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 noisy 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 $V = \{1, \dots, n\}$ with edge set E and squared distances $d \in \mathbb{R}^E$, and an integer r, find (if possible) a set of points $x_1, \dots, x_n$ in $\mathbb{R}^r$ satisfying

$$\|x_i - x_j\|^2 = d_{ij}, \quad \text{for all edges } ij \in E, \tag{1.1}$$

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 [3,15,36], and is broadly applicable for example in wireless networks, statistics, robotics, protein reconstruction, and dimensionality reduction in data analysis; the recent survey [37] 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

$$\left\{ \begin{array}{c} X_{ii} + X_{jj} - 2X_{ij} = d_{ij}, \quad \forall\, ij \in E \\ Xe = 0 \\ X \succeq 0, \end{array} \right\} \tag{1.2a}$$

$$\operatorname{rank} X \leq r, \tag{1.2b}$$

*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. www.math.uwaterloo.ca/~hwolkowi/

where e stands for the vector of all ones. Indeed, if $X = PP^T$ is a maximal rank factorization of a matrix X solving (1.2), then the rows of P yield a solution to the EDM 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 (1.2b). The resulting problem (1.2a) is convex (in fact, a semidefinite program (SDP)) and thus is 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. Consequently, SDP techniques have proven to be extremely useful for this problem; notable references include [7, 8, 10, 11, 32, 38, 41, 44, 54]. 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 (1.2a) if the structure is not exploited—a key theme of this paper. For example, one can show 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 this current work, we attempt to close the gap between what can be done with exact data and what can be done in practice with noisy data for large scale problems. We do this for the EDM completion problem with two approaches: a combinatorial algorithm and an efficient first-order method. 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 (1.2a). This observation immediately motivated the algorithm of [32]. This *primal procedure* proceeds by collecting a large number of cliques in the graph and intersecting the corresponding faces two at a time, 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 *dual procedure* complementary to [32] for the problem (1.2a) 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 [32], the new method is conceptually easy to parallelize. This is discussed briefly in Section 3.2.1 on clique and exposing vector selection.

In the late stages of writing the current paper, we became aware of the related work [43]. There the author proposes the Locally Rigid Embedding (LRE) method for the EDM completion problem. The LRE method is in the same spirit as our algorithm, but is stated in the language of rigidity theory; see also [2, 14, 24, 35]. As a byproduct, our current work yields an interpretation of the LRE algorithm in terms of SDP facial reduction. Moreover, in contrast to [43], we formally justify the robustness of our proposed method and perform extensive numerical tests, justifying its guarantees. We also discuss important implementation issues, not covered in [43]. The LRE method requires the local neighborhood around each node be rigid—any missing distances in each neighborhood are found by solving an SDP. In our proposed method, we find a clique around each node, thus avoiding introducing errors from completing locally missing distances. In addition, we form the final exposing vector as a *simple weighted sum*, in contrast to [43] where all weights are equal, and then

solve a resulting small least squares problem for the final Gram matrix that yields the final positions of nodes. We observe that the appearance of weights can greatly help when the noise is unevenly distributed across the distance measurements.

The clique-based algorithm discussed above is most powerful when the graph G is fairly dense. In contrast, in the second part of the paper, we propose a first-order method for solving the noisy EDM completion problem that is most powerful when the graph G is sparse. We consider the following *max-trace heuristic*—a popular low-rank inducing regularizer [5, 55]—given in the formulation of the problem:

$$\begin{aligned} \max \quad & \operatorname{tr} X \\ \text{s.t.} \quad & \sum_{ij\in E} |X_{ii} + X_{jj} - 2X_{ij} - d_{ij}|^2 \le \sigma \\ & Xe = 0 \\ & X \succeq 0. \end{aligned} \tag{1.3}$$

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 (1.3) instead of perhaps the more usual regularized problem

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

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 (1.3) notoriously fails strict feasibility. In particular, for small $\sigma \ge 0$ 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 difficulties. Keeping this in mind, we propose an *infeasible* first-order method, which is not directly affected by the poor conditioning of the underlying problem.

To this end, consider the following parametric *flipped problem*, obtained by "flipping" the objective and the quadratic constraint in (1.3):

$$\begin{aligned} v(\tau) := \quad \min \quad & \sum_{ij\in E} |X_{ii} + X_{jj} - 2X_{ij} - d_{ij}|^2 \\ \text{s.t.} \quad & \operatorname{tr} X = \tau \\ & Xe = 0 \\ & X \succeq 0. \end{aligned} \tag{1.4}$$

Notice that the problem of evaluating $v(\tau)$ is readily amenable to first-order methods, in direct contrast to (1.3). 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 *structurally* 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 (1.3) amounts to finding the largest value of τ satisfying $v(\tau) \leq \sigma$, a problem that can be solved by an approximate Newton method. A discussion of this general strategy can be found in [4], and originates in [6,50–52]. 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. Comparisons are made with both the *primal procedure* in [32] and the more recent edge-based approach in [41]. Our results significantly improve on those in both papers. Section 4 describes the proposed Pareto search technique with Frank-Wolfe iterations, and presents numerical experiments on sparse graphs.

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 matrices (PSD)* 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 $\operatorname{face}(\Omega; \mathcal{S}^n_+)$ 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 a positive integer $r \geq 1$ 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 positive weights.[1] 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.

[1] We assume that the weights are sufficiently positive so they can be numerically distinguished from zero.

The *EDM completion problem* is to find a set of points $x_1, \ldots, x_n \in \mathbb{R}^r$ satisfying (1.1). 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 points $x_1, \ldots, x_n$ can always be translated so that they are centered around the origin, meaning $\sum_i x_i = 0$. This problem is equivalent to finding a matrix $X \in \mathcal{S}^n$ satisfying the nonconvex system (1.2). Given the realizing points x_i and defining $P = [x_1; \ldots; x_n]^T \in \mathbb{R}^{n \times r}$, the matrix $X := PP^T$, called the *Gram matrix*, is feasible for (1.2). For more details, see, e.g., [32].

As mentioned above, the EDM completion problem (1.2) is nonconvex and is NP-hard in general, [42, 56]. A convex relaxation is obtained simply by ignoring the rank constraint in (1.2) yielding the convex SDP feasibility problem (1.2a). For many EDM completion problems on fairly dense graphs, this convex relaxation is "exact" [44]. For example, the following is immediate.

OBSERVATION 2.1 (Exactness of the relaxation). *If the EDM completion problem* (1.2) *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* (1.2a) *has rank at most r and consequently any solution of* (1.2a) *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 (1.2a) 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 (1.2a) admits a unique solution [44]. 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* (1.2) *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* (1.2a) *has a unique solution.*

Proof. The implication $2 \Rightarrow 1$ is immediate. To see the converse implication $1 \Rightarrow 2$, suppose that the SDP (1.2a) 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 (1.2a). 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 the formulation (1.2). 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 $\mathcal{K}^*\colon \mathcal{S}^n \to \mathcal{S}^n$ is given by

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

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, 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 $e^\perp$.[2] These and other related constructions have appeared in a number of publications; see, e.g., [1, 26, 27, 33, 34, 45–48].

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

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

Define the coordinate projection $\mathcal{P}\colon \mathcal{S}^n \to \mathbb{R}^E$ by setting $\mathcal{P}(X)_{ij} = X_{ij}$, $\forall ij \in E$. In this notation, the feasible set (1.2) can equivalently be written as

$$\{X \in \mathcal{S}^{n,r}_{c,+} : \mathcal{P} \circ \mathcal{K}(X) = d\},$$

while the relaxation (1.2a) 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

$$\mathcal{S}^n_{c,+} = U \mathcal{S}^{n-1}_+ U. \tag{2.2}$$

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 (2.2) 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 (1.2):

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

[2] This definition of the Moore-Penrose pseudo-inverse easily extends to general $D \in \mathcal{S}^n$ by first orthogonally projectiing D onto the space of symmetric matrices with zero diagonal.

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

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

Before describing our algorithmic framework for tackling (3.1), it is instructive to put it into context. The authors of [32] found a way to use the degeneracy of the system (1.2) explicitly to design a *combinatorial algorithm* for solving (1.2), 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 (1.2a) 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 (1.2a) to obtain an equivalent problem involving fewer variables. On a basic level, their method explores cliques in the graph and intersects pairwise such faces in a computationally efficient way. The algorithm is surprisingly fast and accurate for huge problems where the noise is *small*.

An important computational caveat of the (primal) facial reduction algorithm of [32] is that the algorithm is highly unstable when the distance measurements are corrupted by *significantly large* 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 error dramatically accumulates as the algorithm iteratively intersects faces. 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. We see this in the comparison of the primal facial reduction algorithm and the dual exposed vector approach in Section 3.3.1, page 10, below. Further details are provided in Appendix C.

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 $i = 1, \ldots, m$), 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. 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 [32]. The dual form, however, is essential for our purposes. For easy similar alternative proofs, see [20, Theorem 4.9] and [31, 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 (Single clique facial reduction, [20, Theorem 4.13]). *Suppose that the subset of vertices $\alpha := \{1, \ldots, k\} \subset V$ is a clique in G. Define the set*

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

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

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

In particular, under the assumptions of the theorem, the entire feasible region of (1.2a) 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 state our proposed algorithmic framework in Algorithm 1.

Algorithm 1 Basic strategy for EDM completion

INPUT: A weighted graph $G = (V, E, d)$, and a target rank $r \geq 1$;
PREPROCESSING:
1. generate a set of cliques Θ in G;
2. generate a set of positive weights $\{\omega_\alpha\}_{\alpha \in \Theta} \subset \mathbb{R}_{++}$;
3. sort the cliques so the weights are in nondecreasing order;

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 $\mathcal{S}^n_c$ by adding zeros;
end for
$W \leftarrow \sum_{\alpha \in \Theta} \omega_\alpha W_\alpha$;
Let $U \in \mathbb{R}^{n \times r}$ be a matrix satisfying $U^T e = 0$, and whose columns are eigenvectors of W corresponding to r smallest nonzero eigenvalues;
$X \leftarrow UZU^T$, where Z is an optimal solution of

$$val_{lss} := \quad \min \quad \|\mathcal{P} \circ \mathcal{K}(UZU^T) - d\| \quad \text{s.t.} \quad Z \in \mathcal{S}^r_+; \tag{3.2}$$

return X;

Finding the eigenvectors in U is equivalent to finding a *nearest matrix* to $\mathcal{S}^{k,r}_{c,+}$ or to $\mathcal{S}^{n,n-r}_{c,+}$. This is easy as a result of the classical Eckart-Young Theorem [21]. It requires finding the $r+1$ smallest eigenvalues of W where we know the smallest is 0 with eigenvector e, the ones vector. Therefore, we can shift the eigenvalue 0 to a sufficiently positive eigenvalue and use *eigs* in MATLAB to find the eigenvectors U corresponding to the smallest r eigenvalues; due to the orthogonality of eigenvectors of distinct eigenvalues of a symmetric matrix, we have $U^T e = 0$, as desired. The details are worked out in Appendix A on page 24. Solving the small dimensional least squares problem (3.2) is also standard, though we include the use of a *sketch* matrix for this highly overdetermined constrained least squares problem, e.g., [39];

details are presented in Appendix B on page 26. In fact, very often (under the assumptions of Theorem C.5 on page 29) the optimal 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 conic optimization solver to be invoked. Indeed, this is a direct consequence of the rudimentary robustness guarantees of the method, outlined in Appendix C. To see intuitively why the outlined method is robust, observe that when adding up the "rounded" exposing vectors, the error does not propagate; i.e. the error obtained by guessing at an exposing vector coming from one clique does not influence the error in the exposing vector arising from another clique, even if the two overlap.

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 set of weights $\{\omega_\alpha\}_{\alpha\in\Theta}$ (in Section 3.2.1);
- the nearest-point mapping to $\mathcal{S}^{k,r}_{c,+}$ (in Appendix A);
- 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. This emphasizes that our algorithm provides a good initialization for nonlinear solvers.

3.2.1 Choosing clique sets and exposing vectors. We first discuss the choice of the clique set Θ that we use for finding the exposing vector Y. We have combined our approach with the primal facial reduction algorithm and code in [32]. Each time this code uses a **new** clique, we add it to the set Θ and find a corresponding exposing vector and a weight. We emphasize that this step can be parallelized since the work on the cliques are independent even if the cliques intersect.

Now we discuss the set of positive weights $\{\omega_\alpha\}_{\alpha\in\Theta}$. 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 squared distance of $\mathcal{K}^\dagger d_\alpha$ to the rank r PSD matrices:

$$\nu_\alpha(d) := \frac{\sum_{j=1}^{|\alpha|-r}\lambda_j^2(\mathcal{K}^\dagger d_\alpha)+\sum_{j=|\alpha|-r+1}^{|\alpha|}\left(\min\{0,\lambda_j(\mathcal{K}^\dagger d_\alpha)\}\right)^2}{0.5|\alpha|(|\alpha|-1)}. \tag{3.3}$$

Here $\lambda_j(\mathcal{K}^\dagger d_\alpha)$ refers to the j'th smallest eigenvalue of the matrix $\mathcal{K}^\dagger d_\alpha$.

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 weights reflects the contribution of noise in the clique α to the total noise of all cliques. If a clique α is relatively noisy compared to other cliques in Θ

or contains an outlier, the weight $\omega_\alpha(d)$ would be significantly smaller than $\omega_\beta(d)$ for most $\alpha \neq \beta \in \Theta$. In addition, we add up the weighted exposing vectors in order from the smallest weight to the largest weight for improved numerical accuracy.

3.2.2 Postprocessing: local refinement. Following Algorithm 1, we apply a *local refinement* method to 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 [7, 10].

For local refinement, we use the steepest descent subroutine from the SNL-SDP package [8]. Suppose that $X_{\approx} = P_{\approx}(P_{\approx})^T$ is the solution of (3.2) found at the end of Algorithm 1. We use $P_{\approx}$ 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 typically produces excellent initial points for such nonlinear optimization schemes.

3.3 Application on the sensor network localization problem. In this section, we apply the robust facial reduction in Algorithm 1 on the *sensor network localization, SNL* , problem in $\mathbb{R}^2$. The task is to locate n wireless sensors in $\mathbb{R}^2$ given the noisy Euclidean distances between sensors that are within a given *radio range, R*, of each other. Often some of the sensors are *anchors* whose positions are known. Semidefinite programming techniques have been used extensively for the SNL problem; see, e.g., [7, 8, 10, 11, 32, 41, 44, 54].

We look at the quality of the approximation of our data d, the resulting approximate EDM, and the error in the found sensor positions P.

3.3.1 Preliminary look at quality of solutions. Suppose that we are given an anchor-free instance with data $d + \delta d$, where δd is the noise. We consider our algorithm as a *black box* transformation that takes $d + \delta d$ as input and outputs: approximate distance matrix $D_{\approx} \in \mathbb{R}^{n\times n}$, its restriction to the graph $d_{\approx} \in \mathbb{R}^E$, and approximate localized sensors $P_{\approx} \in \mathbb{R}^{n\times r}$. We now see empirically that the relative error and corresponding condition number estimates for this transformation are, surprisingly, similar for all three outputs and that the exposed vector approach is consistently better. The condition number estimates show empirically that our algorithm is *backwards stable*. We provide more details and justification in Appendix B.

1. **Quality of Solution in d and D:** The best approximation we find is

$$d_{\approx} := \mathcal{P} \circ \mathcal{K}(UBU^T) \approx (d + \delta d).$$

The *relative residual error in d* for our found approximate distances is given by

$$[\text{rel. error}]_d := \frac{\|d_{\approx} - d\|}{\|d\|}.$$

The derivative type *condition number for d* of the algorithm can then be estimated from below by using the ratios of the relative errors

$$[\text{condition number}]_d \approx \left(\frac{\|d_{\approx} - d\|}{\|d\|}\right) \Big/ \left(\frac{\|\delta d\|}{\|d\|}\right) = \frac{\|d_{\approx} - d\|}{\|\delta d\|},$$

see, e.g., [23, Sect. 2.5].
The (nonlinear) equation we next look at is the solution of the distance matrix D found from the given data d with noise δd. The corresponding *relative residual error in* D is

$$D_{\approx} := D + \delta D = \mathrm{Alg}(d + \delta d), \quad [\text{rel. error}]_D := \frac{\|\delta D\|}{\|D\|}.$$

We note that the distances $(D+\delta D)_{ij}$, $ij \notin E$ have no *direct* constraints on them, just implicit constraints from nearby cliques.
As discussed above, the condition number of the algorithm for finding $D_{\approx}$ can then be estimated from below by using the ratios of the relative errors

$$[\text{condition number}]_D \approx \left(\frac{\|\delta D\|}{\|D\|}\right) \Big/ \left(\frac{\|\delta d\|}{\|d\|}\right).$$

Figure 1 presents the relative error and condition number estimates for 40 random anchor-free instances with increasing noise factors; these instances use the primal facial reduction algorithm in [32], the exposed vector algorithm (Algorithm 1), and the amalgam that takes the output of primal facial reduction algorithm X and projects it onto $W^{\perp} \cap \mathcal{S}^{k,r}_{c,+}$, where W is the aggregate exposing vector in Algorithm 1.[3] The consistent improvement for Algorithm 1 is clear. The exposed vector approach advantage is further emphasized by noting the improvement over the amalgam strategy. The condition numbers lower estimates for the exposed vector approach are surprisingly low for both finding d and the complete EDM D.

2. **Quality of Solution with** $RMSD$ **estimates:** Though the objective function involves only an approximation of d, the end objective of the algorithm is a good localization for the matrix of points P. We measure the error using the *root-mean-square deviation, RMSD* values. 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 on page 8. Then we may factor $X = \tilde{P}\tilde{P}^T$ for some matrix $\tilde{P} \in \mathbb{R}^{n\times 2}$, whose rows store the estimated centered locations. Before comparing $\tilde{P}$ and P, however, we must allow isometries to act on the points. As a result, the RMSD of the estimated $\tilde{P}$ relative to the true centered locations P is defined as:

$$RMSD := \min\left\{\frac{1}{\sqrt{n}}\|\tilde{P}U - P\|_F \;:\; U^TU = I,\; U \in \mathbb{R}^{r\times r}\right\}. \tag{3.5}$$

Computing RMSD is an instance of the orthogonal procrustes problem, which can be solved using the Algorithm in [23]. The condition number for finding $X_{\approx}$ can then be estimated from below by using the ratios of the relative errors

$$[\text{condition number}]_{RMSD} \approx (RMSD) \Big/ \left(\frac{\|\delta d\|}{\|d\|}\right).$$

The results for the same 40 instances as above are given in Figure 2.

REMARK 3.2. Though the exposed vector approach is clearly significantly better than the primal approach in [32], we see above that the latter still has consistently better results than those reported in [32]. This is because the RMSD values found in [32] used only the anchor positions in the Procrustes problem whereas here we have anchor-free problems and use all the known original sensor positions.

[3] Details on how the instances are generated is given below in Algorithm 2, page 13.

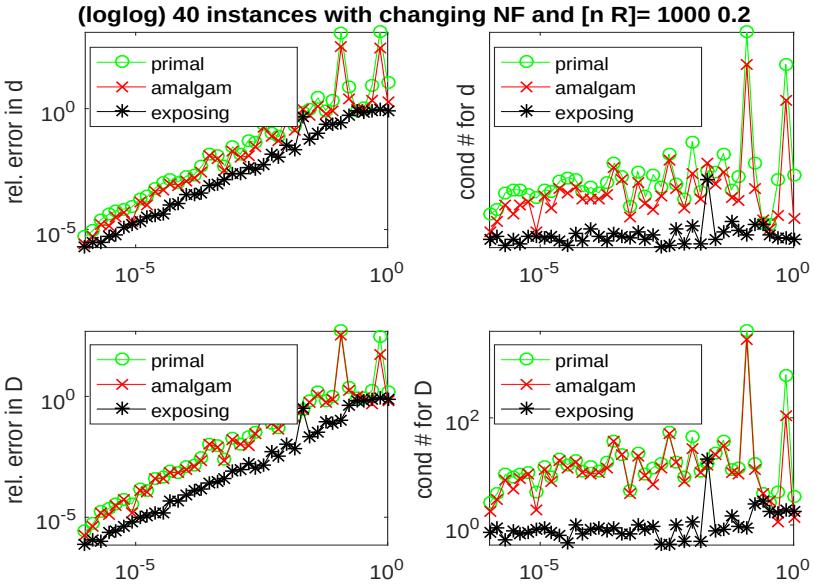


Fig. 1: Primal facial reduction [32], dual exposed vector (Algorithm 1), and the amalgam approach that takes the output of primal facial reduction algorithm X and projects it onto $W^{\perp} \cap \mathcal{S}_{c,+}^{k,r}$, where W is the aggregate exposing vector in Algorithm 1; 40 equally spaced instances with nf $\in [10^{-6}, 1]$

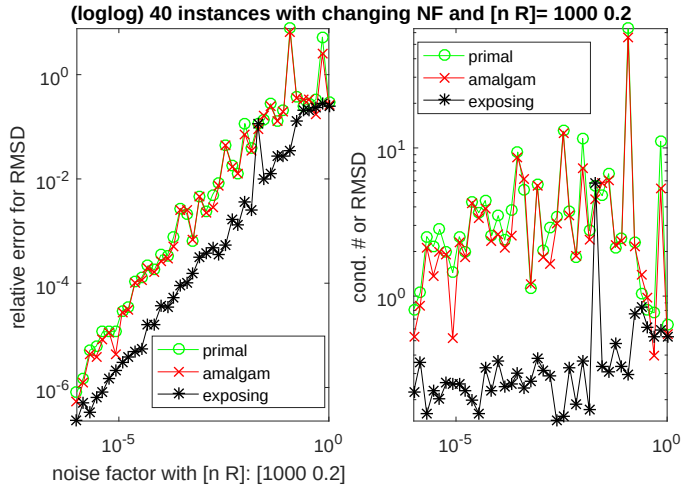


Fig. 2: Primal facial reduction [32], dual exposed vector (Algorithm 1), and the amalgam approach that takes the output of primal facial reduction algorithm X and projects it onto $W^{\perp} \cap \mathcal{S}_{c,+}^{k,r}$, where W is the aggregate exposing vector in Algorithm 1; 40 equally spaced instances with nf $\in [10^{-6}, 1]$; $R = .3$; $n = 1000$ nodes

3.3.2 Numerics. We now provide comprehensive numerical tests. We generate random instances of the SNL problem based on a modified *multiplicative noise*

model from, e.g., [7,8,41]. This multiplicative noise model is the one most commonly considered in sensor network localization, e.g., [9,12,29,30,40,49,53]. The perturbations ϵ_{ij} are from the standard normal distribution $\epsilon_{ij} \in \mathcal{N}(0,1), \forall ij$. We truncate the tails of the distribution to ensure that the multiplicative perturbation is nonnegative $0 \leq 1 + \sigma\epsilon_{ij}$. We outline the details in Algorithm 2.

Algorithm 2 Multiplicative noise model

INPUT: # sensors $.9n$, # anchors $.1n$, noise factor σ, radio range R, machine ϵ;
Generate n centered nodes:

1. pick n uniformly i.i.d. nodes, $p_i \in [-0.5, 0.5]^2 \subset \mathbb{R}^2$;
2. center nodes around the origin: $\bar{p} = \frac{1}{n}\sum_{i=1}^{n} p_i$ and $p_i \leftarrow p_i - \bar{p}, \forall i$
3. ensure that the final distances are significantly positive:

$$p_i \leftarrow (1 + \max\{2\sigma, 100\epsilon\})p_i;$$

and center the nodes around the origin again;

Perturb the known sensor-sensor distances:

1. pick i.i.d. $\epsilon_{ij} \in \mathcal{N}(0,1), \forall ij$ (standard normal distribution);
2. randomly truncate the tails to ensure nonnegativity of the distances and error mean 0;
3. compute $D \in \mathcal{S}^n$ by

$$D_{ij} = \begin{cases} (1+\sigma\epsilon_{ij})^2\|p_i - p_j\|^2, & \text{if } \min\{i,j\} \leq .9n \\ \|p_i - p_j\|^2, & \text{otherwise} \end{cases}$$

Construct the graph:

1. graph $G \leftarrow (V, E, d)$, with

$$V = \{1, \ldots, n\}, \qquad E = \{ij : \|p_i - p_j\| < R \text{ or } \min\{i,j\} > .9n\}.$$

2. vector $d \leftarrow (D_{ij})_{ij \in E,\, i<j} \in \mathbb{R}^E$.

OUTPUT: noisy distance measurements $d \in \mathbb{R}^E$ and graph G.

We gauge the performance of the robust facial reduction on random instances from the multiplicative noise model using first the relative residual values for the perturbed given data and second the *root-mean-square deviation, RMSD*, values defined in (3.5). As would be expected, the RMSD values can be very large when R is small and we may have nonunique solutions. However, we see that the much more revealing residual values still stay small.

Above, Section 3.3.1 shows that the quality of the solutions from the algorithm using the exposed vectors significantly and consistently improves on the primal clique approach in [32] for anchorless problems with $n = 1000$ and increasing noise factor. Table 1 on page 14 further shows the details with times, residuals, and RMSD values before and after refinement. We can see that when we have the small $R = .10$ and resulting low density for the graph, then the residual value does not increase substantially but the RMSD value indicates that the location of the points found are not reasonable. Table 2 on page 15 presents similar results but in addition with increasing numbers of nodes. For comparison, Table 3 on page 15 presents results using the code from [41]. We see the dramatic improvement both in time and also

Table 1: Robust facial reduction; small instances generated using multiplicative noise model on a $[-0.5, 0.5]^2$ grid; n number of sensors/vertices; $m = 0$ anchors; R radio range. CPU seconds, percent residuals, and percent RMSD for: 1. the exposed facial algorithm and 2. the refinement.

Specifications				Time (s)		Residual (%R)		RMSD (%R)	
n	% noise	R	% dens.	initial	refine	initial	refine	initial	refine
1000	0	0.25	15.7	0.7	0.0	0.0	0.0	0.0	0.0
1000	10	0.25	15.7	0.6	1.5	4.5	2.8	17.5	1.2
1000	20	0.25	15.7	0.6	1.7	8.7	5.6	38.2	2.4
1000	30	0.25	15.7	0.6	2.6	12.4	8.5	56.5	3.6
1000	40	0.25	15.7	0.6	2.9	15.3	11.1	67.6	17.1
1000	5	0.30	21.6	0.8	1.1	2.2	1.6	4.1	0.5
1000	5	0.25	15.7	0.6	1.4	2.1	1.4	6.4	0.6
1000	5	0.20	10.6	0.4	0.7	1.9	1.1	9.5	0.7
1000	5	0.15	6.2	0.3	1.0	2.9	0.8	39.8	1.4
1000	5	0.10	2.9	1.0	0.6	5.2	2.1	311.7	286.3

in the RMSD values. Table 4 on page 16 presents numerical results using instances with $n = 1000$ to $15,000$ nodes and with varying radio range and noise factor. We include 10% anchors so as to be able to compare to the results in [41]. As mentioned above our results are significantly better than those in [41]. We solve larger problems in a fraction of the time and to low RMSD. The tests were run on MATLAB version R2016a, on a Dell Optiplex 9020, with Windows 7, Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz and 16 GB RAM. We show the RMSD (as a percentage of the radio range) of the solutions provided by Algorithm 1 and also the RMSD of the solution after the local refinement using the steepest descent subroutine from SNL-SDP. We see that using Algorithm 1 together with local refinement gives excellent results. The time used by Algorithm 1 includes the selection of cliques and computation of the exposing vectors.

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 (1.2a). To this end, we rewrite the *flipped problem* in (1.4).

$$\begin{array}{ll} \max & \operatorname{tr} X \\ \text{s.t.} & \|\mathcal{P} \circ \mathcal{K}(X) - d\| \leq \sigma \\ & Xe = 0 \\ & X \succeq 0. \end{array} \tag{4.1}$$

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

Table 2: Robust facial reduction (Algorithm 1); large instances generated using multiplicative noise model on a $[-0.5, 0.5]^2$ grid; n number of sensors/vertices; $m = 0$ anchors; R radio range. CPU seconds, percent residuals, and percent RMSD for: a. the exposed facial algorithm and b. the refinement.

Specifications				Time (s)		Residual (%R)		RMSD (%R)	
n	% noise	R	% dens.	initial	refine	initial	refine	initial	refine
2000	0	0.20	10.5	2.4	0.1	0.0	0.0	0.0	0.0
2000	10	0.20	10.5	2.0	4.8	3.5	2.2	13.3	1.0
2000	20	0.20	10.5	1.7	5.2	7.1	4.5	34.1	2.0
2000	30	0.20	10.5	1.7	5.3	10.1	6.8	55.8	3.1
3000	0	0.18	8.7	4.2	0.2	0.0	0.0	0.0	0.0
3000	10	0.18	8.7	3.6	13.2	3.5	2.0	16.6	0.9
3000	20	0.18	8.7	3.1	14.2	7.0	4.1	50.3	1.8
3000	30	0.18	8.7	2.9	19.7	9.7	6.2	77.9	2.7
4000	0	0.16	7.0	6.0	0.3	0.0	0.0	0.0	0.0
4000	10	0.16	7.0	5.5	22.5	3.3	1.8	19.0	0.9
4000	20	0.16	7.0	4.4	25.0	6.2	3.6	45.9	1.8
4000	30	0.16	7.0	4.2	26.9	8.5	5.5	74.9	2.6
6000	5	0.14	5.4	11.0	58.2	1.5	0.8	11.0	0.4
8000	5	0.12	4.1	15.6	100.2	1.6	0.7	16.5	0.4
10000	5	0.10	2.9	21.7	145.0	1.5	0.6	22.3	0.4
15000	5	0.10	2.9	36.0	161.2	1.5	0.6	21.8	0.4
20000	5	0.10	2.9	53.9	546.7	1.4	0.6	19.6	0.3

Table 3: Edge based method from [41]; moderate instances generated using multiplicative noise model on a $[-0.5, 0.5]^2$ grid; n number of sensors/vertices; $m = .1n$ anchors; R radio range. CPU seconds, and percent RMSD for: a. the edge based method from [41] and b. the refinement.

Specifications				Time (s)		RMSD (%R)	
n	% noise	R	% dens.	initial	refine	initial	refine
2000	0	0.20	11.4	31.0	0.0	0.0	0.0
2000	10	0.20	11.4	146.6	0.3	8.7	1.3
2000	20	0.20	11.4	305.7	0.3	17.9	2.7
2000	30	0.20	11.4	351.8	0.3	26.7	4.0

have a lower rank. This is in contrast to the usual min-trace strategy in compressed sensing; see [6, 51, 52] for a discussion. Indeed, as was mentioned in the introduction in terms of the factorization $X = PP^T$, the equality $\mathrm{tr}(X) = \frac{1}{2n}\sum_{i,j=1}^{n} \|p_i - p_j\|^2$ 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

Table 4: Robust facial reduction (Algorithm 1); large instances with anchors generated using multiplicative noise model on a $[-0.5, 0.5]^2$ grid; n number of sensors/vertices; $m = .1n$ number of anchors; R radio range. CPU seconds, percent residuals, and percent RMSD for: a. the exposed facial algorithm and b. the refinement.

Specifications				Time (s)		Residual (%R)		RMSD (%R)	
n	% noise	R	% dens.	initial	refine	initial	refine	initial	refine
2000	0	0.20	11.4	1.3	0.1	0.0	0.0	0.0	0.0
2000	10	0.20	11.4	1.3	1.8	2.3	2.1	3.9	1.0
2000	20	0.20	11.4	1.3	1.9	4.7	4.3	8.1	2.0
2000	30	0.20	11.4	1.3	1.8	7.2	6.5	12.5	3.0
3000	0	0.18	9.6	2.4	0.2	0.0	0.0	0.0	0.0
3000	10	0.18	9.6	2.4	3.4	2.1	1.9	3.7	0.9
3000	20	0.18	9.6	2.4	3.6	4.2	3.9	7.6	1.8
3000	30	0.18	9.6	2.4	3.8	6.4	5.9	11.6	2.7
4000	0	0.16	7.9	3.7	0.3	0.0	0.0	0.0	0.0
4000	10	0.16	7.9	3.7	5.9	1.8	1.7	3.6	0.9
4000	20	0.16	7.9	3.7	5.8	3.7	3.4	7.3	1.7
4000	30	0.16	7.9	3.7	6.1	5.6	5.2	11.2	2.6
6000	5	0.14	6.4	7.1	9.7	0.8	0.7	1.7	0.4
8000	5	0.12	5.0	10.4	12.5	0.6	0.6	1.7	0.4
10000	5	0.10	3.9	12.9	13.7	0.5	0.5	1.8	0.4
15000	5	0.10	3.8	31.7	20.1	0.5	0.5	1.6	0.3
20000	5	0.10	3.8	59.6	44.2	0.5	0.5	1.5	0.3

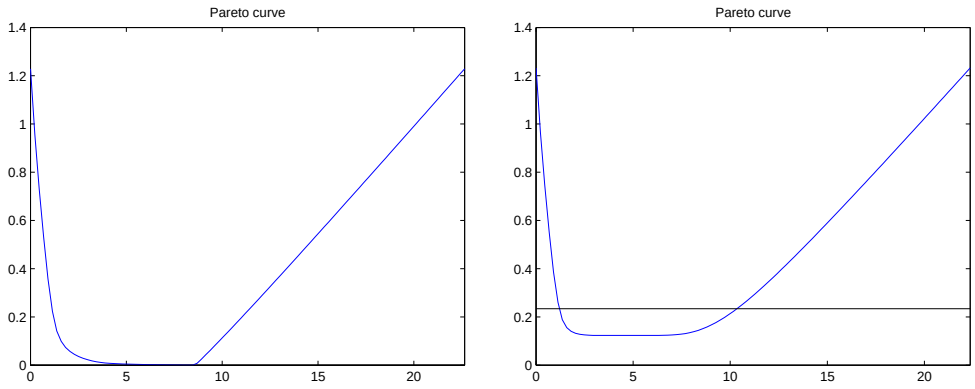
analogous analysis holds for min-trace. At the end of the section we compare the two.

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; see e.g., [50–52] or the survey [4]. 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

$$\begin{aligned} \varphi(\tau) := \quad & \min \quad && \|\mathcal{P} \circ \mathcal{K}(X) - d\| \\ & \text{s.t.} \quad && \operatorname{tr} X = \tau \\ & && Xe = 0 \\ & && X \succeq 0. \end{aligned} \tag{4.2}$$

See Figure 3 below for an illustration.

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



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

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

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 $v\colon I \to \mathbb{R}$ 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 3.

Algorithm 3 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**
  $t_{k+1} \leftarrow t_k - \frac{l_k}{s_k}$;
  $(l_{k+1}, u_{k+1}, s_{k+1}) := \mathcal{O}_v(t_{k+1}, \alpha)$;
  $k \leftarrow k + 1$;
end while
return t_k;

It can be shown that the iterates t_k generated by the inexact Newton method (Algorithm 3), 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

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

iterations, where we set $R = \max\{|s_0|(\bar{t} - t_0), l_0\}$. For a proof and a discussion, see [4, Theorem 2.4].

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* [22], which has recently found many applications in machine learning (see, e.g., [25, 28]). 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 4.

The computational burden of the method is the minimization problem (4.3) in Algorithm 4. 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 τvv^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 4 is an affine minorant oracle for the function $v(\tau) := \varphi(\tau) - \sigma$.*

Algorithm 4 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);

Proof. We first claim that upon termination of Algorithm 4, 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

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

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

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

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 4, 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

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

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 $\frac{u}{l} \leq \alpha$ holds. To see this, observe that

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

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

$$\begin{aligned}\Big((u+\alpha\sigma)^2 + \|\alpha y\|^2 - 2\alpha\sigma\|\alpha y\|\Big) - 2\|\alpha y\|u &= (u+\alpha\sigma)^2 + \|\alpha y\|^2 - 2\|\alpha y\|(u+\alpha\sigma) \\ &= (u + \alpha\sigma - \|\alpha y\|)^2 \geq 0.\end{aligned}$$

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

Thus Algorithm 4 is an affine minorant oracle for $\varphi - \sigma$, and linear convergence guarantees of the inexact Newton method (Algorithm 3) 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 4 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., [19, 22, 28]), 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 $\max\left\{1 + \log_{2/\alpha}\left(2R/\beta\right), 2\right\}$ inexact Newton iterations[4], 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]As before $|s_0|$ is the slope of the value function v at τ_0 and $R = \tau_0 - \mathrm{OPT}$.

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

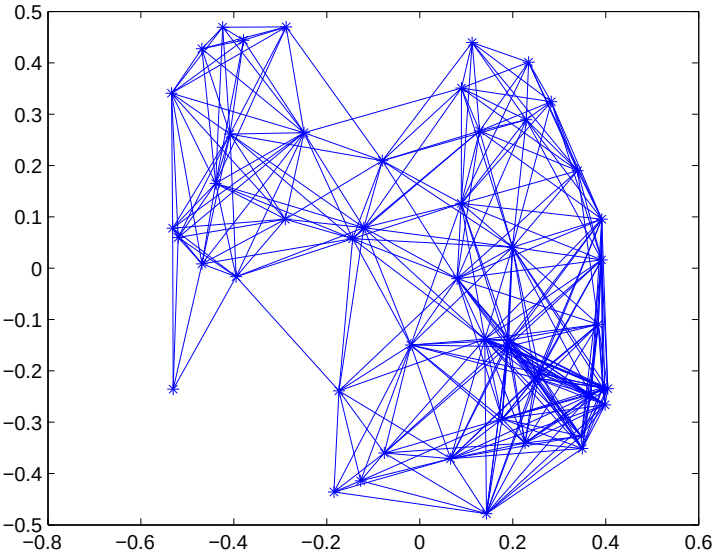
$$\begin{aligned} \min \quad & \operatorname{tr} X \\ \text{s.t.} \quad & \|\mathcal{P} \circ \mathcal{K}(X) - d\| \leq \sigma, \quad Xe = 0, \quad X \succeq 0, \end{aligned}$$

and the maximal trace solution

$$\begin{aligned} \max \quad & \operatorname{tr} X \\ \text{s.t.} \quad & \|\mathcal{P} \circ \mathcal{K}(X) - d\| \leq \sigma, \quad Xe = 0, \quad X \succeq 0. \end{aligned}$$

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.

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



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.

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.

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. 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 $\sigma + \beta = 0.3341$.

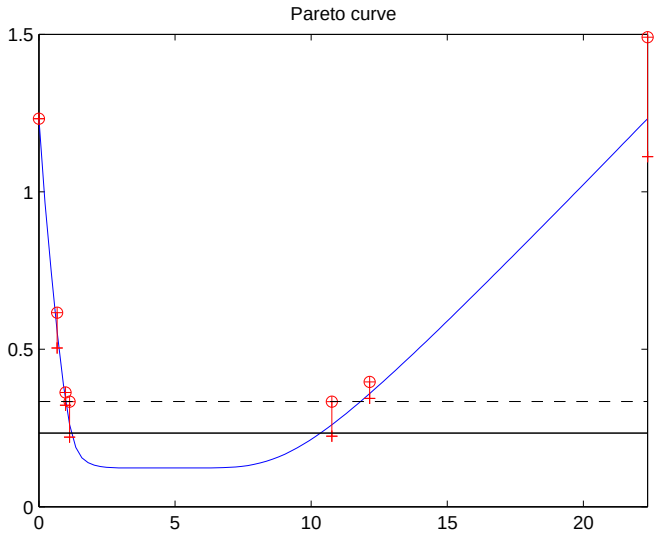
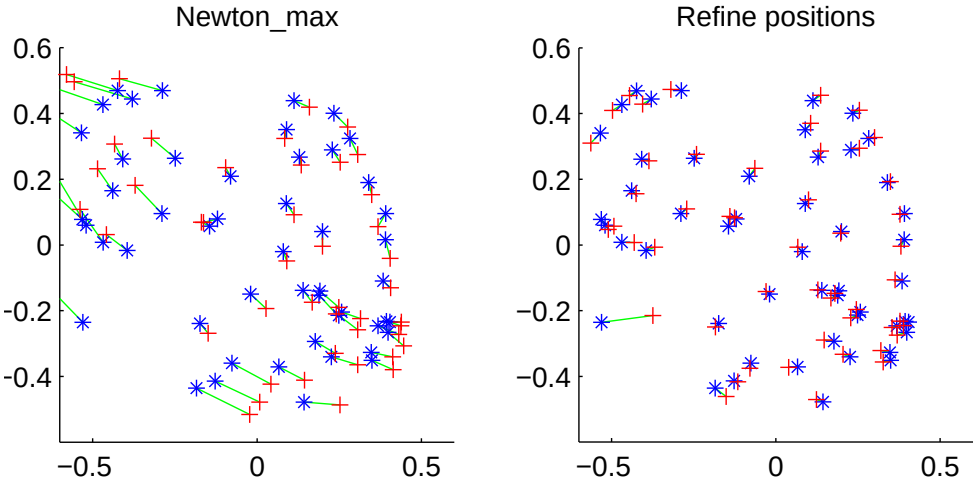


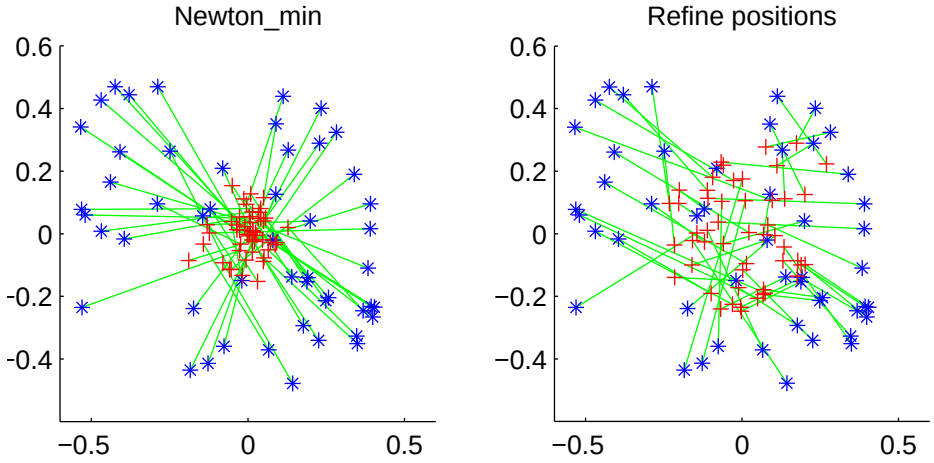
Fig. 6: Maximal trace solution.



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

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

Fig. 7: Minimal trace solution.



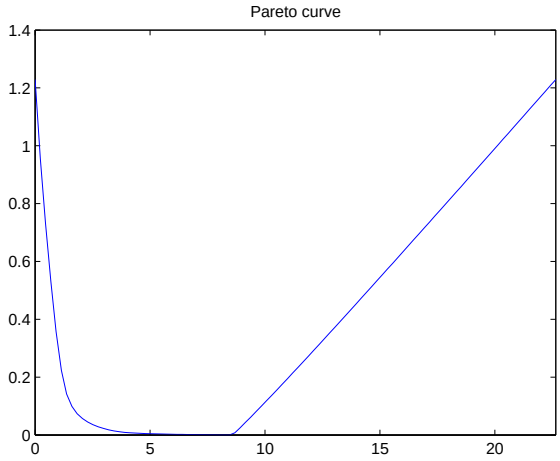


Fig. 8: Graph of φ with $\sigma = 0$.

and its dual

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

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. Throughout we have fixed the tolerance on the misfit $\|\mathcal{P}\circ\mathcal{K}(X)-d\| \leq \sigma + 0.1$. As above, the tests were run on MATLAB version R2016a, on a Dell Optiplex 9020, with Windows 7, Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz and 16 GB RAM.

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. Moreover, instead of identifying cliques, which is computationally heavy, one can try to use other universally rigid (UR) components in the graph, which are easier to identify (e.g. [57]). The difficulty, however, is that this strategy would require an SDP solve for every such component due to noise in the data. 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 $\mathcal{S}_{c,+}^{k,r}$.

We now describe how to evaluate the nearest-point-mapping to the set $\mathcal{S}_{c,+}^{k,r}$—an easy and standard operation due to the Eckart-Young Theorem [21]. To describe this operation, consider any matrix $X \in \mathcal{S}^n$ and a set $\mathcal{Q} \subset \mathcal{S}^n$. The *projection mapping,* $\text{proj}(\cdot)$ is

$$\text{proj}(X;\mathcal{Q}) := \text{argmin}_{Y\in\mathcal{Q}} \|X-Y\|_F.$$

It is well known that $\text{proj}(X;\mathcal{Q})$ is a singleton for every nonempty closed convex set $\mathcal{Q}$. 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

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

Table 5: Numerical results for the Pareto search strategy

Specifications				Time (s)		Residual (%R)		RMSD (%R)	
n	% noise	R	% dens.	initial	refine	initial	refine	initial	refine
1000	0	0.10	2.9	5.2	0.5	0.9	0.0	35.2	0.4
1000	10	0.10	2.9	4.5	0.6	1.9	1.1	64.6	3.8
1000	20	0.10	2.9	4.3	0.6	3.1	2.1	93.3	7.9
1000	30	0.10	2.9	4.1	0.5	4.3	3.2	120.3	12.4
1000	10	0.30	21.6	28.2	1.0	3.4	3.3	5.2	1.0
1000	10	0.25	15.7	14.1	1.2	2.9	2.8	7.5	1.2
1000	10	0.20	10.6	6.6	0.7	2.5	2.2	12.2	1.5
1000	10	0.15	6.2	4.1	0.6	2.1	1.6	23.8	2.1
1000	20	0.10	2.9	4.4	0.6	3.1	2.1	93.3	7.9
2000	20	0.10	2.9	12.3	2.1	2.7	2.2	67.1	4.5
3000	20	0.10	2.9	26.5	3.5	2.6	2.2	56.2	3.5
4000	20	0.08	1.9	63.8	13.3	2.2	1.8	82.2	3.8
5000	20	0.08	1.9	103.7	12.9	2.1	1.8	74.1	3.4
6000	20	0.08	1.9	181.7	29.5	2.1	1.8	68.6	3.0
7000	20	0.06	1.1	247.7	27.2	1.7	1.3	119.8	4.5
8000	20	0.06	1.1	324.0	27.1	1.7	1.4	113.5	3.7
9000	20	0.06	1.1	402.5	38.4	1.6	1.4	108.1	3.4
10000	20	0.06	1.1	482.4	57.2	1.6	1.4	103.9	3.1
11000	20	0.05	0.8	814.5	45.6	1.4	1.1	145.9	3.8
12000	20	0.05	0.8	811.8	55.6	1.4	1.1	142.5	3.6
13000	20	0.05	0.8	957.9	66.5	1.4	1.1	136.1	4.2
14000	20	0.05	0.8	1353.4	83.6	1.4	1.1	134.0	4.4
15000	20	0.05	0.8	1565.2	98.1	1.4	1.1	128.9	4.8

On the other hand, we have

$$\operatorname{proj}\left(Z;\mathcal{S}_+^{k-1,r}\right) = U \operatorname{Diag}\left(0,\dots,0,\lambda_{k-r}^+(Z),\dots,\lambda_{k-1}^+(Z)\right)U^T,$$

where $\lambda_1(Z) \le \ldots \le \lambda_{k-1}(Z)$ are the eigenvalues of Z and the superscript $\lambda_i^+(Z)$ refers to their positive part, and U is an orthogonal matrix of eigenvectors in the orthogonal spectral decomposition $Z = U\operatorname{Diag}(\lambda(Z))U^T$.

Thus computing the matrix $\operatorname{proj}(X;\mathcal{S}_{c,+}^{k,r})$ requires no more than an eigenvalue decomposition. Moreover, if the embedding dimension r is small, then we can take advantage of special routines that find a few (r) eigenpairs. In MATLAB we first shift away from 0 and find the Cholesky factorization of $Y = W + I$: [Ry,˜, S] = chol(W + speye(n))]; the smallest eigenvalue 0 is shifted to 1, $Ye = e$ and $Y^{-1}e = e$. Moreover, $\operatorname{tr}(W) + 1$ is now a valid upper bound for the $r + 1$-st smallest eigenvalue. We can

deflate the eigenvalue for the normalized e by shifting with

$$\sigma = \frac{\mathrm{tr}(W)+1}{n}, \quad Y + \sigma ee^T.$$

This deflation means that we now need to find only the smallest r eigenpairs of $Y + \sigma ee^T$ rather than $r+1$ for W. The MATLAB eigs routine requires repeated evaluations of

$$(Y + \sigma ee^T)^{-1} y = (Y + \sigma ee^T) \backslash y.$$

We can apply the Sherman-Morrison-Woodbury formula with the rank one perturbation of Y and take advantage of the fact that $Y^{-1}e = e$. With $\alpha = \frac{\sigma}{1+\sigma n} e$ we get that the matrix division is equivalent to

$$(Y + \sigma ee^T)^{-1} y = S(Ry \backslash (Ry^T \backslash (S^T y))) - \alpha e^T y.$$

In addition, we can take advantage of the structure of $\mathcal{K}$ in the SNL application. In each step we find $B = \mathcal{K}^\dagger(D)$ for some nonnegative $D \in \mathcal{S}_h^k$. Since D is a *noisy* EDM, we do not necessarily have $B \succeq 0$. However, $B \in \mathcal{S}_c$. Therefore we deflate the eigenvector of ones. This is equivalent to using the matrix U above. We now find the $k-r$ smallest eigenpairs of $B + \alpha ee^T$ for an $\alpha < \min\{-1, \lambda_1(B)\}$, where we zero out any negative eigenvalues to get the nearest matrix in $\mathcal{S}_{c,+}^{k,r}$.

Appendix B. Solving the small least squares problem. We now describe how to easily solve the least squares system (3.2) 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 $\mathrm{svec}\,(Z)$ 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 overdetermined system

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

One approach now is simply to expand the objective

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

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 $\|A(\mathrm{svec}\, Z) - d\|_2^2 = \|R(\mathrm{svec}\, Z) - Q^T d\|^2$. 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, we have found that the cone constraint is often *inactive*. The reason is that under reasonable conditions (see Theorem 2.2), in a noiseless situation, there

is a unique solution to the equation $\mathcal{A}(Z) = d$. This solution then is 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*.

In the case that Z is not positive definite we have generated a random sketch matrix S, e.g., [39]. This changes the highly overdetermined cone constrained least squares problem to one with a reasonable number of constraints after replacing the data A, d with SA, Sd, respectively.

Appendix C. Robustness of facial reduction. In this section, we provide rudimentary robustness guarantees on 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 [16–18]. 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 $\|UU^T - VV^T\|^2 = 2\operatorname{tr}(I - (V^T U)(V^T U)^T) = 2\|\sin(\Theta)\|^2$. 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*

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

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

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

Proof. Optimality conditions for the optimization problem

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

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

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

Combining this with equation (C.1) we deduce

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

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

$$\|UU^T XUU^T - UYU^T\| = \|U^T XU - Y\|,$$

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* range$U$ *and* 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 $\Delta = I - (V^T U)(V^T U)^T$. We successively deduce

$$\begin{aligned}
\operatorname{dist}^2(VZV^T; U\mathcal{S}^r_+ U^T) &= \|VZV^T - UU^T(VZV^T)UU^T\|^2 \\
&= \|VZV^T\|^2 - 2\operatorname{tr}(VZV^T UU^T VZV^T UU^T) + \operatorname{tr}(U^T VZV^T UU^T VZV^T U) \\
&= \|Z\|^2 - 2\operatorname{tr}\Big(\big(Z(V^T U)(V^T U)^T\big)^2\Big) + \operatorname{tr}\Big(\big(Z(V^T U)(V^T U)^T\big)^2\Big) \\
&= \operatorname{tr}\Big(Z^2 - \big(Z(V^T U)(V^T U)^T\big)^2\Big) \\
&= \operatorname{tr}\Big(Z^2 - \big(Z - Z\Delta\big)^2\Big) = \operatorname{tr}\Big(2Z^2\Delta - Z\Delta Z\Delta\Big) = 2\|\Delta^{\frac{1}{2}} Z\|^2 - \|\Delta^{\frac{1}{2}} Z\Delta^{\frac{1}{2}}\|^2.
\end{aligned}$$

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 [44], 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 $\mathrm{Alg}(d)$ 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.

Appendix D.

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, *Graph rigidity via Euclidean distance matrices*, Linear Algebra Appl., 310 (2000), pp. 49–165.

[3] 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.

[4] S. Aravkin, J. Burke, D. Drusvyatskiy, M.P. Friedlander, and S. Roy, *Level-set methods for convex optimization*, Preprint, (2016).

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

[7] 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.

[8] 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.

[9] P. Biswas, T.-C. Liang, K.-C. Toh, Ta-Chung Wang, and Yinyu Ye, *Semidefinite programming approaches for sensor network localization with noisy distance measurements*, IEEE Transactions on Automation Science and Engineering, 3 (2006), pp. 360–371.

[10] 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.

[11] 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.

[12] 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.

[13] R. Connelly and M. Gortler, *iterative universal rigidity*, tech. report, Dept. of Math., Cornell University, Ithaca, NY, 2014.

[14] R. Connelly and S.J. Gortler, *Iterative universal rigidity*, Discrete Comput. Geom., 53 (2015), pp. 847–877.

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

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

[17] C. Davis and W.M. Kahan, *The rotation of eigenvectors by a perturbation. III*, SIAM J. Numer. Anal., 7 (1970), pp. 1–46.

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

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

[20] D. Drusvyatskiy, G. Pataki, and H. Wolkowicz, *Coordinate shadows of semidefinite and Euclidean distance matrices*, SIAM J. Optim., 25 (2015), pp. 1160–1178.

[21] C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrica, 1 (1936), pp. 211–218.

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

[23] G.H. Golub and C.F. Van Loan, *Matrix Computations*, The John Hopkins University Press, Baltimore, 1983.

[24] S.J. Gortler, A.D. Healy, and D.P. Thurston, *Characterizing generic global rigidity*, Amer. J. Math., 132 (2010), pp. 897–939.

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

[26] T.L. Hayden, J. Lee, J. Wells, and P. Tarazaga, *Block matrices and multispherical struc-*

ture of distance matrices, Linear Algebra Appl., 247 (1996), pp. 203–216.
[27] T.L. Hayden, J. Wells, W.M. Liu, and P. Tarazaga, *The cone of distance matrices*, Linear Algebra Appl., 144 (1991), pp. 153–169.
[28] 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.
[29] S. Kim, M. Kojima, and H. Waki, *Exploiting sparsity in SDP relaxation for sensor network localization*, SIAM J. Optim., 20 (2009), pp. 192–215.
[30] S. Kim, M. Kojima, H. Waki, and M. Yamashita, *A sparse version of full semidefinite programming relaxation for sensor network localization problems*, Tech. Report B-457, Department of Mathematical and Computing Sciences Tokyo Institute of Technology, Oh-Okayama, Meguro, Tokyo 152-8552, 2009.
[31] N. Krislock, *Semidefinite Facial Reduction for Low-Rank Euclidean Distance Matrix Completion*, PhD thesis, University of Waterloo, 2010.
[32] N. Krislock and H. Wolkowicz, *Explicit sensor network localization using semidefinite representations and facial reductions*, SIAM Journal on Optimization, 20 (2010), pp. 2679–2708.
[33] H. Kurata and P. Tarazaga, *Multispherical Euclidean distance matrices*, Linear Algebra Appl., 433 (2010), pp. 534–546.
[34] ———, *Majorization for the eigenvalues of Euclidean distance matrices*, Linear Algebra Appl., 436 (2012), pp. 1473–1481.
[35] M. Laurent, *Cuts, matrix completions and graph rigidity*, Math. Programming, 79 (1997), pp. 255–284.
[36] 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.
[37] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino, *Euclidean distance geometry and applications*, SIAM Review, 56 (2014), pp. 3–69.
[38] J. Nie, *Sum of squares method for sensor network localization*, Comput. Optim. Appl., 43 (2009), pp. 151–179.
[39] M. Pilanci and M.J. Wainwright, *Randomized sketches of convex programs with sharp guarantees*, IEEE Trans. Infor. Theory, 61 (2015), pp. 5096–5115.
[40] T.K. Pong and P. Tseng, *(Robust) edge-based semidefinite programming relaxation of sensor network localization*, Tech. Report Jan-09, University of Washington, Seattle, WA, 2009.
[41] ———, *Robust edge-based semidefinite programming relaxation of sensor network localization*, Math. Program., 130 (2011), pp. 321–358.
[42] 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*.
[43] A. Singer, *A remark on global positioning from local distances*, Proc. Natl. Acad. Sci. USA, 105 (2008), pp. 9507–9511.
[44] A. M-C So and Y. Ye, *Theory of semidefinite programming for sensor network localization*, Math. Program. Ser. B, (2007), pp. 367–384.
[45] P. Tarazaga, *Faces of the cone of Euclidean distance matrices: characterizations, structure and induced geometry*, Linear Algebra Appl., 408 (2005), pp. 1–13.
[46] P. Tarazaga and J.E. Gallardo, *Euclidean distance matrices: new characterization and boundary properties*, Linear Multilinear Algebra, 57 (2009), pp. 651–658.
[47] P. Tarazaga, T.L. Hayden, and J. Wells, *Circum-Euclidean distance matrices and faces*, Linear Algebra Appl., 232 (1996), pp. 77–96.
[48] 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.
[49] P. Tseng, *Second-order cone programming relaxation of sensor network localization*, SIAM J. Optim., 18 (2007), pp. 156–185.
[50] 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).
[51] 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.
[52] ———, *Sparse optimization with least-squares constraints*, SIAM J. Optim., 21 (2011), pp. 1201–1229.
[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] Z. Zhu, A.M.-C. So, and Y. Ye, *Universal rigidity and edge sparsification for sensor network localization*, SIAM J. Optim., 20 (2010), pp. 3059–3081.