also be written in dual form as $Y^\perp \cap \mathcal{S}^n_+$ for some PSD matrix $Y \in \mathcal{S}^n_+$. Indeed, suppose that $\mathcal{F}$ has the representation (2.1). Then we may equivalently write $\mathcal{F} = Y^\perp \cap \mathcal{S}^n_+$, with $Y := U \begin{bmatrix} 0 & 0 \\ 0 & B \end{bmatrix} U^T$ for any nonsingular matrix B in $\mathcal{S}^{n-k}_+$. In general, if a face has the form $\mathcal{F} = Y^\perp \cap \mathcal{S}^n_+$ for some PSD matrix Y, then we say that Y *exposes* $\mathcal{F}$. Finally, for any convex subset $\Omega \subset \mathcal{S}^n_+$, the symbol $\operatorname{face}(\Omega; \mathcal{S}^n_+)$ will denote the minimal face of $\mathcal{S}^n_+$ containing Ω. The cone $\operatorname{face}(\Omega; \mathcal{S}^n_+)$ then coincides with $\operatorname{face}(X; \mathcal{S}^n_+)$, where X is any maximal rank matrix in Ω.

2.2 EDM completion problem. Throughout, we fix an integer $r \geq 0$ and a weighted undirected graph $G = (V, E, d)$ on a node set $V = \{1, \ldots, n\}$, with an edge set $E \subseteq \{ij : 1 \leq i < j \leq n\}$ and a vector $d \in \mathbb{R}^E$ of nonnegative weights. The vertices represent points in an r-dimensional space $\mathbb{R}^r$, while the presence of an edge ij joining the vertices i and j signifies that the physical distance between the points i and j is available.

The *EDM completion problem* is to find a set of points in $x_1, \ldots, x_n \in \mathbb{R}^r$ satisfying

$$\|x_i - x_j\|^2 = d_{ij}, \qquad \text{for all } ij \in E.$$

Such a collection of points $x_1, \ldots, x_n$ is said to *realize* the graph G in $\mathbb{R}^r$. Notice that without loss of generality, such realizing points $x_1, \ldots, x_n$ can always be translated so that they are centered around the origin, meaning $\sum_i x_i = 0$.

The EDM completion problem is equivalent to finding a matrix $X \in \mathcal{S}^n$ satisfying the system:

$$\left\{ \begin{array}{ll} X_{ii} + X_{jj} - 2X_{ij} = d_{ij}, & \text{for all } ij \in E, \\ Xe = 0, & \\ \operatorname{rank} X \leq r, & \\ X \succeq 0. & \end{array} \right\} \tag{2.2}$$

Here $e \in \mathbb{R}^n$ denotes the vector of all ones. Indeed, suppose that X satisfies this system. Then since X is positive semidefinite and has rank at most r, we may form a factorization $X = PP^T$ for some $n \times r$ matrix P. It is easy to verify that the rows of P realize G in $\mathbb{R}^r$. Conversely, if some points $x_1, \ldots, x_n \in \mathbb{R}^r$ realize G in $\mathbb{R}^r$, then we may center them around the origin and assemble them into the matrix $P = [x_1; \ldots; x_n]^T \in \mathbb{R}^{n \times r}$. The resulting *Gram matrix* $X := PP^T$ is feasible for the above system. For more details, see for example [23].

The EDM completion problem is nonconvex and is NP-hard in general [29, 42]. A convex relaxation is obtained simply by ignoring the rank constraint yielding a convex

SDP feasibility problem:

$$\left\{ \begin{array}{ll} X_{ii} + X_{jj} - 2X_{ij} = d_{ij}, & \text{for all } ij \in E, \\ Xe = 0, & \\ X \succeq 0. & \end{array} \right\} \tag{2.3}$$

For many EDM completion problems on fairly dense graphs, this convex relaxation is "exact" [31]. For example the following is immediate.

OBSERVATION 2.1 (Exactness of the relaxation). *If the EDM completion problem* (2.2) *is feasible, then the following are equivalent:*

1. *No realization of G in $\mathbb{R}^l$, for $l > r$, spans the ambient space $\mathbb{R}^l$.*
2. *Any solution of the relaxation* (2.3) *has rank at most r and consequently any solution of* (2.3) *yields a realization of G in $\mathbb{R}^r$.*

In theory, the exactness of the relaxation is a great virtue. From a computational perspective, however, exactness implies that the SDP formulation (2.3) does not admit a positive definite solution, i.e., that strict feasibility fails. Moreover, it is interesting to note that a very minor addition to the assumptions of Observation 2.1 implies that the SDP (2.3) admits a unique solution [31]. We provide a quick proof for completeness, though the reader can safely skip it.

OBSERVATION 2.2 (Uniqueness of the solution). *If the EDM completion problem* (2.2) *is feasible, then the following are equivalent:*

1. *The graph G cannot be realized in $\mathbb{R}^{r-1}$, and moreover for any $l > r$ no realization in $\mathbb{R}^l$ spans the ambient space $\mathbb{R}^l$.*
2. *The relaxation* (2.3) *has a unique solution.*

Proof. The implication $2 \Rightarrow 1$ is immediate. To see the converse implication $1 \Rightarrow 2$, suppose that the SDP (2.3) admits two solutions X and Y. Define $\mathcal{F}$ now to be the minimal face of $\mathcal{S}^n_+$ containing the feasible region. Note that by Observation 2.1, any solution of the SDP has rank at most r, and hence every matrix in $\mathcal{F}$ has rank at most r. Consider now the line $L := \{X + \lambda(Y - X) : \lambda \in \mathbb{R}\}$. Clearly L is contained in the linear span of $\mathcal{F}$ and the line segment $L \cap \mathcal{F}$ is contained in the feasible region. Since $\mathcal{F}$ is pointed, the intersection $L \cap \mathcal{F}$ has at least one endpoint Z, necessarily lying in the relative boundary of $\mathcal{F}$. This matrix Z therefore has rank at most $r - 1$, a contradiction since Z yields a realization of G in $\mathbb{R}^{r-1}$. ∎

In principle, one may now apply any off-the-shelf SDP solver to solve problem (2.3). The effectiveness of such methods, however, depends heavily on the "conditioning" of the SDP system. In particular, if the system admits no feasible positive definite matrix, as is often the case (Observation 2.1), then no standard method can be guaranteed to perform very well nor be robust to perturbations in the distance measurements.

2.3 Constraint mapping and the centering issue. To simplify notation, we will reserve some symbols for the mappings and sets appearing in formulations (2.2) and (2.3). To this end, define the mapping $\mathcal{K} : \mathcal{S}^n \to \mathcal{S}^n$ by

$$\mathcal{K}(X)_{ij} := X_{ii} + X_{jj} - 2X_{ij}.$$

The adjoint $\mathcal{K}^* : \mathcal{S}^n \to \mathcal{S}^n$ is given by

$$\mathcal{K}^*(D) = 2(\mathrm{Diag}(De) - D).$$

Moreover, the *Moore-Penrose pseudoinverse* of $\mathcal{K}$ is easy to describe: for any matrix $D \in \mathcal{S}^n$ having all-zeros on the diagonal (for simplicity), we have

$$\mathcal{K}^\dagger(D) = -\frac{1}{2} J \cdot D \cdot J,$$

where $J := I - \frac{1}{n} ee^T$ is the projection onto $e^\perp$. These and other related constructions have appeared in a number of publications; see for example [1, 19, 20, 24, 25, 33–36].

Consider now the sets of *centered symmetric*, *centered PSD*, and *centered PSD low-rank* matrices

$$\begin{aligned} \mathcal{S}^n_c &:= \{X \in \mathcal{S}^n : Xe = 0\}, \\ \mathcal{S}^n_{c,+} &:= \{X \in \mathcal{S}^n_+ : Xe = 0\}, \\ \mathcal{S}^{n,r}_{c,+} &:= \{X \in \mathcal{S}^n_{c,+} : \operatorname{rank} X \le r\}. \end{aligned}$$

Define now the coordinate projection $\mathcal{P}\colon \mathcal{S}^n \to \mathbb{R}^E$ by setting $\mathcal{P}(X)_{ij} = X_{ij}$. In this notation, the feasible set (2.2) can equivalently be written as $\{X \in \mathcal{S}^{n,r}_{c,+} : \mathcal{P} \circ \mathcal{K}(X) = d\}$ while the relaxation (2.3) is then $\{X \in \mathcal{S}^n_{c,+} : \mathcal{P} \circ \mathcal{K}(X) = d\}$.

It is easy to see that $\mathcal{S}^n_{c,+}$ is a face of $\mathcal{S}^n_+$, and is linearly isomorphic to $\mathcal{S}^{n-1}_+$. Indeed, the matrix ee^T exposes $\mathcal{S}^n_{c,+}$. More specifically, for any $n \times n$ orthogonal matrix $\begin{bmatrix} \frac{1}{\sqrt{n}} e & U \end{bmatrix}$, we have the representation

$$\mathcal{S}^n_{c,+} = U \mathcal{S}^{n-1}_+ U. \tag{2.4}$$

Consequently, we now make the following important convention: the *ambient space* of $\mathcal{S}^n_{c,+}$ will always be taken as $\mathcal{S}^n_c$. The notion of faces of $\mathcal{S}^n_{c,+}$ and the corresponding notion of exposing matrices naturally adapts to this convention by appealing to (2.4) and the respective standard notions for $\mathcal{S}^{n-1}_+$. Namely, we will say that $\mathcal{F}$ is a face of $\mathcal{S}^n_{c,+}$ if it has the form $\mathcal{F} = U \widehat{\mathcal{F}} U^T$ for some face $\widehat{\mathcal{F}}$ of $\mathcal{S}^{n-1}_+$, and that a matrix Y exposes $\mathcal{F}$ whenever it has the form $U \widehat{Y} U^T$ for some matrix $\widehat{Y}$ exposing $\widehat{\mathcal{F}}$.

3 Robust facial reduction for EDM completions. In this section, we propose the use of robust facial reduction for solving the least-squares formulation of the nonconvex EDM completion problem (2.2):

$$\begin{array}{ll} \text{minimize} & \sum_{ij \in E} |X_{ii} + X_{jj} - 2X_{ij} - d_{ij}|^2 \\ \text{subject to} & X \in \mathcal{S}^{n,r}_{c,+}. \end{array} \tag{3.1}$$

The main idea is to use the dual certificates arising from the rigid structures of the graph to construct a positive semidefinite matrix Y of rank at least $n - r$, and then solve the convex optimization problem:

$$\begin{array}{ll} \text{minimize} & \sum_{ij \in E} |X_{ii} + X_{jj} - 2X_{ij} - d_{ij}|^2 \\ \text{subject to} & X \in \mathcal{S}^n_{c,+} \cap Y^\perp. \end{array}$$

Before describing our algorithmic framework for tackling (3.1), it is instructive to put it into context. The authors of [23] found a way to use the degeneracy of the system (2.2) explicitly to design a combinatorial algorithm for solving (2.2), under reasonable conditions. The authors observed that each k-clique in the graph G, with $k > r$, certifies that the entire feasible region of the convex relaxation (2.3) lies in

a certain proper face $\mathcal{F}$ of the positive semidefinite cone $\mathcal{S}^n_+$. Therefore, the *facial reduction* technique of replacing $\mathcal{S}^n_+$ by the smaller set $\mathcal{F}$ can be applied on (2.3) to obtain an equivalent problem involving fewer variables. On a basic level, their method explores cliques in the graph, while possibly growing them, and intersects pairwise such faces in a computationally effective way.

An important computational caveat of the facial reduction algorithm of [23] is that the algorithm is highly unstable when the distance measurements are corrupted by noise—a ubiquitous feature of the EDM completion problem in virtually all applications. The reason is simple: randomly perturbed faces of the semidefinite cone typically intersect only at the origin. Hence small perturbations in the distance measurements will generally lead to poor guesses of the face intersection arising from pairs of cliques. Moreover, even if pairs of cliques can robustly yield some facial information, the accumulated error compounds as the algorithm moves from clique to clique. Remarkably, we show that this difficulty can be overcome by using "dual" representations of faces to aggregate the noise. Indeed, the salient feature of the dual representation is that it is much better adapted at handling noise.

Before proceeding with the details of the proposed algorithmic framework, we provide some intuition. To this end, an easy computation shows that if Y_i exposes a face $\mathcal{F}_i$ of $\mathcal{S}^n_+$ (for $i = 1, \dots, m$), then the sum $\sum_i Y_i$ exposes the intersection $\bigcap_i \mathcal{F}_i$. Thus the faces $\mathcal{F}_i$ intersect trivially if and only if the sum $\sum_i Y_i$ is positive definite. On the other hand, if the true exposing vectors arising from the cliques are corrupted by noise, then one can round off the small eigenvalues of $\sum_i Y_i$ (due to noise) to guess at the true intersection of the faces arising from the noiseless data.

3.1 The algorithmic framework. To formalize the outlined algorithm, we will need the following basic result, which in a primal form was already the basis for the algorithm in [23]. The dual form, however, is essential for our purposes. For easy similar alternative proofs, see [11, Theorem 4.9] and [22, Theorem 4.1]. Henceforth, given a clique $\alpha \subseteq V$ (meaning, a subset of vertices such that every two are adjacent), we use $d_\alpha \in \mathcal{S}^{|\alpha|}$ to denote the symmetric matrix formed from restricting d to the edges between the vertices in α.

THEOREM 3.1 (One clique facial reduction). *Suppose that the subset of vertices $\alpha := \{1, \dots, k\} \subset V$ is a clique in G. Define the set*

$$\widehat{\Omega} \;:=\; \{X \in \mathcal{S}^n_{c,+} : [\mathcal{K}(X)]_{ij} = d_{ij} \quad \textit{for all} \quad 1 \le i < j \le k\}.$$

Then for any matrix $\widehat{Y}$ exposing $\operatorname{face}\big(\mathcal{K}^\dagger d_\alpha; \mathcal{S}^k_{c,+}\big)$,

$$\textit{the matrix} \quad \begin{bmatrix} \widehat{Y} & 0 \\ 0 & 0 \end{bmatrix} \quad \textit{exposes} \quad \operatorname{face}(\widehat{\Omega}; \mathcal{S}^n_{c,+}).$$

In particular, under the assumptions of the theorem, the entire feasible region of (2.3) is contained in the face of $\mathcal{S}^n_{c,+}$ exposed by $\begin{bmatrix} \widehat{Y} & 0 \\ 0 & 0 \end{bmatrix}$. The assumption that the first k vertices formed a clique is of course made without loss of generality. We can now state our proposed algorithmic framework, in Algorithm 1 below.

Algorithm 1 Basic strategy for EDM completion

INPUT: A weighted graph $G = (V, E, d)$, and a target rank $r \geq 0$;
PREPROCESSING:
1. Generate a set of cliques Θ in G;
2. Generate a set of weight functions $\{\omega_\alpha \colon \mathbb{R}^E \to \mathbb{R}_+\}_{\alpha\in\Theta}$;

for each clique α in Θ **do**
$k \leftarrow |\alpha|$;
$X_\alpha \leftarrow$ a nearest matrix in $\mathcal{S}^{k,r}_{c,+}$ to $\mathcal{K}^\dagger d_\alpha$;
$W_\alpha \leftarrow$ exposing vector of face$(X_\alpha, \mathcal{S}^k_{c,+})$ extended to $\mathcal{S}^n$ by padding zeros;
end for
$W \leftarrow \sum_{\alpha\in\Theta} \omega_\alpha(d) \cdot W_\alpha$;
$Y \leftarrow$ a nearest matrix in $\mathcal{S}^{n,n-r}_{c,+}$ to W;
$X \leftarrow$ a solution of

$$\begin{array}{ll} \text{minimize} & \|\mathcal{P} \circ \mathcal{K}(X) - d\| \\ \text{subject to} & X \in Y^\perp \cap \mathcal{S}^n_{c,+}; \end{array} \tag{3.2}$$

return X;

Some comments are in order. First, there is great flexibility in the preprocessing stage, and it will be described in Subsection 3.2.1. Secondly, finding "nearest matrices" in $\mathcal{S}^{k,r}_{c,+}$ and in $\mathcal{S}^{n,n-r}_{c,+}$ is easy as a result of the Eckart-Young theorem. The details are worked out in Appendix A. Solving the small dimensional least squares problem (3.2) is also standard. We discuss it in Appendix B. In fact, very often (under the assumptions of Theorem C.5 below) the linear least squares solution of $\min_{X\in\mathcal{V}} \|\mathcal{P} \circ \mathcal{K}(X) - d\|$ already happens to be positive definite, where $\mathcal{V}$ denotes the linear span of the face $Y^\perp \cap \mathcal{S}^n_{c,+}$. Hence this step typically does not require any optimization solver to be invoked. Indeed, this is a direct consequence of the rudimentary robustness guarantees of the method, outlined in Appendix C.

3.2 Implementing facial reduction for noisy EDM. In the following, we elaborate on some of the main ingredients of Algorithm 1:

- the choice of the clique set Θ and weight functions $\{\omega_\alpha\}_{\alpha\in\Theta}$ (in Section 3.2.1);
- the nearest-point mapping to $\mathcal{S}^{k,r}_{c,+}$ (in Appendix A); and
- the solution of the least squares problem (3.2) (in Appendix B).

To improve the solution quality of Algorithm 1, we perform a postprocessing *local refinement* step: we use the solution X from Algorithm 1 as an initial point for existing nonlinear optimization methods to find a local solution of (3.1). While general nonlinear optimization methods often fail to find a global optimal solution, when used as a local refinement procedure they can greatly improve the solution quality of Algorithm 1.

3.2.1 Choosing the clique set and the weights. We first discuss the choice of the clique set Θ, which is crucial for the success of Algorithm 1, since the exposing vector Y is formed based on the clique information. The level of rigidity of the graph known to Algorithm 1 is determined by the clique set Θ: if insufficiently many cliques are present, then the estimate of the exposing vector Y will likely be poor.

In practice, it is inefficient to compute the set of *all* cliques of G (noting that

determining whether a graph has a clique of an arbitrary given size is NP-hard), so we can only hope to find a subset of cliques of the graph. We apply a simple brute-force subroutine on the adjacency matrix H of the given graph to find a collection Θ of cliques, as in Algorithm 2 below.

Algorithm 2 Finding a collection of cliques in a graph

(INPUT) A simple graph $G = (V, E)$, integer $\bar{k} \geq 2$;
Step 1: constructing Θ_1 *and* Θ_2
$\Theta_1 \leftarrow \emptyset$;
for each vertex $v = 1, \ldots, n$ **do**
  find a subset α_v of neighbors of v that forms a clique;
  $\Theta_1 \leftarrow \Theta_1 \cup \{\alpha_v\}$;
end for
$\Theta_2 \leftarrow \{uw \in E : \nexists\, \alpha \in \Theta_1 \text{ such that } u, w \in \alpha\}$;
Step 2: constructing $\Theta_3, \ldots, \Theta_{\bar{k}}$
for $k = 3, \ldots, \bar{k}$ **do**
  $\Theta_k \leftarrow \emptyset$;
  for each $\alpha \in \Theta_{k-1}$ **do**
    if all the vertices in α share a common neighbor v **then**
      $\Theta_k \leftarrow \Theta_k \cup \{\alpha \cup \{v\}\}$;
    end if
  end for
end for
$\Theta \leftarrow \bigcup_{k=1}^{\bar{k}} \Theta_k$
(OUTPUT) Θ, a set of cliques in G of size up to $\bar{k}$.

In the first step, computing Θ_1 is very fast since it involves only repeatedly removing rows and columns of $H(\delta_v, \delta_v) + I$ that contain zero for each vertex v (where H is the $\{0, 1\}$-adjacency matrix of G having zero diagonal and δ_v is the set of neighbors of v). As for the second step, while the brute-force method of listing all cliques of fixed sizes would be prohibitive in practice, we find that the restriction imposed by Θ_1 cuts down a huge number of smaller non-maximal cliques that we need to keep track of, and the use of Θ_1 significantly speeds up the second step.

Algorithm 2 provides a very basic clique-selection framework. When working with a particular application (e.g., the sensor network localization problem, described in Section 3.3 below), the robustness of Algorithm 1 can be improved significantly when the clique selection process is specialized for that application.

Now we discuss the weight functions $\{\omega_\alpha\}_{\alpha \in \Theta}$. In Algorithm 1, we do not treat each clique in Θ equally, given that the noise in the distance measurements does not have to be uniform and it may not be possible to recover all the cliques with the same level of accuracy. We gauge the amount of noise present in the distance measurements of cliques as follows: for each clique $\alpha \in \Theta$, as before letting $d_\alpha \in \mathcal{S}^\alpha$ be the restriction of the distance measurements d to the clique, we estimate the noise present in d_α by considering the eigenvalues of $\mathcal{K}^\dagger d_\alpha$:

$$\nu_\alpha(d) := \frac{\sum_{j=1}^{|\alpha|-r} \lambda_j^2(\mathcal{K}^\dagger d_\alpha)}{0.5|\alpha|(|\alpha|-1)}. \tag{3.3}$$

Here $\lambda_j(\mathcal{K}^\dagger d_\alpha)$ refers to the j'th smallest eigenvalue of the matrix $\mathcal{K}^\dagger d_\alpha$. The value $\nu_\alpha(d)$ is the scaled squared-ℓ_2 norm of the violation of the rank constraint in the clique

α, i.e., the constraint $\operatorname{rank}(\mathcal{K}^\dagger d_\alpha) \le r$. In the case where no noise is present in the distance measurements d, we have $\nu_\alpha(d) = 0$ since the matrix $\mathcal{K}^\dagger d_\alpha \in \mathcal{S}_+^{|\alpha|}$ is of rank at most r. To each clique α, we assign the weight

$$\omega_\alpha(d) := 1 - \frac{\nu_\alpha(d)}{\sum_{\beta\in\Theta} \nu_\beta(d)}.$$

This choice of weight reflects the contribution of noise in the clique α to the total noise of all cliques (where the noise is measured by (3.3)). If a clique α is relatively noisy compared to other cliques in Θ or contains an outlier, the weight $\omega_\alpha(d)$ would be smaller than $\omega_\beta(d)$ for most $\beta \in \Theta$.

3.2.2 Postprocessing: local refinement. Following Algorithm 1, we implement a *local refinement*, which could greatly improve the solution quality. By local refinement, we mean the use of a nonlinear optimization algorithm for solving the nonconvex problem (3.1) (which has a lot of local minima) using the output of Algorithm 1 as the initial point. Local refinement has been commonly used for SDP-based algorithms for SNL problems and noisy EDM completion problem; see [6, 8].

For local refinement, we use the steepest descent subroutine from the SNL-SDP package [7]. Suppose that $X^* = P^*(P^*)^T$ is the solution of (3.2) found at the end of Algorithm 1. We use P^* as an initial point for the steepest descent method to solve the nonlinear optimization problem

$$\min_{P\in\mathbb{R}^{n\times r}} \|\mathcal{P} \circ \mathcal{K}(PP^T) - d\|^2. \tag{3.4}$$

By itself, the steepest descent method usually fails to find a *global* optimal solution of (3.4) and instead gets trapped at one of the many critical points, since the problem is highly nonconvex. On the other hand, we observe that Algorithm 1 can produce excellent initial points for such nonlinear optimization schemes.

3.3 Application on the sensor network localization problem. In this section, we apply robust facial reduction (Algorithm 1) on the anchorless *sensor network localization* (SNL) problem in $\mathbb{R}^2$. The task is to locate n wireless sensors in $\mathbb{R}^2$, given the noisy squared Euclidean distances between sensors that are within a given radio range of each other. Semidefinite programming techniques have been used extensively for the SNL problem; see for example [6–10, 23, 28, 32, 40].

One important characteristic of the SNL problem is the presence of a radio range: the distance between two sensors is available if and only if the distance is no larger than the radio range. This simple feature allows us to specialize the preprocessing step of finding the clique set Θ in the basic robust facial reduction framework, using the *nonrigid clique union technique* from [23]: we refine the feasible region in the EDM completion problem by removing solutions that violate the implicit constraints imposed by the radio range. This step approximately completes the partial EDM *locally*, while generating a set of larger cliques and reducing the error in calculating the exposing vector using the noisy distance measurements.

3.3.1 Preprocessing via clique union. The first step is to determine an ordering of the cliques $\Theta = \{\alpha_1, \ldots, \alpha_{|\Theta|}\}$ such that

$$\alpha_{j-1} \text{ and } \alpha_j \text{ intersect in at least 2 vertices, } \forall\, j. \tag{3.5}$$

Such an ordering can be found using a greedy approach: start with the largest clique $\alpha_1 \in \Theta$ and $\widehat{\Theta} := \Theta$, and for each $j \ge 2$, pick α_j among all the cliques in $\widehat{\Theta}$ intersecting

with α_{j-1} in at least 2 vertices, the one that maximizes the set difference $|\alpha_j \backslash \alpha_{j-1}|$; then update $\widehat{\Theta}$ by removing from it all the cliques whose nodes are covered by $\bigcup_{l=1}^{j} \alpha_l$.

If the graph G is sparse, such an ordering may not exist. Nonetheless, as long as G does not have a cut vertex, it is possible to cover all the vertices of G with multiple sequences of cliques, each satisfying the condition (3.5). (Note that if a noiseless SNL instance is uniquely r-localizable, in the sense of [31], for any $r \geq 2$, then the corresponding graph cannot have a cut vertex.)

Suppose that we have found an ordering of the elements of Θ satisfying the condition (3.5). Then we would perform a sequential clique union procedure, whose goals are to ensure that the matrix U found in Algorithm 1 is not too far from $\mathcal{S}_{c,+}^{n,n-r}$, and to avoid errors arising from (nearly) nonrigid intersection, which we illustrate in the following example.

EXAMPLE 3.2. Suppose that we have 5 sensors with radio range 0.05, whose true locations are given by

$$P = \begin{bmatrix} 0.4582 & 0.4793 & 0.5031 & 0.4360 & 0.4467 \\ -0.4116 & -0.3952 & -0.3221 & -0.3150 & -0.3393 \end{bmatrix}^T.$$

Then the corresponding graph is as in the left picture in Figure 1: only the distance between sensors 1 and 5 is missing. The graph has two cliques $\alpha_1 = \{1, 2, 3, 4\}$ and $\alpha_2 = \{2, 3, 4, 5\}$. Sensors $2, 3, 4$ in the clique intersection are almost collinear; α_1 and α_2 almost intersect *nonrigidly* (locally), in the sense that the realization of sensor locations:

$$\tilde{P} = \begin{bmatrix} 0.4582 & 0.4793 & 0.4360 & 0.4467 & 0.4051 \\ -0.4116 & -0.3952 & -0.3150 & -0.3393 & -0.3750 \end{bmatrix}^T$$

obtained by reflecting α_1 along the line passing through vertices 2 and 3 (in the center of Figure 1) would give almost the same partial EDM:

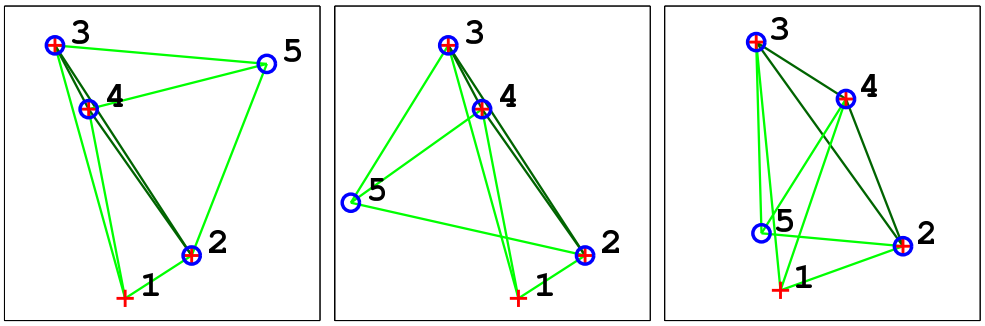
$$\left\| \mathcal{P}\big(\mathcal{K}(PP^T) - \mathcal{K}(\tilde{P}\tilde{P}^T)\big) \right\| \approx 6.84 \times 10^{-4},$$

where $\mathcal{P} : \mathcal{S}^n \to \mathbb{R}^E$ is the canonical projection. In the presence of uncertainty in distance measurements, both P and $\tilde{P}$ seem to be reasonable realization of the sensor locations. Yet with the additional knowledge of the radio range, we know that it is unlikely $\tilde{P}$ gives the approximate sensor locations, since that would mean sensors 1 and 5 are in each other's radio range.

Now suppose that the distance measurements are corrupted with 5% Gaussian noise (see the multiplicative noise model outlined in Algorithm 3). If we apply Algorithm 1 on the noisy input, then the realization of sensor locations could be as in the right picture in Figure 1, where sensors 1 and 5 are much closer than they should be. Note that the right picture in Figure 1 shows a minor perturbation of the "incorrect" realization $\tilde{P}$.

Scenarios depicted in Example 3.2 can be quite prevalent: two of the cliques in Θ may intersect (almost) nonrigidly (locally), and there would be two localizations that give similarly good least squares solutions, corresponding to two different "reflections". To ensure the robustness of the facial reduction algorithm, we perform a clique union on α_{j-1} and α_j (for each $j = 2, \ldots, |\Theta|$), by using a Procrustes rotation to match the cliques α_{j-1}, α_j at the intersection and to ensure also that the unknown distances

Fig. 1: Left: true location of 5 sensors (given by P), with edges indicating known distances. Center: realization of sensor location (given by $\tilde{P}$) satisfying the known distances but violating the radio range. Right: solution from Algorithm 1. Circles $\circ$: clique α_1; pluses $+$: clique α_2.



calculated are not much smaller than the radio range. This constitutes a *local EDM completion*: this approach localizes the two cliques α_{j-1} and α_j, and as a result we obtain the distances between all the vertices in $\alpha_{j-1} \cup \alpha_j$. After we obtain a realization of $\alpha_{j-1} \cup \alpha_j$, we use that realization to compute an exposing vector corresponding to $\alpha_{j-1} \cup \alpha_j$. This preprocessing step results in exposing vectors for the larger cliques $\beta_j = \alpha_j \cup \alpha_{j+1}$ for $j = 1, \ldots, |\Theta| - 1$. The larger cliques $\beta_1, \ldots, \beta_{|\Theta|-1}$ intersect at more vertices, lowering both the possibility that some of the clique intersections are nonrigid and the error of the exposing vector calculation.

3.3.2 Numerics. For the numerical tests, we generate random instances of the SNL problem based on a *multiplicative noise model* ([6,7]) outlined in Algorithm 3.

Algorithm 3 Multiplicative noise model

INPUT: # sensors n, noise factor σ, radio range R;
For each $i, j = 1, \ldots, n$:
- pick i.i.d. $p_i \in [-0.5, 0.5]^2$ with uniform distribution
- pick i.i.d. $\epsilon_{ij} \in \mathcal{N}(0, 1)$ (standard normal distribution)

Compute $D \in \mathcal{S}^n$ by

$$D_{ij} = (1 + \sigma\epsilon_{ij})^2 \|p_i - p_j\|^2, \quad \forall\, i, j = 1, \ldots, n;$$

Build graph $G = (\{1, \ldots, n\}, E)$, where

$$ij \in E \iff \|p_i - p_j\| \leq R;$$

$d \leftarrow [D_{ij}]_{ij \in E,\, i<j} \in \mathbb{R}^E$;
OUTPUT: noisy distance measurements $d \in \mathbb{R}^E$ and graph G.

Since the instances generated by the multiplicative noise model come with the true sensor locations, we can gauge the performance of the robust facial reduction on random instances from the multiplicative noise model using the *root-mean-square deviation* (RMSD). Suppose that the true centered locations of the sensors are stored in the rows of the matrix $P \in \mathbb{R}^{n \times 2}$, and $X \in \mathcal{S}^{n,r}_{c,+}$ is the output of Algorithm 1. Then

$X = \tilde{P}\tilde{P}^T$ for some $\tilde{P} \in \mathbb{R}^{n\times 2}$, whose rows store the estimated centered locations. The RMSD of the estimated $\tilde{P}$ relative to the true centered locations P is defined as:

$$RMSD := \min\left\{\frac{1}{\sqrt{n}}\|\tilde{P}U - P\|_F \ :\ U^TU = I,\ U \in \mathbb{R}^{r\times r}\right\}. \tag{3.6}$$

A typical output of Algorithm 1 applied on an instance generated by the multiplicative noise model is illustrated in Figure 2. While the solution produced by Algorithm 1 may not seem very impressive, with the help of standard local refinement techniques we can attain very high quality solution even with the high number of sensors and in the presence of noise.

Fig. 2: Illustration of robust facial reduction with refinement applied on an instance with 1000 sensors (no anchors) on a $[-0.5, 0.5]^2$ box, with noise factor 0.05 and radio range 0.1. Top left: using Algorithm 1 without refinement (RMSD= 28.48%R). Top right: using Algorithm 1 with refinement via the steepest descent method (RMSD= 1.05%R). Bottom: using only the steepest descent method with a randomly generated initial point (RMSD= 399.45%R). Line segments represent discrepancy between estimated and true locations.

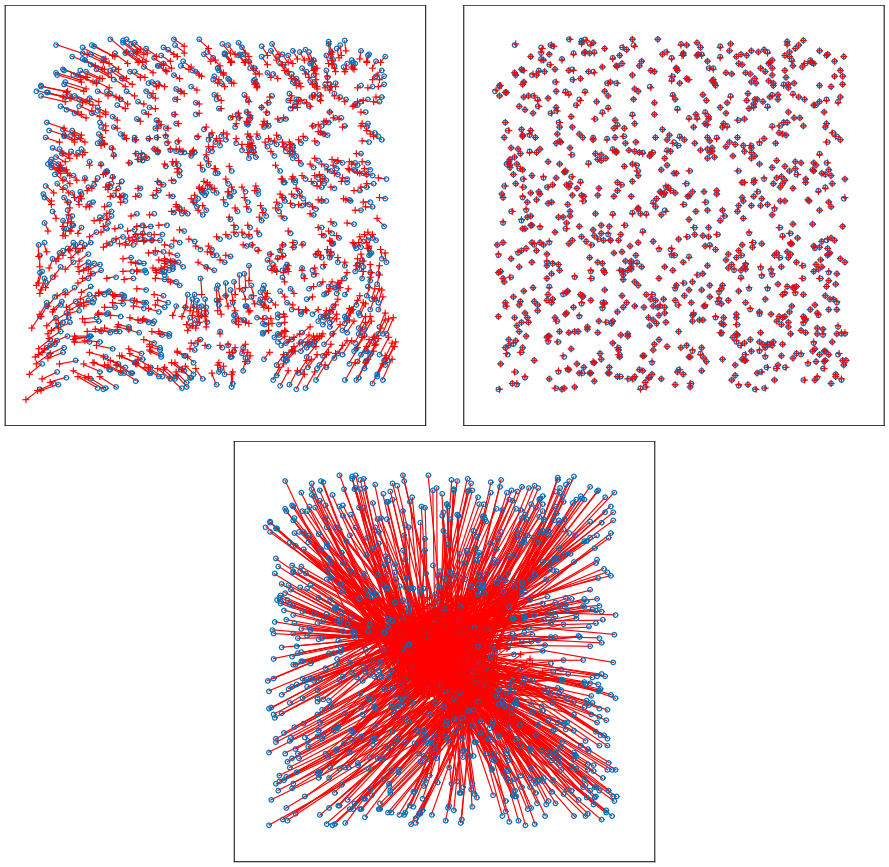


Table 1 shows some numerical results on instances with 1000 sensors (and no anchors) generated as in Algorithm 3, with varying noise factor and radio range. The tests were run on MATLAB version R2014b, on a Linux machine with Intel(R) Core(TM) i7-4650U CPU @ 1.70GH and 8 GB RAM. We show the RMSD (as a percentage of the radio range) of the solutions provided by Algorithm 1 (in the column "initial"), and also the RMSD of the solution after the local refinement using the steepest descent subroutine from SNL-SDP (in the column "refined"). We see that using Algorithm 1 together with local refinement gives rather satisfactory results. The time used by Algorithm 1 includes the selection of cliques and computation of the exposing vectors, but excludes the postprocessing time, which is reported separately. Table 2 shows some numerical results on larger instances.

Table 1: Numerical results of robust facial reduction on instances with 1000 vertices, generated using the multiplicative noise model on a $[-0.5, 0.5]^2$ grid. Each row contains the average result over 10 instances with fixed n, nf and R, where n is the number of sensors/vertices in G, R is the radio range; nf is the noise factor. The *density* refers to the ratio $\frac{\text{number of edges}}{0.5n(n-1)}$.

n	nf	R	density	Time used by Alg. 1 (s)	Time used for refinement	RMSD $\%R$ initial	RMSD $\%R$ refined
1000	0.0	0.25	15.8%	53.1	0.5	0.0%	0.0%
1000	0.1	0.25	15.7%	51.4	3.8	2.3%	0.6%
1000	0.2	0.25	15.7%	51.9	2.3	49.7%	2.0%
1000	0.3	0.25	15.7%	67.1	6.5	76.3%	2.9%
1000	0.4	0.25	15.7%	64.8	7.0	72.8%	5.6%
1000	0.1	0.15	6.2%	9.5	2.0	24.4%	1.1%
1000	0.1	0.20	10.5%	20.3	1.5	4.2%	0.8%
1000	0.1	0.25	15.7%	51.4	3.8	2.3%	0.6%
1000	0.1	0.30	21.3%	140.6	1.1	1.6%	0.5%
1000	0.1	0.35	27.8%	240.6	1.3	1.2%	0.5%

Table 2: Numerical results of robust facial reduction on instances with more than 1000 vertices, generated using the multiplicative noise model on a $[-0.5, 0.5]^2$ grid. Each row contains the average result over 5 instances with fixed n, nf and R, where n is the number of sensors/vertices in G, R is the radio range; nf is the noise factor. The *density* refers to the ratio $\frac{\text{number of edges}}{0.5n(n-1)}$.

n	nf	R	density	Time used by Alg. 1 (s)	Time used for refinement	RMSD $\%R$ initial	RMSD $\%R$ refined
2000	0.1	0.20	10.6%	223.5	3.1	2.2%	0.6%
2000	0.2	0.20	10.5%	220.2	7.5	69.5%	2.0%
2000	0.3	0.20	10.5%	222.3	7.0	81.1%	3.1%
2000	0.4	0.20	10.6%	230.2	6.8	85.1%	5.3%
3000	0.1	0.20	10.5%	1011.6	11.7	2.4%	0.5%
3000	0.2	0.20	10.4%	986.5	23.0	64.3%	1.3%
3000	0.3	0.20	10.5%	1063.9	17.8	67.5%	3.0%
3000	0.4	0.20	10.6%	1016.4	18.9	74.8%	5.0%
4000	0.1	0.20	10.5%	3184.0	13.7	1.8%	0.4%
4000	0.2	0.20	10.5%	3129.9	22.5	62.8%	1.3%
4000	0.3	0.20	10.5%	3226.1	27.3	79.8%	2.8%
4000	0.4	0.20	10.6%	3220.1	24.1	71.1%	4.9%
4000	0.2	0.175	8.3%	1618.1	30.9	56.7%	1.5%
4000	0.3	0.175	8.3%	1554.1	43.2	88.4%	3.2%
4000	0.4	0.175	8.2%	1535.8	30.5	86.1%	5.7%
4000	0.2	0.15	6.2%	801.9	41.3	90.5%	1.7%
4000	0.3	0.15	6.2%	783.0	36.2	106.4%	4.0%
4000	0.4	0.15	6.2%	759.0	30.7	109.1%	6.8%
4000	0.2	0.125	4.4%	616.8	28.1	110.3%	2.1%
4000	0.3	0.125	4.4%	541.2	29.8	128.3%	4.5%
4000	0.4	0.125	4.4%	420.7	31.2	128.6%	13.1%
5000	0.2	0.125	4.4%	905.0	59.5	110.8%	2.0%
6000	0.2	0.125	4.4%	1627.2	67.3	99.6%	2.6%
7000	0.2	0.125	4.4%	2237.8	93.9	100.3%	1.9%
8000	0.2	0.125	4.4%	3704.7	120.4	92.6%	1.9%
9000	0.2	0.125	4.4%	5883.7	87.6	97.9%	1.9%

4 The Pareto frontier of the unfolding heuristic. The facial reduction algorithm presented in the previous section is effective when G is fairly dense (so that

many cliques are available) and the SDP relaxation of the EDM completion problem without noise is exact. In this section, we consider problems at the opposite end of the spectrum. We will suppose that G is sparse and we will seek a low rank solution approximately solving the SDP (2.3). To this end, consider the problem:

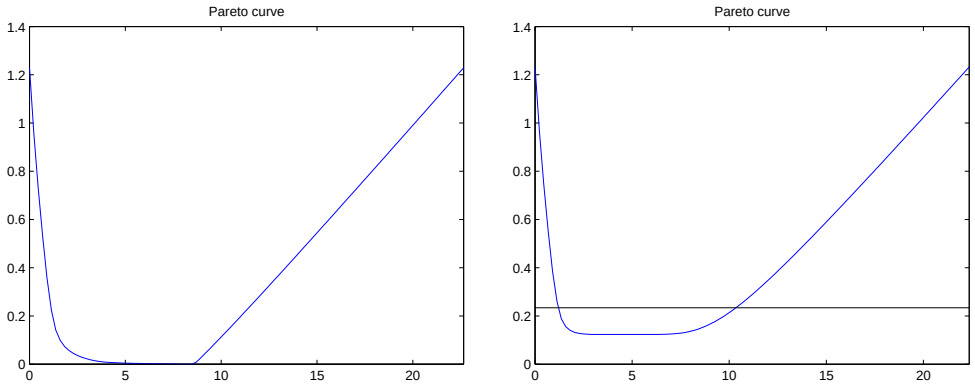
$$\begin{array}{lll} & \text{maximize} & \operatorname{tr} X \\ (4.1) & \text{subject to} & \|\mathcal{P} \circ \mathcal{K}(X) - d\| \leq \sigma \\ & & Xe = 0 \\ & & X \succeq 0. \end{array}$$

Here, an estimate of the tolerance $\sigma > 0$ on the misfit is typically available based on the physical source of the noise. Trace maximization encourages the solution X to have a lower rank. This is in contrast to the usual min-trace strategy in compressed sensing; see [3, 38, 39] for a discussion. Indeed, as was mentioned in the introduction in terms of the factorization $X = PP^T$, the equality $\operatorname{tr}(X) = \frac{1}{2n}\sum_{i,j=1}^{n} \|p_i - p_j\|^2$ holds, where p_i are the rows of P. Thus trace maximization serves to "flatten" the realization of the graph. We focus on the max-trace regularizer, though an entirely analogous analysis holds for min-trace. At the end of the section we compare the two.

We propose a first-order method for this problem using a Pareto search strategy originating in portfolio optimization. This technique has recently garnered much attention in wider generality; e.g., [37–39]. The idea is simple: exchange the objective and the difficult constraint, and then use the easier flipped problem to solve the original. Thus we are led to consider the parametric optimization problem

$$\begin{array}{lll} \varphi(\tau) := & \text{minimize} & \|\mathcal{P} \circ \mathcal{K}(X) - d\| \\ (4.2) & \text{subject to} & \operatorname{tr} X = \tau \\ & & Xe = 0 \\ & & X \succeq 0. \end{array}$$

See Figure 3 below for an illustration.



(a) $\sigma = 0$ (b) $\sigma = 0.24$

Fig. 3: Graph of φ with noise tolerance $\sigma = 0$ and $\sigma = 0.24$

Observe that the evaluation of $\varphi(\tau)$ is well adapted to first-order methods, since the feasible region is so simple. It is well-known that φ is a convex function, and therefore to solve the original problem (4.1), we simply need to find the largest τ satisfying $\varphi(\tau) \leq \sigma$. We note that the smallest value of τ satisfying $\varphi(\tau) \leq \sigma$ corresponds instead to minimizing the trace. We propose to evaluate $\varphi(\tau)$ by the Frank-Wolfe algorithm and then solve for the needed value of τ by an inexact Newton method. We will see that this leads to an *infeasible method* that is unaffected by the inherent ill-conditioning of the underlying EDM completion problem discussed in the previous sections.

4.1 An inexact Newton method. We now describe an inexact Newton method for finding the largest value τ satisfying $\varphi(\tau) \leq \sigma$. To this end, we introduce the following definition.

DEFINITION 4.1 (Affine minorant oracle). Given a function $v : I \to \mathbb{R}$ on an interval $I \subset \mathbb{R}$, an *affine minorant oracle* is a mapping $\mathcal{O}_v$ that assigns to each pair $(t, \alpha) \in I \times [1, \infty)$ real numbers (l, u, s) such that $0 \leq l \leq v(t) \leq u$, $\frac{u}{l} \leq \alpha$, and the affine function $t' \mapsto l + s(t' - x)$ minorizes v.

For the EDM completion problem, the function v is given by $v(\tau) = \varphi(\tau) - \sigma$. The inexact Newton method based on an affine minorant oracle is described in Algorithm 4.

Algorithm 4 Inexact Newton method

Input: Convex function $v\colon I \to \mathbb{R}$ on an interval $I \subset \mathbb{R}$ via an affine minorant oracle $\mathcal{O}_v$, target accuracy $\beta > 0$, initial point $t_0 \in I$ with $v(t_0) > 0$, and a constant $\alpha \in (1, 2)$.

$(l_0, u_0, s_0) := \mathcal{O}_v(t_0, \alpha)$;
$k \leftarrow 0$;
$l_0 \leftarrow 0$;
$u_0 \leftarrow +\infty$;
while $\frac{u_k}{l_k} > \alpha$ and $u_k > \beta$ **do**
  $t_{k+1} \leftarrow t_k - \frac{l_k}{s_k}$;
  $(l_{k+1}, s_{k+1}) := \mathcal{O}_v(t_{k+1}, \alpha)$;
  $k \leftarrow k + 1$;
end while
return t_k;

It can be shown that the iterates t_k generated by the inexact Newton method (Algorithm 4), when applied to a convex function $v\colon I \to \mathbb{R}$ having a root on the interval I, converge to the root $\bar{t}$ of v closest to t_0. Moreover, the convergence is linear in function value: the algorithm is guaranteed to terminate after at most

$$K \leq \max\left\{\log_{2/\alpha}\left(\frac{|s_0|R}{\beta}\right) + \log_{2/\alpha}(2) \cdot \log_{2/\alpha}\left(\frac{2l_0}{\beta}\right), 1\right\}$$

iterations, where we set $R = \bar{t} - t_0$. For a proof and a discussion, see the preprint [4].

Thus to implement this method, for the problem (4.1), we must describe an affine minorant oracle for $v(t) = \varphi(t) - \sigma$. Then, after the number of iterations given above, we can obtain a centered PSD matrix X satisfying

$$\|\mathcal{P} \circ \mathcal{K}(X) - d\| \leq \sigma + \beta \qquad \text{and} \qquad \operatorname{tr}(X) \geq OPT,$$

where OPT denotes the optimal value of (4.1). A key observation is that the derivative of v at the root *does not* appear in the iteration bound. This is important because for the function $v(t) = \varphi(t) - \sigma$, the inherent ill-conditioning of (4.1) can lead to the derivative of v at the root being close to zero.

4.2 Solving the inner subproblem with Frank-Wolfe algorithm. In this subsection, we describe an affine minorant oracle for $\varphi(\tau)$ based on the *Frank-Wolfe algorithm* [17], which has recently found many applications in machine learning (see, e.g., [18, 21]). Throughout, we fix a value τ satisfying $\varphi(\tau) > \sigma$. To apply the *Frank-Wolfe algorithm*, we must first square the objective in (4.1) to make it smooth. To simplify notation, define

$$\mathcal{A} := \mathcal{P} \circ \mathcal{K}, \quad f(X) := \frac{1}{2}\|\mathcal{A}(X) - d\|^2 \quad \text{and} \quad \mathcal{D} := \{X \succeq 0 : \operatorname{tr} X = 1,\, Xe = 0\}.$$

Thus we seek a solution to

$$\min\{f(X) : X \in \tau\mathcal{D}\}.$$

The Frank-Wolfe scheme is described in Algorithm 5.

Algorithm 5 Affine minorant oracle based on the Frank-Wolfe algorithm

Input: $\tau \geq 0$, relative tolerance $\alpha > 1$, and $\beta > 0$.
Let $k \leftarrow 0$, $l_0 \leftarrow \frac{1}{2}\sigma^2$, and $u_0 \leftarrow +\infty$. Pick any point X_0 in $\tau\mathcal{D}$.
while $\sqrt{2u_k} - \sigma > \alpha(\sqrt{2l_k} - \sigma)$ and $\sqrt{2u_k} - \sigma > \beta$ **do**
  Choose a direction

$$S_k \in \operatorname*{argmin}_{S \in \tau\mathcal{D}} \langle \nabla f(X_k), S\rangle; \tag{4.3}$$

  Set the stepsize: $\gamma_k \in \operatorname{argmin}_{\gamma\in[0,1]} f(X_k + \gamma(S_k - X_k))$;
  Update the iterate: $X_{k+1} \leftarrow X_k + \gamma_k(S_k - X_k)$;
  Update the upper bound: $u_{k+1} \leftarrow f(X_{k+1})$;
  Update the lower bound:

$$l_{k+1} \leftarrow \max\left\{l_k, f(X_k) + \langle \nabla f(X_k), S_k - X_k\rangle\right\};$$

  Increment the iterate: $k \leftarrow k + 1$;
  if $l_{k+1} > l_k$ **then**
    $y \leftarrow d - \mathcal{P} \circ \mathcal{K}(X_k)$;
    $X \leftarrow X_k$
    $S \leftarrow S_k$
  end if
end while
$l \leftarrow \frac{l_k + \frac{1}{2}\|y\|_2^2}{\|y\|_2} - \sigma$;
$u \leftarrow \sqrt{2u_k} - \sigma$;
$s = \frac{1}{\tau\|y\|}\langle \nabla f(X), S\rangle$;
return (l, u, s);

The computational burden of the method is the minimization problem (4.3). To elaborate on this, observe first that

$$\nabla f(X) = \mathcal{K}^* \circ \mathcal{P}^*\big(\mathcal{P} \circ \mathcal{K}(X) - d\big).$$

Notice that the matrix $\mathcal{K}^* \circ \mathcal{P}^*\big(\mathcal{P} \circ \mathcal{K}(X) - d\big)$ has the same sparsity pattern, modulo the diagonal, as the adjacency matrix of the graph. As a result, when the graph G is *sparse*, we claim that the linear optimization problem (4.3) is easy to solve. Indeed, observe $\nabla f(X)e = 0$ and consequently an easy computation shows that $\min_{S \in \tau\mathcal{D}} \langle \nabla f(X), S\rangle$ equals τ times the minimal eigenvalue of the restriction of $\nabla f(X)$ to $e^\perp$; this minimum in turn is attained at the matrix $\tau v v^T$ where v is the corresponding unit-length eigenvector. Thus to solve (4.3) we must find only the minimal eigenvalue-eigenvector pair of $\nabla f(X)$ on $e^\perp$, which can be done fairly quickly by a Lanczos method, and in particular, by orders of magnitude faster than the full eigenvalue decomposition. Thus, the Frank-Wolfe method is perfectly adapted to our problem instance.

THEOREM 4.1 (Affine minorant oracle). *Algorithm 5 is an affine minorant oracle for the function $v(\tau) := \varphi(\tau) - \sigma$.*

Proof. We first claim that upon termination of Algorithm 5, the line $t' \mapsto l + s(\tau - \tau')$ is a lower minorant of $v(\tau') - \sigma$. To see this, observe that the dual of the problem

$$\varphi(\tau) = \min_{X \in \tau\mathcal{D}} \ \|\mathcal{A}(X) - d\|$$

is given by

$$\max_{z \in \mathbb{R}^E:\, \|z\| \le 1} \ h_\tau(z) := \langle d, z\rangle - \tau\delta^*_{\mathcal{D}}(\mathcal{A}^* z),$$

where $\delta^*_{\mathcal{D}}$ denotes the support function of $\mathcal{D}$. Then by weak duality for any vector z with $\|z\|_2 \le 1$ and any τ', we have the inequality

$$\varphi(\tau') \ge h_{\tau'}(z) = \langle d, z\rangle - \tau'\delta^*_{\mathcal{D}}(\mathcal{A}^* z) = h_\tau(z) - (\tau' - \tau)\delta^*_{\mathcal{D}}(\mathcal{A}^* z). \tag{4.4}$$

Hence the affine function $\tau' \mapsto h_\tau(z) - (\tau' - \tau)\delta^*_{\mathcal{D}}(\mathcal{A}^* z)$ minorizes the value function $\varphi(\tau')$. Now a quick computation shows that upon termination of Algorithm 5, we have

$$l_k + \frac{1}{2}\|y\|^2 = h_\tau(y). \tag{4.5}$$

Setting $z = \frac{y}{\|y\|_2}$ in inequality (4.4) and using the identity (4.5), we obtain for all $\tau' \in \mathbb{R}$ the inequality

$$\begin{aligned}\varphi(\tau') &\ge \frac{l_k + \frac{1}{2}\|y\|^2}{\|y\|} - (\tau' - \tau)\frac{\delta^*_{\mathcal{D}}(\mathcal{A}^* y)}{\|y\|} \\ &= l + \sigma + s(\tau' - \tau).\end{aligned}$$

Hence the line $t' \mapsto l + s(\tau - \tau')$ is a lower minorant of $v(\tau') - \sigma$, as claimed. Next, we show that upon termination, the inequality $\frac{u}{l} \le \alpha$ holds. To see this, observe that

$$\begin{aligned}\frac{u}{l} &= \frac{2\|y\|u}{2l_k + \|y\|^2 - 2\sigma\|y\|} \le \frac{2\|y\|u}{\left(\frac{u+\alpha\sigma}{\alpha}\right)^2 + \|y\|^2 - 2\sigma\|y\|} \\ &= \alpha\left(\frac{2\|\alpha y\|u}{(u + \alpha\sigma)^2 + \|\alpha y\|^2 - 2\alpha\sigma\|\alpha y\|}\right).\end{aligned}$$

Now, observe that the numerator of the rightmost expression is always less than the denominator:

$$\Big((u+\alpha\sigma)^2 + \|\alpha y\|^2 - 2\alpha\sigma\|\alpha y\| \Big) - 2\|\alpha y\| u = (u+\alpha\sigma)^2 + \|\alpha y\|^2 - 2\|\alpha y\|(u+\alpha\sigma)$$
$$= (u + \alpha\sigma - \|\alpha y\|)^2 \geq 0.$$

We conclude that $\frac{u}{l} \leq \alpha$, as claimed. This completes the proof. □

Thus Algorithm 5 is an affine minorant oracle for $\varphi - \sigma$, and linear convergence guarantees of the inexact Newton method (Algorithm 4) apply.

Finally let us examine the iteration complexity of the Frank-Wolfe algorithm itself. Suppose that that for some iterate k, we have $\frac{\sqrt{2u_k}-\sigma}{\sqrt{2l_k}-\sigma} > \alpha$ and $\sqrt{2u_k} - \sigma > \beta$. Dropping the subscripts k for clarity, observe that $\frac{\sqrt{2u}-\sqrt{2l}}{\beta} > \frac{(\sqrt{2u}-\sigma)-(\sqrt{2l}-\sigma)}{\sqrt{2u}-\sigma} > 1 - \frac{1}{\alpha}$. Consequently in terms of the duality gap $\epsilon := u - l$, we have

$$2\epsilon \geq (\sqrt{2u} - \sqrt{2l})^2 > \beta^2 \left(1 - \frac{1}{\alpha}\right)^2.$$

Hence Algorithm 5 terminates provided $\epsilon \leq \frac{1}{2}\beta^2 \left(1 - \frac{1}{\alpha}\right)^2$. Standard convergence guarantees of the Frank-Wolfe method (e.g., [16, 17, 21]), therefore imply that the method terminates after $\mathcal{O}\left(\frac{\tau_k L^2}{\beta^2}\right)$ iterations, where L is the Lipschitz constant of the gradient ∇f.

Summarizing, consider an instance of the problem (4.1) with optimal value OPT. Then given a target accuracy $\beta > 0$ on the misfit $\|\mathcal{P} \circ \mathcal{K}(\cdot) - d\|$, we can find a matrix $X \succeq 0$ with $Xe = 0$ that is super-optimal and nearly feasible, meaning

$$\operatorname{tr}(X) \geq \mathrm{OPT} \qquad \text{and} \qquad \|\mathcal{P} \circ \mathcal{K}(X) - d\| \leq \sigma + \beta$$

using at most $\max\left\{\log_{2/\alpha}\left(\frac{|s_0|R}{\beta}\right) + \log_{2/\alpha}(2) \cdot \log_{2/\alpha}\left(\frac{2l_0}{\beta}\right), 1\right\}$ inexact Newton iterations[1], with each inner Frank-Wolfe algorithm terminating in at most $\mathcal{O}\left(\frac{\tau_0 L^2}{\beta^2}\right)$ many iterations. Finally, we mention that in the implementation of the method, it is essential to warm start the Frank-Wolfe algorithm using iterates from previous Newton iterations.

4.3 Comparison of minimal and maximal trace problems. It is interesting to compare the properties of the minimal trace solution

$$\begin{array}{ll} \text{minimize} & \operatorname{tr} X \\ \text{subject to} & \|\mathcal{P} \circ \mathcal{K}(X) - d\| \leq \sigma, \quad Xe = 0, \quad X \succeq 0, \end{array}$$

and the maximal trace solution

$$\begin{array}{ll} \text{maximize} & \operatorname{tr} X \\ \text{subject to} & \|\mathcal{P} \circ \mathcal{K}(X) - d\| \leq \sigma, \quad Xe = 0, \quad X \succeq 0. \end{array}$$

In this section, we illustrate the difference using the proposed algorithm. Consider the following EDM completion problem coming from wireless sensor networks (Figure 4). The iterates generated by the inexact Newton method are plotted in Figure 5.

[1] As before $|s_0|$ is the slope of the value function v at τ_0 and $R = \tau_0 - \mathrm{OPT}$.

Fig. 4: An instance of the sensor network localization problem on $n = 50$ nodes with radio range $R = 0.35$ and noise factor $nf = 0.1$.

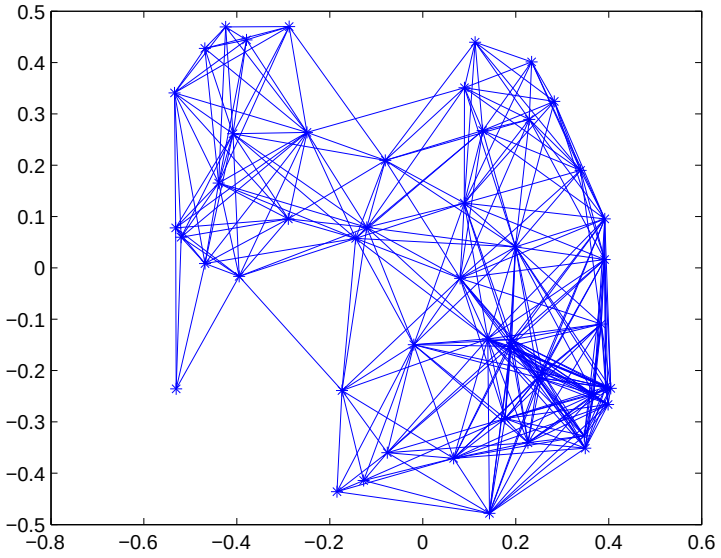
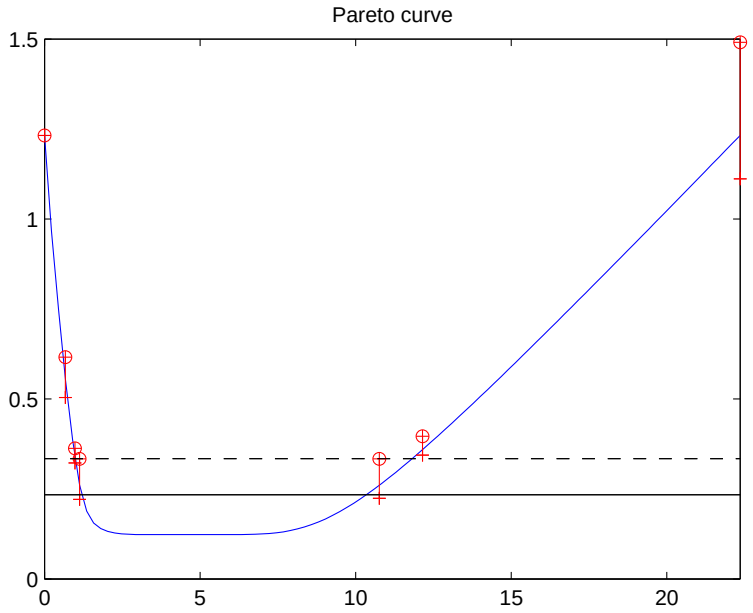
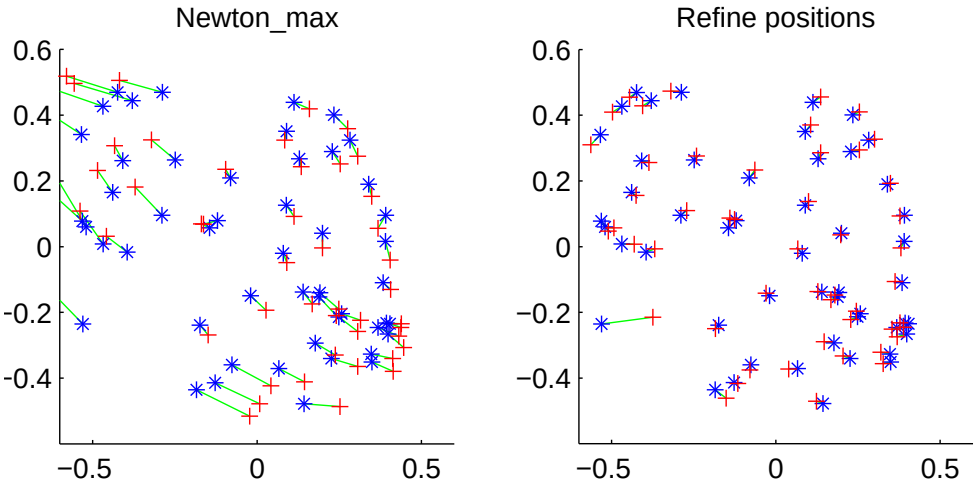


Fig. 5: Graph of φ and inexact Newton iterates for solving the minimal trace and the maximal trace problems. Here $\sigma = 0.2341$ (the dark horizontal line) and the tolerance on the misfit in the l_2-norm (the dashed horizontal line) is $\sigma + \beta = 0.3341$.



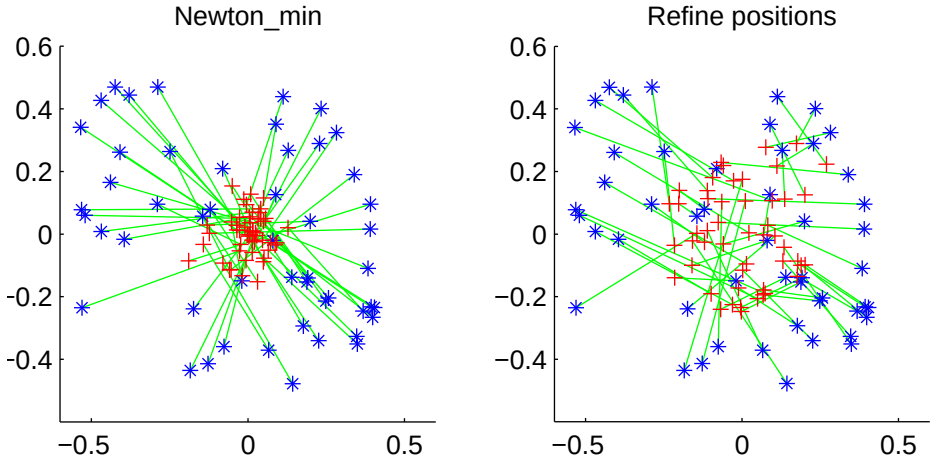
Let us consider first the maximal trace solution X. In Figure 6, the asterisks $*$ indicate the true locations of points in both pictures. In the picture on the left, the pluses $+$ indicate the points corresponding to the *maximal trace* solution X after projecting X onto rank 2 PSD matrices, while in the picture on the right they denote the locations of these points after local refinement. The edges indicate the deviations.

Fig. 6: Maximal trace solution.



In contrast, we now examine the minimal trace solution, Figure 7. Notice that even after a local refinement stage, the realization is very far from the true realization that we seek, an indication that a local search algorithm has converged to an extraneous critical point of the least squares objective. We have found this type of behavior to be very typical in our numerical experiments.

Fig. 7: Minimal trace solution.



Finally we mention an interesting difference between the maximal trace and the

minimal trace solutions as far the as the value function φ is concerned. When $\sigma = 0$, the typical picture of the graph of φ is illustrated in Figure 8. The different shapes of

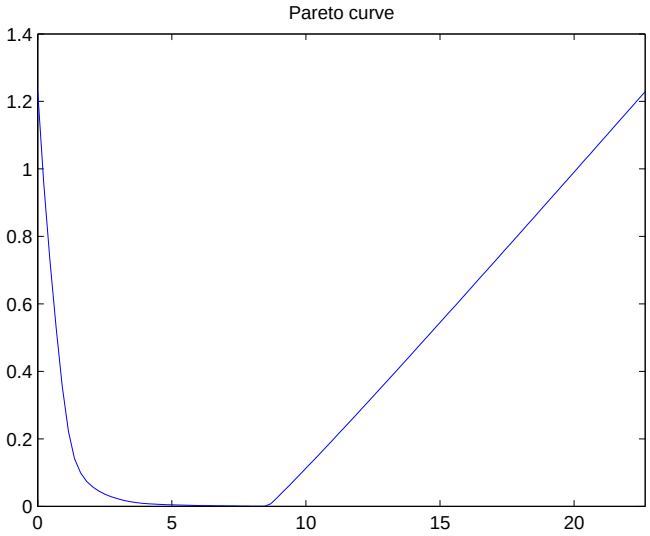


Fig. 8: Graph of φ with $\sigma = 0$.

the curve on the left and on the right sides are striking. To elucidate this phenomenon, consider the primal problem

$$\begin{aligned} &\text{minimize} && \operatorname{tr} X \\ &\text{subject to} && \mathcal{P} \circ \mathcal{K}(X) = d, \quad Xe = 0, \quad X \succeq 0, \end{aligned}$$

and its dual

$$\begin{aligned} &\text{maximize} && y^T d \\ &\text{subject to} && \mathcal{K}^* \circ \mathcal{P}^*(y) + \beta e e^T \preceq I, \end{aligned}$$

In particular, the dual is strictly feasible and hence there is no duality gap. On the other hand, suppose that the dual optimal value is attained by some pair (y, β) and suppose without loss of generality that $\mathcal{K}^* \circ \mathcal{P}^*(y)$ has an eigenvalue equal to one corresponding to an eigenvector orthogonal to e. Then letting τ be the optimal value (the minimal trace), and appealing to equation (4.4) we deduce

$$\begin{aligned} \varphi(\tau') &\geq \frac{1}{\|y\|} \left(d^T y - \tau \delta^*_{\mathcal{D}}(\mathcal{K}^* \circ \mathcal{P}^*(y)) - (\tau' - \tau) \delta^*_{\mathcal{D}}(\mathcal{K}^* \circ \mathcal{P}^*(y)) \right) \\ &\geq -\frac{(\tau' - \tau)}{\|y\|} \qquad \text{for all } \tau'. \end{aligned}$$

Hence the fact that slope $\varphi'(\tau)$ is close to zero in Figure 8 indicates that the dual problem is either unattained (not surprising since the primal fails the Slater condition) or that the dual is attained only by vectors y of very large magnitude. The reason why such phenomenon does not occur for the max-trace problem is an intriguing subject for further investigation.

4.4 Numerical illustration. In this section, we illustrate the proposed method on sensor network localization instances. The data was generated in the same manner as the numerical experiments in Section 3.3. The following tables illustrate the outcome of the method by varying the noise factor (nf), the radio range (R), and the number of sensors (n). Throughout we have fixed the tolerance on the misfit $\|\mathcal{P} \circ \mathcal{K}(X) - d\| \leq \sigma + 0.1$. We report the density of the graph, the CPU time that our algorithms runs, the number of the Frank-Wolfe iterations (FW#), the RMSD of the resulting configuration, the RMSD of the configuration after local refinement, and the CPU time that the local refinement algorithm takes. The tests were run on MATLAB version R2011b, on a Linux machine with an Intel(R) Xeon(R) CPU E3-1225 @ 3.10GHz and 12 GB RAM.

Table 3: Numerical results for the Pareto search strategy

n	nf	R	density	CPU time (s)	FW#	RMSD %R initial	RMSD %R refined	Refine time (s)
1000	0.0	0.10	2.9%	9.2	181	36.3%	0.1%	1.2
1000	0.1	0.10	2.9%	8.8	147	59.6%	3.6%	1.2
1000	0.2	0.10	2.9%	7.3	136	89.6%	7.5%	1.2
1000	0.3	0.10	2.9%	7.9	140	115.1%	11.8%	1.2
1000	0.1	0.10	2.9%	8.9	147	59.6%	3.6%	1.2
1000	0.1	0.15	6.3%	6.6	176	22.6%	2.1%	1.1
1000	0.1	0.20	10.7%	12.4	356	11.5%	1.4%	1.3
1000	0.1	0.25	15.9%	20.3	586	7.3%	1.2%	1.6
1000	0.1	0.30	22.0%	45.0	1074	4.9%	0.9%	1.4
1000	0.2	0.10	2.9%	7.3	136	89.6%	7.5%	1.2
2000	0.2	0.10	2.9%	17.1	169	66.3%	4.7%	5.0
3000	0.2	0.10	2.9%	30.8	189	56.4%	3.5%	5.0
4000	0.2	0.08	1.9%	63.8	227	80.6%	3.7%	11.6
5000	0.2	0.08	1.9%	75.1	179	74.0%	3.3%	16.9
6000	0.2	0.08	1.9%	179.6	264	68.3%	3.0%	26.9
7000	0.2	0.06	1.1%	253.7	345	119.1%	4.2%	28.8
8000	0.2	0.06	1.1%	355.4	370	112.0%	3.5%	25.8
9000	0.2	0.06	1.1%	425.8	338	108.0%	3.4%	42.4
10000	0.2	0.06	1.1%	611.9	408	101.9%	3.1%	55.1
11000	0.2	0.05	0.8%	744.9	435	149.3%	3.8%	39.5
12000	0.2	0.05	0.8%	981.4	498	143.1%	3.9%	36.1
13000	0.2	0.05	0.8%	1240.6	526	138.4%	4.5%	67.3
14000	0.2	0.05	0.8%	1219.4	468	131.8%	6.7%	80.4
15000	0.2	0.05	0.8%	1518.8	490	131.0%	5.1%	89.2

5 Conclusion and work in progress. In this paper, we described two algorithms (robust facial reduction and a search along the Pareto frontier) to solve the EDM completion problem with possibly inaccurate distance measurements, which has important applications and is numerically challenging. The two algorithms are intended for EDM completion problems of different densities: the Pareto frontier algorithm discussed in Section 4 is designed for sparse graphs whereas the robust facial reduction outlined in Algorithm 1 in Section 3 tends to work better for denser graphs. Though not studied in this work, it is possible to develop a distributed implementation of the robust facial reduction technique in order to solve even larger scale completion problems. The Pareto frontier estimation technique is promising for handling large scale EDM completion problems, since first-order methods become immediately applicable and sparsity of the underlying graph can be exploited when searching for a maximum eigenvalue-eigenvector pair via a Lanczos procedure. Numerical experiments have illustrated the effectiveness of both strategies.

Appendix A. Nearest-point mapping to $\mathcal{S}^{k,r}_{c,+}$.

We now describe how to evaluate the nearest-point-mapping to the set $\mathcal{S}^{k,r}_{c,+}$—an easy and standard operation due to the Eckart-Young Theorem. To describe this operation, consider any matrix $X \in \mathcal{S}^n$ and a set $\mathcal{Q} \subset \mathcal{S}^n$. Define the *distance function* and the *projection*, respectively:

$$\operatorname{dist}(X;\mathcal{Q}) = \inf_{Y\in\mathcal{Q}} \|X-Y\|_F,$$

$$\operatorname{proj}(X;\mathcal{Q}) = \{Y \in \mathcal{Q} : \|X-Y\|_F = \operatorname{dist}(X;\mathcal{Q})\}.$$

In this notation, we would like to find a matrix Y in the set $\operatorname{proj}(X;\mathcal{S}^{k,r}_{c,+})$. To this end, let $\left[\frac{1}{\sqrt{k}}e \quad U\right]$ be any $k\times k$ orthogonal matrix. First dealing with the centering constraint, one can verify

$$\operatorname{proj}\left(X;\mathcal{S}^{k,r}_{c,+}\right) = U\Big[\operatorname{proj}\left(U^TXU;\mathcal{S}^{k-1,r}_{+}\right)\Big]U^T.$$

On the other hand, we have

$$\operatorname{proj}\left(Z;\mathcal{S}^{k-1,r}_{+}\right) = W\operatorname{Diag}\Big(0,\ldots,0,\lambda^+_{k-r}(Z),\ldots,\lambda^+_{k-1}(Z)\Big)W^T,$$

where $\lambda_1(Z)\le\ldots\le\lambda_{k-1}(Z)$ are the eigenvalues of Z and the subscript $\lambda^+_i(Z)$ refers to their positive part, and W is any orthogonal matrix in the eigenvalue decomposition $Z = W\operatorname{Diag}(\lambda(Z))W^T$. Thus computing a matrix in $\operatorname{proj}(X;\mathcal{S}^{k,r}_{c,+})$ requires no more than an eigenvalue decomposition.

Appendix B. Solving the small least squares problem. We now describe how to easily solve the least squares system (3.2). Typically, the matrix Y will have rank $n-r$. Then the face $\mathcal{S}^n_{c,+}\cap Y^\perp$ can be written as $\mathcal{S}^n_{c,+}\cap Y^\perp = U\mathcal{S}^r_+U^T$, where the $n\times r$ matrix U has as columns an orthonormal basis for the kernel of Y. Consequently we are interested in solving an optimization problem of the form

$$\begin{aligned}\min_{Z}\ & \|\mathcal{A}(Z)-d\|_2^2\\ \text{s.t.}\ & Z\in\mathcal{S}^r_+,\end{aligned}$$

where the linear operator $\mathcal{A}\colon \mathcal{S}^n \to \mathbb{R}^E$ is defined by $[\mathcal{A}(Z)]_{ij} = [\mathcal{K}(UZU^T)]_{ij}$ for all $ij \in E$. Let $\operatorname{svec}(Z)$ be the vectorization of Z and let A be a $|E| \times \frac{r(r+1)}{2}$ matrix representation of the operator $\mathcal{A}$. Thus we are interested in solving the system

$$\begin{aligned} \min_{Z} \ & \|A(\operatorname{svec} Z) - d\|_2^2 \\ \text{s.t.} \ & Z \in \mathcal{S}_+^r, \end{aligned} \tag{B.1}$$

where A is a tall-skinny matrix. One approach now is simply to expand the objective

$$\|A(\operatorname{svec} Z) - d\|_2^2 = \langle (A^T A)(\operatorname{svec} Z), \operatorname{svec} Z\rangle - 2\langle A^T d, \operatorname{svec} Z\rangle + \|d\|^2,$$

and then apply any standard iterative method to solve the problem (B.1). Alternatively, one may first form an economic QR factorization $A = QR$ (where $Q \in \mathbb{R}^{|E|\times \frac{1}{2}r(r+1)}$ has orthonormal columns and $R \in \mathbb{R}^{\frac{1}{2}r(r+1)\times\frac{1}{2}r(r+1)}$ is upper triangular) and then write the objective as $\|A(\operatorname{svec} Z) - d\|_2^2 = \|R(\operatorname{svec} Z) - Q^T d\|^2$. We can then pose the problem (B.1) as a small linear optimization problem over the product of the semidefinite cone $\mathcal{S}_+^r$ and a small second-order cone of dimension $\mathbb{R}^{\frac{r(r+1)}{2}}$, and quickly solve it by an off-the-shelf Interior Point Method.

In practice, very often the cone constraint in (3.2) is *inactive*. The reason is that under reasonable conditions (see Theorem 2.2), in a noiseless situation, there is a unique solution to the equation $\mathcal{A}(Z) = d$, which happens to be positive definite. Hence by the robustness guarantees (Theorem C.5) a small amount of noise in d will lead to a matrix solving $\min_Z \|A(\operatorname{svec} Z) - d\|_2^2$ that is automatically positive definite. Heuristically, we can simply drop the cone constraint in (3.2) and consider the unconstrained least squares problem

$$\min_{Z} \ \|A(\operatorname{svec} Z) - d\|_2^2, \tag{B.2}$$

which can be solved very efficiently by classical methods. With this observation, we often can solve (3.2) *without using any optimization software*.

Appendix C. Robustness of facial reduction. In this section, we provide rudimentary robustness guarantees on the Algorithm 1. To this end, consider two $n\times r$ matrices U and V, each with orthonormal columns. Then the *principal angles* between $\operatorname{range} U$ and $\operatorname{range} V$ are the arccosines of the singular values of $U^T V$. We will denote the vector of principal angles between these subspaces, arranged in nondecreasing order, by Γ. The symbols $\sin^k(\Gamma)$ and $\cos^k(\Gamma)$ will have obvious meanings. Thus the vector of singular values $\sigma(U^T V)$, arranged in nondecreasing order, coincides with $\cos(\Gamma)$. Consequently in terms of the matrix

$$\Delta = I - (V^T U)(V^T U)^T,$$

the eigenvalue vector $\lambda(\Delta)$ coincides with $\sin^2(\Gamma)$. An important property is that the principal angles between $\operatorname{range} U$ and $\operatorname{range} V$ and the principal angles between $(\operatorname{range} U)^\perp$ and $(\operatorname{range} V)^\perp$, coincide modulo extra $\frac{\pi}{2}$ angles that appear for dimensional reasons. The following is a deep result that is fundamental to our analysis [13–15]. It estimates the deviation in range spaces of matrices that are nearby in norm.

THEOREM C.1 (Distances and principal angles). *Consider two matrices $X, Y \in \mathcal{S}_+^n$ of rank r and let Γ be the vector of principal angles between* $\operatorname{range} X$ *and* $\operatorname{range} Y$.

Then the inequality

$$\|\sin(\Gamma)\| \leq \frac{\|X-Y\|}{\delta(X,Y)} \qquad \textit{holds,}$$

where $\delta(X,Y) := \min\{\lambda_r(X), \lambda_r(Y)\}$.

The following is immediate now.

COROLLARY C.2 (Deviation in exposing vectors). *Consider two rank* r *matrices* $X, Y \in \mathcal{S}^n_+$ *and let* U *and* V *be* $n \times r$ *matrices with orthonormal columns that span* $\ker X$ *and* $\ker Y$ *respectively. Then we have*

$$\|UU^T - VV^T\| = \sqrt{2}\left(\frac{\|X-Y\|}{\delta(X,Y)}\right).$$

Proof. Observe $\|UU^T - VV^T\|^2 = 2\operatorname{tr}(I - (V^TU)(V^TU)^T) = 2\|\sin(\Theta)\|^2$. Applying Theorem C.1, the result follows. ∎

Next, we will need the following lemma.

LEMMA C.3 (Projections onto subsets of symmetric matrices). *For any* $n \times r$*-matrix* U *with orthonormal columns, and a matrix* $X \in \mathcal{S}^n$*, we have*

$$\operatorname{proj}(X; U\mathcal{S}^rU^T) = UU^TXUU^T, \tag{C.1}$$

and for any subset $\mathcal{Q} \in \mathcal{S}^r$*, we have*

$$\operatorname{proj}(X; U\mathcal{Q}U^T) = U\operatorname{proj}(U^TXU; \mathcal{Q})U^T. \tag{C.2}$$

Proof. Optimality conditions for the optimization problem

$$\min_{Y \in \mathcal{S}^r} \|X - UYU^T\|^2$$

immediately imply $\operatorname{proj}(X; U\mathcal{S}^rU^T) = UU^TXUU^T$. Since $U\mathcal{Q}U^T$ is contained in the linear space $U\mathcal{S}^rU^T$, the projection $\operatorname{proj}(X; U\mathcal{Q}U^T)$ factors into a composition

$$\operatorname{proj}(X; U\mathcal{Q}U^T) = \operatorname{proj}\Big(\operatorname{proj}(X; U\mathcal{S}^rU^T); U\mathcal{Q}U^T\Big),$$

Combining this with equation (C.1) we deduce

$$\operatorname{proj}(X; U\mathcal{Q}U^T) = \operatorname{proj}\Big(UU^TXUU^T; U\mathcal{Q}U^T\Big).$$

On the other hand, since the columns of U are orthonormal, for any $Y \in \mathcal{S}^r$ we clearly have

$$\|UU^TXUU^T - UYU^T\| = \|U^TXU - Y\|,$$

and equation (C.2) follows immediately. ∎

COROLLARY C.4 (Distances between faces). *Consider two* $n \times r$ *matrices* U *and* V*, each with orthonormal columns and let* Γ *be the vector of principal angles between* $\operatorname{range} U$ *and* $\operatorname{range} V$*. Then for any* $Z \in \mathcal{S}^r_+$ *the estimate holds:*

$$\operatorname{dist}(VZV^T; U\mathcal{S}^r_+U^T) \leq \sqrt{2} \cdot \|Z\| \cdot \|\sin(\Gamma)\|.$$

Proof. Appealing to Lemma C.3, we obtain the equation $\operatorname{proj}(VZV^T; U\mathcal{S}_+^r U^T) = UU^T(VZV^T)UU^T$. Define now the matrix $\Delta = I - (V^TU)(V^TU)^T$. We successively deduce

$$\begin{aligned}
\operatorname{dist}^2(VZV^T; U\mathcal{S}_+^r U^T) &= \|VZV^T - UU^T(VZV^T)UU^T\|^2 \\
&= \|VZV^T\|^2 - 2\operatorname{tr}(VZV^TUU^TVZV^TUU^T) + \operatorname{tr}(U^TVZV^TUU^TVZV^TU) \\
&= \|Z\|^2 - 2\operatorname{tr}\left(\left(Z(V^TU)(V^TU)^T\right)^2\right) + \operatorname{tr}\left(\left(Z(V^TU)(V^TU)^T\right)^2\right) \\
&= \operatorname{tr}\left(Z^2 - \left(Z(V^TU)(V^TU)^T\right)^2\right) \\
&= \operatorname{tr}\left(Z^2 - \left(Z - Z\Delta\right)^2\right) = \operatorname{tr}\left(2Z^2\Delta - Z\Delta Z\Delta\right) = 2\|\Delta^{\frac{1}{2}}Z\|^2 - \|\Delta^{\frac{1}{2}}Z\Delta^{\frac{1}{2}}\|^2.
\end{aligned}$$

Hence we deduce

$$\begin{aligned}
\operatorname{dist}^2(VZV^T; U\mathcal{S}^rU^T) &= 2\operatorname{tr}(Z^2\Delta) - \|Z^{\frac{1}{2}}\Delta Z^{\frac{1}{2}}\|^2 \leq 2\operatorname{tr}(Z^2\Delta) \leq 2\cdot\|Z\|^2\cdot\|\Lambda\| \\
&= 2\cdot\|Z\|^2\cdot\|\sin^2(\Theta)\| = 2\cdot\|Z\|^2\cdot\|\sin(\Theta)\|^2.
\end{aligned}$$

The result follows. ∎

We are now ready to formally prove robustness guarantees on the method. For simplicity, we will assume that the exposing matrices W_α are of the form UU^T where U have orthonormal columns, and that $\omega_\alpha(d) = 1$ for all cliques α and all $d \in \mathbb{R}^E$. The arguments can be easily adapted to a more general setting. For any subgraph H of G, we let $d[H]$ denote the restriction of d to H. Following [31], the EDM completion problem is said to be *uniquely r-localizable* if either of the equivalent conditions in Observation 2.2 holds. In what follows, let $Alg(d)$ be the output of Algorithm 1 on the EDM completion problem.

THEOREM C.5 (Robustness). *Suppose the following:*

- *for any clique $\alpha \in \Theta$, the subgraph on α has embedding dimension r;*
- *the EDM completion problem is uniquely r-localizable and Alg(d) is the realization of G.*
- *the matrix Y obtained during the run on the noiseless problem has rank $n-r$;*

Then there exist constants $\varepsilon > 0$ and $\kappa > 0$ so that

$$\|\mathcal{P}\circ\mathcal{K}(Alg(\hat{d})) - \hat{d}\| \leq \kappa\|\hat{d} - d\| \text{ whenever } \|\hat{d} - d\| < \varepsilon.$$

Proof. Throughout the proof, we will use the hat superscript to denote the objects (e.g. $\widehat{X}_\alpha$, $\widehat{W}_\alpha$) generated by Algorithm 1 when it is run with the distance measurements $\hat{d} \in \mathbb{R}^E$. Clearly for any $\hat{d} \in \mathbb{R}^E$, we have $\|\mathcal{K}^\dagger\hat{d}_\alpha - \mathcal{K}^\dagger d_\alpha\| = \mathcal{O}(\|d_\alpha - \hat{d}_\alpha\|)$ for any clique $\alpha \in \Theta$. Fix any such clique α, and notice by our assumptions $\mathcal{K}^\dagger d_\alpha$ has rank r. Consequently $\|\widehat{X}_\alpha - X_\alpha\| = \mathcal{O}(\|d_\alpha - \hat{d}_\alpha\|)$ whenever $\hat{d}$ is sufficiently close to d. Appealing then to Corollary C.2, we deduce $\|\widehat{W}_\alpha - W_\alpha\| = \mathcal{O}(\|\widehat{X}_\alpha - X_\alpha\|) = \mathcal{O}(\|d_\alpha - \hat{d}_\alpha\|)$. Hence $\|\widehat{W} - W\| = \mathcal{O}(\|d - \hat{d}\|)$ for all $\hat{d}$ sufficiently close to d. Since W has rank $n-r$, we deduce $\|\widehat{Y} - Y\| = \mathcal{O}(\|d - \hat{d}\|)$. Appealing to Theorem C.1, we then deduce $\|\sin(\Gamma)\| = O(\|d - \hat{d}\|)$, where Γ is the principle angle vector between the null spaces of $\widehat{Y}$ and Y. By Corollary C.4, then

$$\operatorname{dist}\left(X; \operatorname{face}(\widehat{X}, \mathcal{S}_+^n)\right) = \mathcal{O}(\|\hat{d} - d\|).$$

The result follows. □

Acknowledgments. We thank Sasha Aravkin for pointing out a part of the proof of Theorem 4.1.

REFERENCES

[1] 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.

[2] 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.

[3] A.Y. Aravkin, J.V. Burke, and M.P. Friedlander, *Variational properties of value functions*, SIAM J. Optim., 23 (2013), pp. 1689–1717.

[4] S. Aravkin, J. Burke, D. Drusvyatskiy, M.P. Friedlander, and S. Roy, *Optimization over misfit-constrained sets*, Preprint, (2015).

[5] J. Aspnes, T. Eren, D.K. Goldenberg, A.S. Morse, W. Whiteley, Y.R. Yang, B.D.O. Anderson, and P.N. Belhumeur, *Semidefinite programming approaches for sensor network localization with noisy distance measurements*, IEEE Transactions on Automation Science and Engineering, 3 (2006), pp. 360–371.

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

[7] 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.

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

[9] P. Biswas and Y. Ye, *Semidefinite programming for ad hoc wireless sensor network localization*, tech. report, Dept. of Management Science and Engineering, Stanford University, 2003.

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

[11] G. Pataki D. Drusvyatskiy and H. Wolkowicz, *Coordinate shadows of semi-definite and euclidean distance matrices*, To appear in SIAM J. Optim., arXiv:1405.2037, (2015).

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

[13] C. Davis, *The rotation of eigenvectors by a perturbation. II*, J. Math. Anal. Appl., 11 (1965), pp. 20–27.

[14] C. Davis and W. M. Kahan, *Some new bounds on perturbation of subspaces*, Bull. Amer. Math. Soc., 75 (1969), pp. 863–868.

[15] ———, *The rotation of eigenvectors by a perturbation. III*, SIAM J. Numer. Anal., 7 (1970), pp. 1–46.

[16] V.F. Demyanov and A.M. Rubinov, *Approximate methods in optimization problems*, American Elsevier Publishing Co., Inc., New York, 1970.

[17] M. Frank and P. Wolfe, *An algorithm for quadratic programming*, Naval Res. Logist. Quart., 3 (1956), pp. 95–110.

[18] Z. Harchaoui, A. Juditsky, and A. Nemirovski, *Conditional gradient algorithms for norm-regularized smooth convex optimization*, Mathematical Programming, (2014), pp. 1–38.

[19] T.L. Hayden, J. Lee, J. Wells, and P. Tarazaga, *Block matrices and multispherical structure of distance matrices*, Linear Algebra Appl., 247 (1996), pp. 203–216.

[20] T.L. Hayden, J. Wells, W-M. Liu, and P. Tarazaga, *The cone of distance matrices*, Linear Algebra Appl., 144 (1991), pp. 153–169.

[21] M. Jaggi, *Revisiting Frank-Wolfe: Projection-free sparse convex optimization*, in Proceedings of the 30th International Conference on Machine Learning (ICML-13), 2013, pp. 427–435.

[22] N. Krislock, *Semidefinite Facial Reduction for Low-Rank Euclidean Distance Matrix Completion*, PhD thesis, University of Waterloo, 2010.

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

[24] H. Kurata and P. Tarazaga, *Multispherical Euclidean distance matrices*, Linear Algebra Appl., 433 (2010), pp. 534–546.
[25] ———, *Majorization for the eigenvalues of Euclidean distance matrices*, Linear Algebra Appl., 436 (2012), pp. 1473–1481.
[26] 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.
[27] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino, *Euclidean distance geometry and applications*, SIAM Review, 56 (2014), pp. 3–69.
[28] T.K. Pong and P. Tseng, *Robust edge-based semidefinite programming relaxation of sensor network localization*, tech. report, U. of Washington, 2009.
[29] J.B. Saxe, *Embeddability of weighted graphs in k-space is strongly NP-hard*, in Seventeenth Annual Allerton Conference on Communication, Control, and Computing, Proceedings of the Conference held in Monticello, Ill., October 10–12, 1979, Urbana, 1979, University of Illinois Department of Electrical Engineering, pp. xiv+1036. Proceedings of the International School of Physics "Enrico Fermi", LXX*.
[30] A. Singer, *A remark on global positioning from local distances*, Proc. Natl. Acad. Sci. USA, 105 (2008), pp. 9507–9511.
[31] A.M. So and Y. Ye, *Theory of semidefinite programming for sensor network localization*, Math. Program., 109 (2007), pp. 367–384.
[32] A. M-C So and Y. Ye, *Theory of semidefinite programming for sensor network localization*, Math. Program. Ser. B, (2007), pp. 367–384.
[33] P. Tarazaga, *Faces of the cone of Euclidean distance matrices: characterizations, structure and induced geometry*, Linear Algebra Appl., 408 (2005), pp. 1–13.
[34] P. Tarazaga and J.E. Gallardo, *Euclidean distance matrices: new characterization and boundary properties*, Linear Multilinear Algebra, 57 (2009), pp. 651–658.
[35] P. Tarazaga, T.L. Hayden, and J. Wells, *Circum-Euclidean distance matrices and faces*, Linear Algebra Appl., 232 (1996), pp. 77–96.
[36] P. Tarazaga, B. Sterba-Boatwright, and K. Wijewardena, *Euclidean distance matrices: special subsets, systems of coordinates and multibalanced matrices*, Comput. Appl. Math., 26 (2007), pp. 415–438.
[37] E. van den Berg and M.P. Friedlander, *SPGL1: A solver for large-scale sparse reconstruction*, http://www.cs.ubc.ca/labs/scl/spgl1, (2007).
[38] E. van den Berg and M.P. Friedlander, *Probing the Pareto frontier for basis pursuit solutions*, SIAM J. Sci. Comput., 31 (2008/09), pp. 890–912.
[39] ———, *Sparse optimization with least-squares constraints*, SIAM J. Optim., 21 (2011), pp. 1201–1229.
[40] Z. Wang, S. Zheng, Y. Ye, and S. Boyd, *Further relaxations of the semidefinite programming approach to sensor network localization*, SIAM J. Optim., (2008), pp. 655–673.
[41] 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.
[42] 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.