A Fast Matrix Majorization-Projection Method for Penalized Stress Minimization with Box Constraints

Shenglong Zhou, Naihua Xiu and Hou-Duo Qi

***Abstract*—Kruskal's stress minimization, though nonconvex and nonsmooth, has been a major computational model for dissimilarity data in multidimensional scaling. Semidefinite Programming (SDP) relaxation (by dropping the rank constraint) would lead to a high number of SDP cone constraints. This has rendered the SDP approach computationally challenging even for problems of small size. In this paper, we reformulate the stress optimization as an Euclidean Distance Matrix (EDM) optimization with box constraints. A key element in our approach is the conditional positive semidefinite cone with rank cut. Although nonconvex, this geometric object allows a fast computation of the projection onto it and it naturally leads to a majorization-minimization algorithm with the minimization step having a closed-form solution. Moreover, we prove that our EDM optimization follows a continuously differentiable path, which greatly facilitated the analysis of the convergence to a stationary point. The superior performance of the proposed algorithm is demonstrated against some of the state-of-the-art solvers in the field of sensor network localization and molecular conformation.**

***Index Terms*—Raw stress, multidimensional scaling, Euclidean distance matrix, semidefinite programming, majorization-minimization, sensor network localization.**

I. Introduction

KRUSKAL'S stress minimization [1], though nonconvex and nonsmooth, has been a major computational model for dissimilarity data in multidimensional scaling (MDS) [2], [3]. Its popularity among the practitioners has been significantly enhanced by its companion algorithm SMACOF [4], [5]. In the particular application of range-based sensor network localization (SNL), the stress minimization is equivalent to the maximum likelihood criterion if the disturbances of the observed ranges are of white noises. In its original form, for a given subset of dissimilarities (e.g., noisy distances) denoted by $\{\delta_{ij}\}$ among n items, the stress minimization tries to find a best set of embedding points $\mathbf{x}_i \in \Re^r$, $i = 1, \ldots, n$ such that they solve (see [3, P. 171])

$$\min\ \sigma_r(X) := \sum_{i,j} W_{ij} \left(\|\mathbf{x}_i - \mathbf{x}_j\| - \delta_{ij}\right)^2, \qquad (1)$$

where the weights $W_{ij} > 0$ if δ_{ij} is known and 0 otherwise, the norm $\|\cdot\|$ is the Euclidean norm in $\Re^r$, and $X := [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ is the matrix of coordinates. The most interesting case is when r is small (e.g., $r = 2, 3$ for visualization). The function $\sigma_r(X)$ is known as the raw stress.

In many applications such as molecular conformation [6], lower and upper bounds data on the distances are also known:

$$\ell_{ij} \le \|\mathbf{x}_i - \mathbf{x}_j\| \le u_{ij}, \quad \forall\ (i,j), \qquad (2)$$

where $0 \le \ell_{ij} \le u_{ij}$ and $\ell_{ii} = u_{ii} = 0$. In applications such as nonlinear dimensionality reduction [7] and sensor network localization (SNL) [8], [9], upper bounds u_{ij} can be computed by the shortest path distances and ℓ_{ij} can be simply set to be zeros.

Prior to the stress criterion, the classical MDS (cMDS) [10]–[12] (see also [8], [13]) may be the only viable method for dissimilarity data. The key difference is that cMDS uses "squared" distances $\|\mathbf{x}_i - \mathbf{x}_j\|^2$, which naturally lead to advanced Euclidean Distance Matrix (EDM) models [14, Sect. III(A)]. In contrast, the stress function makes use of "plain" distances $\|\mathbf{x}_i - \mathbf{x}_j\|$, which often lead to models based on coordinates [15]. Existing research that attempts to represent plain distances by EDM often leads to a large number of positive semidefinite cone constraints, making the resulting matrix optimization problem extremely challenging to solve (see, e.g., [14, Eq. (8)]). The purpose of this paper is to propose a new EDM reformulation of the stress criterion under box constraints. We will develop a fast majorization-projection method, which falls in the general framework of [16]. Its superior performance against several state-of-the-art algorithms will be demonstrated through a number of artificial SNL data and real data from molecular conformation.

In the following, we give a short literature review that motivated our research, followed by our proposed approach and main contributions.

A. Literature Review

We will discuss two groups of algorithms, namely the coordinates descent algorithms (enhanced by the majorization technique) and methods of matrix optimization including the EDM and the Semi-Definite Programming (SDP) approach.

(a) Algorithms of coordinates descent. Early popularity of the stress minimization criterion was largely due to the fact that classical optimization methods can be applied directly and was subsequently enhanced by the well-known algorithm of SMACOF [4]. The key idea of SMACOF was to construct a majorization function $m(X, X^k)$ at the current iterate X^k such that $\sigma_r(X) \le m(X, X^k)$. Instead of minimizing $\sigma_r(X)$, it minimizes $m(X, X^k)$ to get the next iterate X^{k+1}. Since the

First version: August 15, 2017. This version: January 25, 2018.

S. Zhou is with School of Mathematics, University of Southampton. Email: sz3g14@soton.ac.uk.

N. Xiu is with the Department of Applied Mathematics, Beijing Jiaotong University, Beijing, China. Email: nhxiu@bjtu.edu.cn.

H.-D. Qi is with the School of Mathematics, University of Southampton, UK. Email: hdqi@soton.ac.uk.

majorization function is convex and quadratic, a system of linear equations is solved each step to get X^{k+1}. The algorithm is well documented in [3, Chp. 8]. However, as demonstrated in [9], SMACOF performs poorly for SNL problems.

In another important development, the stress function can be majorized componentwise in the sense that

$$\sigma_r(X) \leq m_1(\mathbf{x}_1, X^k)+m_2(\mathbf{x}_2, X^k)+\cdots+m_n(\mathbf{x}_n, X^k), \quad (3)$$

with each piece $m_i(\mathbf{x}_i, X^k)$ being easy to be minimized. Therefore, (3) leads to a distributed optimization (see [15, Sect. I (C)] for a relevant review). A nice example can be found in [14, Sect. III (C)], where it is showed that both the squared and the plain distances can be majorized as

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 \leq q(\mathbf{x}_i, \mathbf{x}_j, X^k) \quad (4)$$

and

$$-\|\mathbf{x}_i - \mathbf{x}_k\| \leq l(\mathbf{x}_i, \mathbf{x}_j, X^k), \quad (5)$$

with the functions $q(\cdot)$ and $l(\cdot)$ being respectively quadratic and linear in $\mathbf{x}_i$ and $\mathbf{x}_j$, constructed in such a way that each piece $m_i(\cdot)$ in (3) can be obtained through those functions. We will show that this simple majorization scheme works well for some problems while having difficulties with other problems described in the numerical part.

Convex relaxation also represents a major approach to (1). For example, based on the observation

$$(\|\mathbf{x}_i - \mathbf{x}_j\| - \delta_{ij})^2 = \min_{\|\mathbf{y}\|=\delta_{ij}} \|\mathbf{x}_i - \mathbf{x}_j - \mathbf{y}\|^2,$$

Soares et al. [15] studied the convex relaxation by replacing the constraint $\|\mathbf{y}\| = \delta_{ij}$ by its convex counterpart $\|\mathbf{y}\| \leq \delta_{ij}$. This convex relaxation scheme is further enhanced by Piovesan and Erseghe [17] and it is solved by an Alternating Direction Method of Multipliers (ADMM). The convex relation with the majorization technique was considered in [18]. Another type of stress majorization was proposed in [9], resulting in the well-known ARAP (As Rigid As Possible) algorithm for SNL. All of those methods are of distributed nature and we will compare them with our method in the numerical part.

(b) EDM and SDP optimization. They are matrix optimization and are very popular in the past decade because they often provide a flexible framework to obtain convex relaxation that can be solved by off-shelf SDP solvers. Early applications of EDM and SDP to molecular conformation and SNL can be respectively found in [6] and [19]. There exist a large body of publications that are beyond our scope to review here. We only focus on those that are pertinent to the problem (1).

Let $\mathcal{S}^n$ denote the space of all $n \times n$ symmetric matrices, endowed with the standard inner product. Let $\mathcal{S}^n_+$ be the cone of positive semidefinite matrices in $\mathcal{S}^n$. A matrix $D \in \mathcal{S}^n$ is called an EDM if there exist a set of points $\mathbf{x}_i \in \Re^r$, $i = 1, 2 \ldots, n$ such that the (i, j)th element of D is given by

$$D_{ij} := \|\mathbf{x}_i - \mathbf{x}_j\|^2, \quad i, j = 1, \ldots, n.$$

Here ":=" means "define". The smallest dimension r is called the embedding dimension of D and $r = \text{rank}(JDJ)$, where $J := I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ is known as the centring matrix with I being the identity matrix in $\mathcal{S}^n$ and $\mathbf{1}$ being the vector of all ones in $\Re^n$. We use $\mathcal{D}^n$ to denote the set of all Euclidean distance matrices of size $n \times n$.

If $D \in \mathcal{D}^n$ is given, one can easily generate a set of the embedding points $\{\mathbf{x}_i\}$ by applying the classical MDS [3, Chp. 12]. Therefore, the stress problem (1) can be reformulated in terms of EDM as

$$\begin{array}{ll} \min_D & \sum_{i,j} W_{ij} \left(\sqrt{D_{ij}} - \delta_{ij}\right)^2 \\ \text{s.t.} & D \in \mathcal{D}^n, \ \text{rank}(JDJ) \leq r. \end{array} \quad (6)$$

In the application of SNL, some of the data points of $\mathbf{x}_i$ are already known to be anchors. That is, $\mathbf{x}_i = \mathbf{a}_i$, $i = 1, \ldots, m$ are known. In this case, the distances D_{ij} among the anchors are known. The problem (6) with the fixed distance constraints $D_{ij} = \|\mathbf{a}_i - \mathbf{a}_j\|^2$, $i, j = 1, \ldots, m$ is same as [14, Problem (6)]. By dropping the rank constraint, Problem (6) has a natural SDP reformulation as shown in [14, Problem (8)]:

$$\begin{array}{ll} \min_{D,T \in \mathcal{S}^n} & \sum_{i,j} W_{ij} \left(D_{ij} - 2T_{ij}\delta_{ij}\right) \\ \text{s.t.} & D \in \mathcal{D}^n, D_{ij} = \|\mathbf{a}_i - \mathbf{a}_j\|^2, i < j = 2, \ldots, m \\ & T_{ij}^2 \leq D_{ij} \quad \text{for } i < j = 2, \ldots, n. \end{array}$$

This problem is SDP because the inequalities $T_{ij}^2 \leq D_{ij}$ can be represented as SDP cone constraints via the Schur complement:

$$\begin{bmatrix} 1 & T_{ij} \\ T_{ij} & D_{ij} \end{bmatrix} \in \mathcal{S}^2_+, \quad i < j = 2, \ldots, n$$

and $D \in \mathcal{D}^n$ can also be represented as SDP constraints on $\mathcal{S}^n_+$ due to the known characterization [10]:

$$D \in \mathcal{D}^n \iff \text{diag}(D) = 0 \ \text{ and } \ -(JDJ) \in \mathcal{S}^n_+.$$

Hence, there are about $n(n-1)/2$ cone constraints on $\mathcal{S}^2_+$ and one big cone constraint on $\mathcal{S}^n_+$ in addition to at least $(n + m(m-1)/2)$ linear constraints. Even for a small n, this presents a challenging task for off-shelf SDP solvers such as SDPT3 [20]. We note that this challenge has not taken the rank constraint into account.

We finish our review by mentioning two variants of the stress function. When the squared distance is used, we have the so-called S-stress problem [3, Chp. 11]:

$$\min \sigma_S(X) = \sum_{i,j} W_{ij}(\|\mathbf{x}_i - \mathbf{x}_j\|^2 - \delta_{ij}^2)^2.$$

Its SDP relaxation is simpler than that for (1) (see [21, Sect. III] for a detailed description). Its EDM relaxation has been studied in [22]–[25] and [26]–[28]. When the absolute value is used to measure the error, we end up with the so-called robust MDS problem:

$$\min \sigma_R(X) = \sum_{i,j} W_{ij} \left| \|\mathbf{x}_i - \mathbf{x}_j\|^2 - \delta_{ij}^2 \right|, \quad (7)$$

whose SDP relaxation is initially studied by Biswas and Ye [19]. This framework of [19] has been followed up by many. In particular, the edge-based SDP relaxation seems to stand out as a viable numerical model [29], [30] and the software SFSDP [31] is a high-level implementation of such SDP specifically developed for SNL. However, a common drawback

among those EDM/SDP models is that they are "centralized", meaning that a large linear systems is usually solved each step and hence it is computationally expensive.

B. Our Approach and Main Contributions

The question to solve is the stress minimization (1) with the box constraints (2). In terms of EDM, it is the problem (6) with the box constraints. We state it below:

$$\begin{array}{ll} \min_D & \sum_{i,j} W_{ij}\left(\sqrt{D_{ij}}-\delta_{ij}\right)^2 \\ \text{s.t.} & D\in\mathcal{D}^n,\ \text{rank}(JDJ)\le r \\ & D\in\mathcal{B}:=\{A\in\mathcal{S}^n \mid L\le A\le U\}, \end{array} \tag{8}$$

where $L_{ij}:=\ell_{ij}^2$ and $U_{ij}:=u_{ij}^2$, $i,j=1,\ldots,n$.

It is known [32] that

$$D\in\mathcal{D}^n \iff \text{diag}(D)=0 \quad \text{and} \quad -D\in\mathcal{K}_+^n, \tag{9}$$

where $\mathcal{K}_+^n$ is the conditional positive semidefinite cone:

$$\mathcal{K}_+^n:=\left\{A\in\mathcal{S}^n | \mathbf{v}^TA\mathbf{v}\ge 0 \text{ for all } \mathbf{v}\in\Re^n \text{ with } \mathbf{v}^T\mathbf{1}=0\right\}.$$

The geometric object $\mathcal{K}_+^n(r)$, known as the *conditional positive semidefinite cone with rank-r cut* [33], is defined by

$$\mathcal{K}_+^n(r):=\mathcal{K}_+^n\cap\{A\in\mathcal{S}^n \mid \text{rank}(JAJ)\le r\}.$$

We define the distance between a point $A\in\mathcal{S}^n$ and $\mathcal{K}_+^n(r)$:

$$\text{dist}(A,\ \mathcal{K}_+^n(r)):=\min\{\|A-Y\| \mid Y\in\mathcal{K}_+^n(r)\},$$

where the matrix norm $\|A\|$ is the Frobenius norm. Define the function $g:\mathcal{S}^n\mapsto\Re$ by

$$g(A):=\frac{1}{2}\text{dist}^2(-A,\ \mathcal{K}_+^n(r)),\quad \forall\ A\in\mathcal{S}^n. \tag{10}$$

We emphasize that $(-A)$ is used in the definition of $g(A)$ because it is $(-D)$ that belongs to $\mathcal{K}_+^n$ in (9). It is obvious that $-D\in\mathcal{K}_+^n(r)$ if and only if $g(D)=0$. Therefore, the problem (8) is equivalent to

$$\begin{array}{ll} \min_D & f(D):=\sum_{i,j}W_{ij}\left(\sqrt{D_{ij}}-\delta_{ij}\right)^2 \\ \text{s.t.} & g(D)=0,\quad D\in\mathcal{B}, \end{array} \tag{11}$$

where the diagonal constraint $\text{diag}(D)=0$ in (9) has been integrated into the box constraint in $\mathcal{B}$ due to $L_{ii}=U_{ii}=0$. We refer to (11) as the Square-Root EDM (`SQREDM`) model for the stress minimization (1) with the box constraint (2).

Let us take a close look at the model (11). The objective $f(D)$ is convex, though it may not be differentiable at some points. The box constraint $\mathcal{B}$ is as simple as we can wish for. The difficult part is the nonlinear equation defined by $g(D)$, which measures the violation of the feasibility of a matrix $-D$ belonging to $\mathcal{K}_+^n(r)$. It has long been known that `cMDS` works very well as long as the matrix D is close to be Euclidean. This means that small violation of being Euclidean would not cause a major concern for the final embedding. Therefore, we propose to penalize the function $g(D)$ to get the following optimization problem:

$$\min\ F_\rho(D):=f(D)+\rho g(D),\quad \text{s.t.}\ \ D\in\mathcal{B}, \tag{12}$$

where $\rho>0$ is a penalty parameter. We further propose a majorization method for (12). At the current iterate D^k, we will construct a convex majorization function $g_m(D,D^k)$ for $g(D)$ and update D^k by

$$D^{k+1}=\arg\min f(D)+\rho g_m(D,D^k)\ \ \text{s.t.}\ \ D\in\mathcal{B}. \tag{13}$$

The rest of the paper is to provide the water-tight evidences both in theory and numerically to justify the proposed approach. The main contributions are summarized as follows.

(i) We will show in Theorem 3.2 that the optimal solution of the penalized problem (12) is an approximately optimal (i.e., ϵ-optimal) solution of the original problem (11). Moreover, any accumulation point of the generated sequence $\{D^k\}$ is an approximate KKT point of (11) (Theorem3.7(ii)). We note that the classical results on penalty methods [34] are not applicable here because both the function $f(D)$ and $g(D)$ are not differentiable.

(ii) The majorization function can be economically constructed via PCA (Principle Component Analysis) on a centralized data matrix. Furthermore, the subproblem (13) can be computed in a distributed fashion (i.e., computed elementwise) each with a close-form formula (Prop. 3.5 and Eq. (29)). The use of the *depressed cubic equation* in deriving the formula is interesting on its own, given its recent success in compressed sensing [35].

(iii) Although the objective function $f(D)$ is not differentiable, we will show that it follows a continuously differentiable path during the iteration process (Prop. 3.6). This technical result is important because it avoids using the subdifferential of $f(\cdot)$ to perform the convergence analysis in Theorem 3.7, which shows that any accumulation point is a stationary point of (12).

(iv) Finally, the efficiency of the proposed algorithm is demonstrated against a few state-of-the-art methods (`SMACOF` (Matlab implementation from [36]), `ARAP` [9], `ADMMSNL` [17] and `SFSDP` [31]) on a number of artificial and real data sets, which include SNL and molecular conformation problems. The embedding quality of our method is comparable to or exceeds the best results by these benchmark methods and our method only uses a fraction of the computing time by the others. The speed advantage becomes extremely superior for large network localizations.

C. Organization of the paper

In Sect. II, we will describe how the penalty function $g(A)$ is constructed through a PCA-style formula. We will study its properties, which will lead to a natural choice of majorization. Sect. III contains the main theoretical contributions. We will develop our square-root EDM model and a fast algorithm. We will show that the subproblem by majorization is well defined and has a closed-form solution. We will also establish the convergence results for the proposed algorithm under reasonable conditions. The superior performance of the algorithm is demonstrated in Sect. IV against a few of state-of-the-art methods on test problems from SNL and molecular conformation. We conclude the paper in Sect. V.

II. Penalty Function and Its Majorization

The main purpose of this section is to show how the penalty function $g(\cdot)$ in (10) can be efficiently computed and how its majorization function $g_m(D, D^k)$ can be constructed at a given point D^k.

For a given matrix $A \in \mathcal{S}^n$, we consider its orthogonal projection onto $\mathcal{K}^n_+(r)$. Since $\mathcal{K}^n_+(r)$ is not convex (unless $r \geq n-1$), the projection is not unique. Let us denote all the projections by $\Pi^B_{\mathcal{K}^n_+(r)}(A)$, which is defined by

$$\Pi^B_{\mathcal{K}^n_+(r)}(A) := \arg\min_{D\in\mathcal{S}^n} \ \|A-D\| \ \text{ s.t. } \ D \in \mathcal{K}^n_+(r). \quad (14)$$

We use $\Pi_{\mathcal{K}^n_+(r)}(A)$ to denote any one element in $\Pi^B_{\mathcal{K}^n_+(r)}(A)$. We will show below that one particular element can be explicitly computed by the eigen-value decomposition (EVD) of the matrix (JAJ). We make it precise below because it is important to understand our model and for the fast implementation of our algorithm.

Suppose $A \in \mathcal{S}^n$ has the following EVD:

$$A = \lambda_1 \mathbf{p}_1\mathbf{p}_1^T + \lambda_2 \mathbf{p}_2\mathbf{p}_2^T + \cdots + \lambda_n \mathbf{p}_n\mathbf{p}_n^T,$$

where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ are the eigenvalues of A in non-increasing order, and $\mathbf{p}_i$, $i = 1, \ldots, n$ are the corresponding orthonormal eigenvectors. We define a PCA-style matrix truncated at r:

$$\mathrm{PCA}^+_r(A) := \sum_{i=1}^{r} \max\{0, \lambda_i\}\mathbf{p}_i\mathbf{p}_i^T. \quad (15)$$

The following results have been proved in [33].

Lemma 2.1: For a given matrix $A \in \mathcal{S}^n$ and an integer $r \leq n$. Let $\Pi_{\mathcal{K}^n_+(r)}(A)$ be any one element in $\Pi^B_{\mathcal{K}^n_+(r)}(A)$. The following results hold.

(i) [33, Eq. (26), Prop. 3.3] We have

$$\langle \Pi_{\mathcal{K}^n_+(r)}(A),\ A - \Pi_{\mathcal{K}^n_+(r)}(A)\rangle = 0.$$

(ii) [33, Prop. 3.4] The function

$$h(A) := \frac{1}{2}\|\Pi_{\mathcal{K}^n_+(r)}(A)\|^2$$

is well defined and is convex. Moreover,

$$\Pi_{\mathcal{K}^n_+(r)}(A) \in \partial h(A),$$

where $\partial h(A)$ is the subdifferential of $h(\cdot)$ at A.

(iii) [33, Eq. (22), Prop. 3.3] One particular $\Pi_{\mathcal{K}^n_+(r)}(A)$ can be computed through

$$\Pi_{\mathcal{K}^n_+(r)}(A) = \mathrm{PCA}^+_r(JAJ) + (A - JAJ) \quad (16)$$

Remarks:

(R1) In fact, the computational formula for $\Pi_{\mathcal{K}^n_+(r)}(A)$ in Lemma 2.1(iii) is a special choice of what is proved in [33, Prop. 3.3], where $\Pi_{\mathcal{K}^n_+(r)}(A)$ is characterized by $\Pi_{\mathcal{S}^n_+(r)}(JAJ)$ with $\mathcal{S}^n_+(r)$ being the positive semidefinite cone with rank-r cut. The $\mathrm{PCA}^+_r(JAJ)$ is just a special choice of $\Pi_{\mathcal{S}^n_+(r)}(JAJ)$ following (15) and [33, Lemma 2.2] or [37, Lemma 2.9]. We choose PCA^+_r mainly because of its computational simplicity. Form now on, we use $\Pi_{\mathcal{K}^n_+(r)}(A)$ defined by (16).

(R2) It follows from the definition $g(A)$ in (10) and (14) that

$$g(A) = \frac{1}{2}\|A + \Pi_{\mathcal{K}^n_+(r)}(-A)\|^2. \quad (17)$$

Lemma 2.1 allows us to represent $g(A)$ in terms of $h(A)$. This relationship is so important that we include it in the following result.

Lemma 2.2: We have for any $A \in \mathcal{S}^n$

$$g(A) = \frac{1}{2}\|A\|^2 - h(-A) \ \text{ and } \ \|\Pi_{\mathcal{K}^n_+(r)}(A)\| \leq 2\|A\|.$$

Hence, $g(A)$ is a difference of two convex functions.

Proof: It follows from Lemma 2.1(i) that

$$\langle -A,\ \Pi_{\mathcal{K}^n_+(r)}(-A)\rangle = \|\Pi_{\mathcal{K}^n_+(r)}(-A)\|^2.$$

Substituting this into the first equation below to get

$$\begin{aligned} g(A) &= \frac{1}{2}\|A\|^2 + \frac{1}{2}\|\Pi_{\mathcal{K}^n_+(r)}(-A)\|^2 + \langle A,\ \Pi_{\mathcal{K}^n_+(r)}(-A)\rangle \\ &= \frac{1}{2}\|A\|^2 + \frac{1}{2}\|\Pi_{\mathcal{K}^n_+(r)}(-A)\|^2 - \|\Pi_{\mathcal{K}^n_+(r)}(-A)\|^2 \\ &= \frac{1}{2}\|A\|^2 - \frac{1}{2}\|\Pi_{\mathcal{K}^n_+(r)}(-A)\|^2 \\ &= \frac{1}{2}\|A\|^2 - h(-A). \end{aligned}$$

Since $0 \in \mathcal{K}^n_+(r)$ and $\Pi_{\mathcal{K}^n_+(r)}(A) \in \Pi^B_{\mathcal{K}^n_+(r)}(A)$, we have

$$\|A - \Pi_{\mathcal{K}^n_+(r)}(A)\| = \mathrm{dist}(A,\ \mathcal{K}^n_+(r)) \leq \|A - 0\| = \|A\|,$$

which, by the triangle inequality, yields

$$\|\Pi_{\mathcal{K}^n_+(r)}(A)\| \leq \|\Pi_{\mathcal{K}^n_+(r)}(A) - A\| + \|A\| \leq 2\|A\|.$$

This is the second claim in the lemma. ■

It follows from the convexity of $h(\cdot)$ and $\Pi_{\mathcal{K}^n_+(r)}(\cdot)$ being a subgradient of $h(\cdot)$ (Lemma 2.1(ii)) that

$$h(-D) \geq h(-Z) + \langle \Pi_{\mathcal{K}^n_+(r)}(-Z),\ -D + Z\rangle, \ \ \forall\ D, Z \in \mathcal{S}^n.$$

This, with Lemma 2.2, implies for any $D, Z \in \mathcal{S}^n$

$$\begin{aligned} g(D) &= \frac{1}{2}\|D\|^2 - h(-D) \\ &\leq \frac{1}{2}\|D\|^2 - h(-Z) + \langle \Pi_{\mathcal{K}^n_+(r)}(-Z),\ D - Z\rangle \\ &=: g_m(D, Z). \end{aligned} \quad (18)$$

Obviously, $g(D) = g_m(D, D)$ for any D. Hence, $g_m(\cdot,\cdot)$ is a majorization of $g(\cdot)$ [3, Chp. 8].

III. Square-Root EDM Model (SQREDM): Theory and Algorithm

This is the major section that establishes the theory and algorithmic analysis for our proposed approach. It has three parts. In the first part, we study the relationship between the square-root EDM model (11) and its penalized problem (12). A key concept in this part is the ϵ-optimality. In the second part, we show that the majorized subproblem (13) has a closed form solution, which can be computed componentwise. Convergence analysis in included in the final part.

A. Quality of the penalized problem

We note that the classical results on penalty methods [34] for the differentiable case (i.e., all functions involved are differentiable) are not applicable here. Our investigation on the penalty problem (12) is concerned on the quality of its optimal solution when the penalty parameter is large enough. We first introduce the concept of ϵ-optimality.

Definition 3.1: (ϵ-optimal solution) Suppose D^* is an optimal solution of (11). For a given error tolerance $\epsilon > 0$, a point $\widehat{D}$ is called an ϵ-optimal solution of (11) if it satisfies

$$\widehat{D} \in \mathcal{B}, \quad g(\widehat{D}) \leq \epsilon \quad \text{and} \quad f(\widehat{D}) \leq f(D^*).$$

Obviously, if $\epsilon = 0$, $\widehat{D}$ would be an optimal solution of (11). We will show that the optimal solution of (12) is ϵ-optimal provided that ρ is large enough. Let D^*_ρ be an optimal solution of the penalized problem (12) and D_r be any feasible solution of the original problem (11). If the lower bound matrix $L \equiv 0$, then we can simply choose $D_r = 0$. Define

$$\rho_\epsilon := \frac{f(D_r)}{\epsilon}.$$

We have following theorem.

Theorem 3.2: Let $\epsilon > 0$ be given. For any $\rho \geq \rho_\epsilon$, D^*_ρ must be ϵ-optimal. That is,

$$D^*_\rho \in \mathcal{B}, \qquad g(D^*_\rho) \leq \epsilon \qquad \text{and} \qquad f(D^*_\rho) \leq f(D^*).$$

Proof: Since D^*_ρ is an optimal solution of (12), we have $D^*_\rho \in \mathcal{B}$. The rest follows from the following chain of inequalities.

$$\begin{aligned} f(D_r) &= f(D_r) + \rho g(D_r) \qquad (\text{because } g(D_r) = 0) \\ &= F_\rho(D_r) \geq F_\rho(D^*_\rho) \\ &\qquad (\text{because } D^*_\rho \text{ is an optimal solution of (12)}) \\ &= f(D^*_\rho) + \rho g(D^*_\rho) \\ &\geq \rho g(D^*_\rho). \qquad (\text{because } f(D^*_\rho) \geq 0) \end{aligned}$$

Therefore, we have

$$g(D^*_\rho) \leq \frac{f(D_r)}{\rho} \leq \frac{f(D_r)}{\rho_\epsilon} = \epsilon.$$

Furthermore, we have

$$\begin{aligned} f(D^*) &= f(D^*) + \rho g(D^*) \\ &\qquad (\text{because } D^* \in \mathcal{K}^n_+(r), \text{ hence } g(D^*) = 0) \\ &= F_\rho(D^*) \geq F_\rho(D^*_\rho) \\ &\qquad (\text{because } D^*_\rho \text{ is an optimal solution of (12)}) \\ &= f(D^*_\rho) + \rho g(D^*_\rho) \\ &\geq f(D^*_\rho). \qquad (\text{because } \rho g(D^*_\rho) \geq 0) \end{aligned}$$

This completes our proof. ■

Theorem 3.2 states that a global solution of the penalized problem is also an ϵ-optimal for the original problem provided that ρ is large enough. The local version of this result is about ϵ-approximate KKT point. Let the Lagrangian function for (11) be

$$\mathcal{L}(D,\ \beta) := f(D) + \beta g(D), \quad \forall\ D \in \mathcal{S}^n,\ \beta \in \Re.$$

We say a given $\widehat{D}$ is a KKT point of (11) if there exist $\widehat{\beta} > 0$ and $\widehat{\Gamma} \in \partial_D \mathcal{L}(\widehat{D}, \widehat{\beta})$ such that $g(\widehat{D}) = 0$ and

$$\langle \widehat{\Gamma},\ D - \widehat{D} \rangle \geq 0,\ \forall\ D \in \mathcal{B}. \tag{19}$$

For a given $\epsilon > 0$, we say $\widehat{D}$ is an ϵ-approximate KKT point of (11) if $\widehat{\beta} > 0$, $g(\widehat{D}) \leq \epsilon$ and (19) holds. We will show in Theorem 3.7(ii) that any accumulation point of the generated sequence by our algorithm will be an ϵ-approximate KKT point.

B. Solving the Subproblem

We now address how the subproblem (13) is to be solved. For ease of reference, we write the objective function $f(D)$ as in the following form:

$$f(D) = \|\sqrt{W} \circ (\sqrt{D} - \Delta)\|^2,$$

where $\sqrt{W}$ and $\sqrt{D}$ are the componentwise square-root of D and W respectively, and $\circ$ is the Hadamard product between two matrices (e.g., $A \circ B := (A_{ij}B_{ij})$). The solution of the subproblem (13) is about computing an improved solution, denoted by Z^+, from the current point Z by solving the problem:

$$Z^+ := \arg\min \{f(D) + \rho g_m(D, Z)\}, \quad \text{s.t.} \quad D \in \mathcal{B}. \tag{20}$$

This subproblem has a perfect separability property that makes it very easy to solve as we see below.

$$\begin{aligned} Z^+ &= \arg\min_{D \in \mathcal{B}}\ f(D) + \rho g_m(D, Z) \\ &= \arg\min_{D \in \mathcal{B}}\ \|\sqrt{W} \circ (\sqrt{D} - \Delta)\|^2 \\ &\qquad + \frac{\rho}{2}\|D\|^2 + \rho\langle \Pi_{\mathcal{K}^n_+(r)}(-Z), D - Z\rangle \\ &= \arg\min_{D \in \mathcal{B}}\ \langle W, D\rangle - 2\langle W \circ \Delta, \sqrt{D}\rangle \\ &\qquad + \frac{\rho}{2}\|D\|^2 + \rho\langle \Pi_{\mathcal{K}^n_+(r)}(-Z), D\rangle \\ &= \arg\min_{D \in \mathcal{B}}\ \frac{\rho}{2}\|D - Z_\rho\|^2 - 2\langle W \circ \Delta, \sqrt{D}\rangle \\ &= \arg\min_{D \in \mathcal{B}}\ \frac{1}{2}\|D - Z_\rho\|^2 - \frac{2}{\rho}\langle W \circ \Delta, \sqrt{D}\rangle, \end{aligned} \tag{21}$$

where the matrix $Z_\rho := -W/\rho - \Pi_{\mathcal{K}^n_+(r)}(-Z)$. Therefore, the solution Z^+ can be computed elementwise due to the separable property in (21):

$$\begin{aligned} Z^+_{ij} &= \arg\min\ \frac{1}{2}\left[D_{ij} - (Z_\rho)_{ij}\right]^2 - \frac{2}{\rho}W_{ij}\delta_{ij}\sqrt{D_{ij}} \\ &\qquad \text{s.t.} \qquad L_{ij} \leq D_{ij} \leq U_{ij}. \end{aligned} \tag{22}$$

We denote the subproblem solution process by

$$Z^+ = \texttt{SQREDM}(W, \Delta, Z). \tag{23}$$

We will show how SQREDM can be computed.

Let us consider a simplified one-dimensional optimization problem, whose solution will eventually give rise to SQREDM. For given $\omega \in \Re$ and $\alpha \geq 0$, we aim to compute

$$\begin{aligned} x^+(\omega, \alpha) &:= \arg\min_{x \geq 0}\ p(x) := \frac{1}{2}(x - \omega)^2 - 2\alpha\sqrt{x} \\ &=: \texttt{dcroot}[\omega, \alpha]. \end{aligned} \tag{24}$$

We will prove below that x^+ (we often drop its dependence on (ω, α) when no confusion is caused) is well defined and it can be computed through finding the positive root of a depressed cubic equation. This is why we denote $x^+(\omega, \alpha)$ by $\texttt{dcroot}[\omega, \alpha]$ for easy reference later on.

If $\alpha = 0$, it is obvious that $x^+ = \max\{0, \omega\}$. The following result considers the case $\alpha > 0$.

Proposition 3.3: Suppose $\alpha > 0$. Define

$$u := \frac{\alpha}{2}, \ v := \frac{\omega}{3}, \text{ and } \tau := u^2 - v^3.$$

Then the solution x^+ is unique and $x^+ > 0$. Moreover, x^+ depends on the sign of τ and is stated as follows

(i) If $\tau \geq 0$, then

$$x^+ = \left[(u + \sqrt{\tau})^{1/3} + (u - \sqrt{\tau})^{1/3}\right]^2$$

(ii) If $\tau < 0$, then $\omega > 0$ and

$$x^+ = 4v\cos^2(\phi/3) \quad \text{with } \cos(\phi) = u/v^{3/2} > 0.$$

Proof: For $x > 0$, the objective function $p(x)$ in (24) is differentiable and the first and second derivatives are

$$p'(x) = x - \omega - \alpha/\sqrt{x} \text{ and } p''(x) = 1 + \frac{\alpha}{2}x^{-3/2}.$$

It follows that $p'(x) < 0$ when $x > 0$ is close to 0 and $p''(x) \geq 1$ for all $x > 0$. Hence, $p(x)$ is decreasing near 0 and it is strongly convex on the half line $(0, +\infty)$. Therefore, the problem (24) has a unique solution and $x^+ > 0$. Moreover, we must have

$$p'(x^+) = x^+ - \omega - \frac{\alpha}{\sqrt{x^+}} = 0. \tag{25}$$

Introducing $y := \sqrt{x^+}$, we get

$$y^3 - \omega y - \alpha = 0, \tag{26}$$

which is known as the depressed cubic equation and has three roots (in the complex planes). However, we need to find the positive real root. For Case (i) $\tau \geq 0$, the positive root of (26) is given by the Cardan formula [38, Chp. 7] (the other two roots are complex)

$$\overline{y} = (u + \sqrt{\tau})^{1/3} + (u - \sqrt{\tau})^{1/3}.$$

and hence $x^+ = \overline{y}^2$ gives the solution in Case (i).

For Case (ii), $\tau < 0$ implies $v^3 > u^2$, which yields $v > 0$ (hence $\omega > 0$). Once again, by Cardan's formula, the cubic equation (26) has three real roots, namely $y_1 = 2\sqrt{v}\cos\phi/3$,

$$y_2 = 2\sqrt{v}\cos\left(\frac{\pi + \phi}{3}\right) \text{ and } y_3 = 2\sqrt{v}\cos\left(\frac{2\pi + \phi}{3}\right),$$

where $\cos(\phi) = u/v^{3/2}$ (a detailed proof for the above three roots can be found in [39] and a more assessable reference is [35]). Since $\cos(\phi) > 0$, it is easy to see that the only positive root is y_1. And $x^+ = y_1^2$ gives the result in Case (ii). ■

The above result shows that $x^+(\omega, \alpha) > 0$ whenever $\alpha > 0$. The next result states that it can be bounded away from 0 by a constant whenever ω and α satisfy certain bounds.

Proposition 3.4: Suppose there are two given constants $C > 0$ and $c > 0$. Then there exists $\gamma > 0$ such that

$$x^+(\omega, \alpha) \geq \gamma \ \ \forall\ (\omega, \alpha) \text{ staisfying } |\omega| \leq C, \ \alpha \geq c.$$

Proof: Suppose the result is not true. Then there exists a sequence $\{\omega_k, \alpha_k\}$, $k = 1, \ldots,$ with $|\omega_k| \leq C$ and $\alpha_k \geq c$ such that

$$\lim_{k\to\infty} x^+(\omega_k, \alpha_k) = 0.$$

By the proof in Prop. 3.3 (see (25)), $x^+(\omega_k, \alpha_k) > 0$ must be the solution of the following equation:

$$x^+(\omega_k, \alpha_k) - \omega_k - \alpha_k/\sqrt{x^+(\omega_k, \alpha_k)} = 0.$$

Multiplying $\sqrt{x^+(\omega_k, \alpha_k)}$ on the both sides of the equation above and taking limits, we get

$$\begin{aligned} 0 &= \lim_{k\to\infty} \left[(x^+(\omega_k, \alpha_k))^{3/2} - \omega_k\sqrt{x^+(\omega_k, \alpha_k)}\right] \\ &= \lim_{k\to\infty} \alpha_k \geq c > 0. \end{aligned}$$

The contradiction establishes the result claimed. ■

Prop. 3.3 can be readily extended to the case where the constraint is an interval instead of $x \geq 0$.

Proposition 3.5: Let B denote the interval $[a, b]$ in $\Re$ with $0 \leq a \leq b$. Let

$$x_B^+ := \arg\min_{x\in B}\ p(x) = \frac{1}{2}(x - \omega)^2 - 2\alpha\sqrt{x}. \tag{27}$$

Then we have

$$x_B^+ = \Pi_B(x^+) = \Pi_B(\texttt{dcroot}[\omega, \alpha]),$$

where $\Pi_B(x^+)$ denote the nearest point in B from x^+ and it is given by

$$\Pi_B(x^+) = \Pi_{[a,b]}(x^+) = \min\{b,\ \max\{a, x^+\}\}. \tag{28}$$

Proof: Prop. 3.3 showed that $x^+ > 0$ is the unique optimal solution of the problem (24) and $p'(x^+) = 0$. Since $p(x)$ is strongly convex over $x > 0$, this means that $p'(x) < 0$ for $x < x^+$ and $p'(x) > 0$ for $x > x^+$. We consider three cases. Case 1: $x^+ \in [a, b]$; Case 2: $x^+ > b$ and Case 3: $x^+ < a$.

For Case 1, it is obvious that x^+ is also the optimal solution of (27). Therefore, $x_B^+ = x^+$ and $x^+ = \Pi_B(x^+)$ because $x^+ \in B$. For Case 2, it follows that $p'(x) < 0$ for all $x \in [a, b]$. This means that $p(x)$ is strictly decreasing over the interval $[a, b]$. Hence, b is the optimal solution of (27) and $x_B^+ = b$. It is obvious that $b = \Pi_B(x^+)$ since $x^+ > b$. For Case 3, it follows that $p'(x) > 0$ for all $x \in [a, b]$. This means that $p(x)$ is strictly increasing over the interval $[a, b]$. Hence, a is the optimal solution of (27) and $x_B^+ = a$. It is obvious that $a = \Pi_B(x^+)$ since $x^+ < a$. For all three cases, we proved $x_B^+ = \Pi_B(x^+)$ as claimed in the result. ■

It follows from Prop. 3.3 and Prop. 3.5 that the optimal solution Z_{ij}^+ in (22) can be computed as follows:

$$Z_{ij}^+ = \Pi_{[L_{ij}, U_{ij}]}\Big(\texttt{dcroot}[(Z_\rho)_{ij}, W_{ij}\delta_{ij}/\rho]\Big). \tag{29}$$

Consequently, $Z^+ = \texttt{SQREDM}(W, \Delta, Z)$ in (23) is well defined and its elements can be computed by (29).

C. The Majorization Algorithm and Its Convergence

With the preparations above, we are ready to state our majorization algorithm. Let $D^k \in \mathcal{B}$ be the current iterate. We update it by solving the majorization subproblem (13) to get D^{k+1}. It follows from the solution of (20) with Z replaced by D^k that

$$D^{k+1} = \texttt{SQREDM}(W, \Delta, D^k_\rho) \tag{30}$$

with

$$D^k_\rho := -W/\rho - \Pi_{\mathcal{K}^n_+(r)}(-D^k). \tag{31}$$

It is easy to see that the update scheme falls within the general framework of majorization-minimization [16]. Moreover, our minimization problem has the closed-form formula (30) with (29) being a projection. Hence, it is a majorization-projection algorithm. Because it is based on the square-root EDM model (11), we refer to this matrix majorization-projection method as SQREDM, which is summarized in Alg.1.

Algorithm 1 SQREDM Method

1: **Input data:** Dissimilarity matrix Δ, weight matrix W, penalty parameter $\rho > 0$, lower-bound matrix L, upper-bound matrix U and the initial D^0. Set $k := 0$.
2: **Update:** Compute D^{k+1} by (30) and (31).
3: **Convergence check:** Set $k := k + 1$ and go to Step 2 until convergence.

Being a majorization update, (30) enjoys the commonly known majorization inequalities as follows: For $k = 1, 2, \ldots,$

$$\begin{aligned} F_\rho(D^k) &= f(D^k) + \rho g(D^k) = f(D^k) + \rho g_m(D^k, D^k) \\ &\overset{(13)}{\geq} f(D^{k+1}) + \rho g_m(D^{k+1}, D^k) \\ &\overset{(18)}{\geq} f(D^{k+1}) + \rho g(D^{k+1}) \\ &= F_\rho(D^{k+1}). \end{aligned} \tag{32}$$

The functional sequence $\{F_\rho(D^k)\}$ is non-increasing and converges because it is bounded from below by 0. This is also the type of convergence that is enjoyed by all majorization methods. However, we would like to establish stronger convergence on the iterates sequence $\{D^k\}$ itself.

A major obstacle in analysing the convergence for the square-root EDM model (11) is the non-differentiability of the objective function. Our first result below shows that the objective $f(\cdot)$ is actually differentiable along the generated sequence. We need the following two reasonable assumptions:
Assumption 1: The constrained box $\mathcal{B}$ is bounded.
Assumption 2: For Δ and U, we require $U_{ij} > 0$ if $\delta_{ij} > 0$.

Assumption 1 can be easily satisfied (e.g., setting the upper bound to be twice the largest δ^2_{ij}). Assumption 2 means that if $\delta_{ij} > 0$, then we certainly do not want the upper bound $U_{ij} = 0$; otherwise $L_{ij} = 0$ and the corresponding D_{ij} is forced to be 0, a very poor approximation to positive δ_{ij}. With these two assumptions, we are able to establish the differentiability of $f(\cdot)$ along the generated sequence.

Proposition 3.6: Suppose Assumptions 1 and 2 hold. Let $\{D^k\}$ be the sequence generated by Alg. 1. Then the following hold.

(i) $f(D)$ is continuously differentiable at D^k, $k = 1, 2, \ldots, .$
(ii) The sequence $\{D^k\}$ is bounded and $f(D)$ is continuously differentiable at any of its limits.

Proof: (i) We write $f(D)$ in terms of D_{ij}:

$$f(D) = \sum_{i,j} W_{ij} D_{ij} - 2 \sum_{i,j} W_{ij}\delta_{ij}\sqrt{D_{ij}} + \sum_{i,j} W_{ij}\delta^2_{ij}.$$

We will prove for any given pair (i, j), $\partial f(D)/\partial D_{ij}$ exists and is continuous at any point D^k. We consider two cases. Case 1: $W_{ij}\delta_{ij} = 0$. This implies $f(D)$ is a linear function of D_{ij} and $\partial f(D)/\partial D_{ij} = 2W_{ij}$ is constant and hence is continuous. Case 2: $W_{ij}\delta_{ij} > 0$. It follows from (29) and (30) that

$$D^k_{ij} = \Pi_{[L_{ij}, U_{ij}]}\Big(\texttt{dcroot}[(D^{k-1}_\rho)_{ij}, (W_{ij}\delta_{ij})/\rho]\Big).$$

Let $\alpha_{ij} := (W_{ij}\delta_{ij})/\rho > 0$ and $\omega^{k-1}_{ij} := (D^{k-1}_\rho)_{ij}$. It follows from Prop. 3.3 (because $\alpha_{ij} > 0$) that $(x^k_{ij})^+ := \texttt{dcroot}[\omega^{k-1}_{ij}, \alpha_{ij}] > 0$ and from (28) that

$$D^k_{ij} = \Pi_{[L_{ij}, U_{ij}]}((x^k_{ij})^+) \geq \min\{U_{ij}, x^+_{ij}\} > 0, \tag{33}$$

where the last inequality used the fact that $U_{ij} > 0$ because of $\delta_{ij} > 0$ by Assumption 2. It is obvious that

$$\frac{\partial f(D)}{\partial D_{ij}}\Big|_{D^k_{ij}} = W_{ij}(1 - \delta_{ij}/\sqrt{D^k_{ij}}),$$

which is continuous at $D^k_{ij} > 0$. This proved (i).

(ii) Since $\mathcal{B}$ is bounded (Assumption 1) and $D^k \in \mathcal{B}$, the sequence $\{D^k\}$ is bounded. Let $\widehat{D}$ be one of its limits. Without loss of any generality, let us assume $D^k \to \widehat{D}$. The proof below is the continuation in (i). For a given pair (i, j), if $W_{ij}\delta_{ij} = 0$, we have seen in (i) that $\partial f/\partial D_{ij}$ is a constant (independent of D^k). We only need to consider the case $W_{ij}\delta_{ij} > 0$, which implies $\delta_{ij} > 0$ and $U_{ij} > 0$ by Assumption 2.

It follows from (31) that there exists a constant $C > 0$ such that

$$\begin{aligned} |\omega^k_{ij}| = |(D^k_\rho)_{ij}| \leq \|D^k_\rho\| &\leq \|W\|/\rho + \|\Pi_{\mathcal{K}^n_+(r)}(-D^k)\| \\ &\leq \|W\|/\rho + 2\|D^k\| \leq C, \end{aligned}$$

where we used the boundedness of $\{D^k\}$ and Lemma 2.2. Prop. 3.4 implies that there exists $\gamma > 0$ such that $(x^k_{ij})^+ \geq \gamma$ for $k = 1, \ldots, .$. It follows from (33) that

$$D^k_{ij} \geq \min\{U_{ij},\ (x^k_{ij})^+\} \geq \min\{U_{ij}, \gamma\}.$$

Taking limit on the left-hand side, we get $\widehat{D}_{ij} \geq \min\{U_{ij}, \gamma\} > 0$. Hence, $\partial f(D)/\partial D_{ij}$ exists and is continuous at $\widehat{D}_{ij}$. This proved (ii). ∎

We are ready to state our main convergence result

Theorem 3.7: Let the function $F_\rho(D)$ be defined in (12) and let $\{D^k\}$ be the sequence generated by the SQREDM method.

(i) We have

$$F_\rho(D^{k+1}) - F_\rho(D^k) \leq -\frac{\rho}{2}\|D^{k+1} - D^k\|^2,\ k = 1, 2, \ldots.$$

Consequently, $\|D^{k+1} - D^k\| \to 0$.

(ii) Let $\widehat{D}$ be an accumulation point of $\{D^k\}$. Then for any $D \in \mathcal{B}$, we have

$$\langle \nabla f(\widehat{D}) + \rho\widehat{D} + \rho\Pi_{\mathcal{K}_+^n(r)}(-\widehat{D}),\ D - \widehat{D}\rangle \geq 0. \quad (34)$$

That is, $\widehat{D}$ is a stationary point of the problem (12). Moreover, for a given $\epsilon > 0$, if $D^0 \in \mathcal{K}_+^n(r) \cap \mathcal{B}$ and

$$\rho \geq \rho_\epsilon := \frac{f(D^0)}{\epsilon},$$

then $\widehat{D}$ is an ϵ-approximate KKT point of (11).

(iii) If $\widehat{D}$ is an isolated accumulation point of the sequence $\{D^k\}$, then the whole sequence $\{D^k\}$ converges to $\widehat{D}$.

Proof: (i) We are going to use the following facts that are stated on D^{k+1} and D^k. The first one is due to the convexity of $f(D)$

$$f(D^k) \geq f(D^{k+1}) + \langle \nabla f(D^{k+1}),\ D^k - D^{k+1}\rangle. \quad (35)$$

The second fact is the identity:

$$\|D^{k+1}\|^2 - \|D^k\|^2 = 2\langle D^{k+1} - D^k,\ D^{k+1}\rangle - \|D^{k+1} - D^k\|^2. \quad (36)$$

The third fact is due to the convexity of $h(D)$ (see Lemma 2.1(ii)):

$$h(-D^{k+1}) - h(-D^k) \geq \langle \Pi_{\mathcal{K}_+^n(r)}(-D^k),\ -D^{k+1} + D^k\rangle. \quad (37)$$

The last one is the optimality condition of the problem (13): for all $D \in \mathcal{B}$, we have

$$\langle \nabla f(D^{k+1}) + \rho D^{k+1} + \rho\Pi_{\mathcal{K}_+^n(r)}(-D^k),\ D - D^{k+1}\rangle \geq 0, \quad (38)$$

which is well-defined because we already established the differentiability of f at D^{k+1} (Prop. 3.6(i)) and the problem (13) is convex. Those four facts yield the following chain of inequalities:

$$\begin{aligned}
& F_\rho(D^{k+1}) - F_\rho(D^k) \\
= \ & f(D^{k+1}) - f(D^k) + \rho g(D^{k+1}) - \rho g(D^k) \\
\overset{(35)}{\leq} \ & \langle \nabla f(D^{k+1}),\ D^{k+1} - D^k\rangle + \rho g(D^{k+1}) - \rho g(D^k) \\
= \ & \langle \nabla f(D^{k+1}),\ D^{k+1} - D^k\rangle \\
+ \ & (\rho/2)(\|D^{k+1}\|^2 - \|D^k\|^2) - \rho[h(-D^{k+1}) - h(-D^k)] \\
\overset{(36)}{=} \ & \langle \nabla f(D^{k+1}) + \rho D^{k+1},\ D^{k+1} - D^k\rangle \\
& -(\rho/2)\|D^{k+1} - D^k\|^2 - \rho[h(-D^{k+1}) - h(-D^k)] \\
\overset{(37)}{\leq} \ & \langle \nabla f(D^{k+1}) + \rho D^{k+1} + \rho\Pi_{\mathcal{K}_+^n(r)}(-D^k),\ D^{k+1} - D^k\rangle \\
& -(\rho/2)\|D^{k+1} - D^k\|^2 \\
\overset{(38)}{\leq} \ & -(\rho/2)\|D^{k+1} - D^k\|^2.
\end{aligned}$$

This proves that the sequence $\{F_\rho(D^k)\}$ is non-increasing and it is also bounded below by 0. Taking the limits on both sides yields $\|D^{k+1} - D^k\| \to 0$.

(ii) Suppose $\widehat{D}$ is the limit of a subsequence $\{D^{k_\ell}\}$, $\ell = 1, \ldots,$. Since we have established in (i) that $(D^{k_\ell+1} - D^{k_\ell}) \to 0$, the sequence $\{D^{k_\ell+1}\}$ also converges to $\widehat{D}$. Now taking the limits on both sides of (38) on $\{k_\ell\}$, we reach the desired inequality (34). We now prove $\widehat{D}$ is an ϵ-approximate KKT point. It follows from Lemma 2.1(ii) and Lemma 2.2 that

$$\nabla f(\widehat{D}) + \rho\widehat{D} + \rho\Pi_{\mathcal{K}_+^n(r)}(-\widehat{D}) \in \partial\mathcal{L}(\widehat{D},\ \rho),$$

which is the condition (19) with $\widehat{\beta} = \rho$. We only need to show $g(\widehat{D}) \leq \epsilon$. Since $D^0 \in \mathcal{K}_+^n(r) \cap \mathcal{B}$, we have

$$\begin{aligned}
f(D^0) &= f(D^0) + \rho g(D^0) \quad (\text{because } g(D^0) = 0) \\
&\overset{(13)}{\geq} f(D^1) + \rho g_m(D^1, D^0) \quad (\text{because } D^0 \in \mathcal{B}) \\
&\overset{(18)}{\geq} f(D^1) + \rho g_m(D^1) \geq \cdots \\
&\overset{(32)}{\geq} f(D^k) + \rho g(D^k).
\end{aligned}$$

Taking the limit on the right-hand side yields

$$f(D^0) \geq f(\widehat{D}) + \rho g(\widehat{D}) \geq \rho g(\widehat{D}),$$

where we used $f(\widehat{D}) \geq 0$. Therefore, it has

$$g(\widehat{D}) \leq \frac{f(D^0)}{\rho} \leq \frac{f(D^0)}{\rho_\epsilon} = \epsilon.$$

We proved that $\widehat{D}$ is an ϵ-approximate KKT point of (11).

(iii) We note that we have proved in (i) that $(D^{k+1} - D^k) \to 0$. The convergence of the whole sequence to $\widehat{D}$ follows from [40, Prop. 7]. ■

We finish this section with two more remarks.

(R1) A direct consequence of Prop. 3.6 is that the objective $f(D)$ is continuously differentiable on the path $\mathcal{P} := \text{cl}\,(\cup_{k=1}^\infty \mathcal{P}_k)$, where $\text{cl}(\Omega)$ denotes the closure of a set Ω,

$$\mathcal{P}_k := \left\{D \mid D = \beta D^k + (1-\beta)D^{k-1},\ 0 \leq \beta \leq 1\right\}.$$

Moreover, $\mathcal{P}$ is bounded.

(R2) The continuous differentiability along the path $\mathcal{P}$ of the generated points saves us from making extensive use of subdifferential in nonsmooth optimization in order to prove the optimality result in Thm. 3.7.

IV. Numerical Experiments and Comparison

In this part, we will conduct extensive numerical experiments of our algorithm `SQREDM` using MATLAB (R2014a) on a desktop of 8GB memory and Inter(R) Core(TM) i5-4570 3.2Ghz CPU, against 4 leading solvers on the problems of SNL in two dimensions ($r = 2$) and Molecular Conformation (MC) in three dimensions ($r = 3$). Our conclusion is that `SQREDM` is very competitive and significantly exceeds the performance of all 4 solvers in many scenarios. For instance, the solution quality of `SQREDM` is comparable to the best results by the 4 solvers and the time used is only a small fraction of what was used by them. This section includes the following parts: Test problems, Implementation of `SQREDM`, Selection of benchmark methods and Numerical comparison.

A. Test Problems

We first describe our test problems so that our implementation and the selection of the benchmark methods may be related to them.

(a) SNL test problems. As pointed out in the Introduction, stress minimization coincides with the maximum likelihood principle in SNL if the observed ranges among sensors are perturbed by the white noise. Hence, SNL has been widely used to test the viability of the proposed methods for stress minimization (e.g., [15]). In such a problem, we typically have m anchors (e.g., sensors with known locations) and the rest sensors need to be located. We use two examples for our test. One has a regular network topology and the other is non-regular.

Example 4.1: (Square Network) This example is widely tested since its detailed study in [41]. In the square region $[-0.5, 0.5]^2$, 4 anchors $\mathbf{x}_1 = \mathbf{a}_1, \cdots, \mathbf{x}_4 = \mathbf{a}_4$ ($m = 4$) are placed at $(\pm 0.2, \pm 0.2)$. The generation of the rest $(n - m)$ sensors $(\mathbf{x}_{m+1}, \cdots, \mathbf{x}_n)$ follows the uniform distribution over the square region. The noisy Δ is usually generated as follows.

$$\begin{aligned} \delta_{ij} &:= \|\mathbf{x}_i - \mathbf{x}_j\| \times |1 + \epsilon_{ij} \times \mathtt{nf}|, \ \forall\ (i,j) \in \mathcal{N} \\ \mathcal{N} &:= \mathcal{N}_x \cup \mathcal{N}_a \\ \mathcal{N}_x &:= \{(i,j) \mid \|\mathbf{x}_i - \mathbf{x}_j\| \le R,\ i > j > m\} \\ \mathcal{N}_a &:= \{(i,j) \mid \|\mathbf{x}_i - \mathbf{a}_j\| \le R,\ i > m,\ 1 \le j \le m\}, \end{aligned}$$

where R is known as the radio range, ϵ_{ij}'s are independent standard normal random variables, and $\mathtt{nf}$ is the noise factor (e.g., $\mathtt{nf} = 0.1$ was used in the tests and it corresponds to 10% noise level). In literature (e.g., [41]), this type of perturbation in δ_{ij} is known to be multiplicative and follows the unit-ball rule in defining $\mathcal{N}_x$ and $\mathcal{N}_a$ (see [42, Sect. 3.1] for more detail). The corresponding weight matrix W and the lower and upper bound matrices L and U are given as in the table below. Here, M is a large positive quantity. For example, $M := n \max_{ij} \Delta_{ij}$ is the upper bound of the longest shortest path if the network is viewed as a graph.

(i,j)	W_{ij}	Δ_{ij}	L_{ij}	U_{ij}
$i = j$	0	0	0	0
$i, j \le m$	0	0	$\|\mathbf{a}_i - \mathbf{a}_j\|^2$	$\|\mathbf{a}_i - \mathbf{a}_j\|^2$
$(i,j) \in \mathcal{N}$	1	δ_{ij}	0	R^2
otherwise	0	0	R^2	M^2

Example 4.2: (EDM word network) This problem has a non-regular topology and is first used in [42] to challenge existing localization methods. In this example, n points are randomly generated in a region whose shape is similar to the letters "E", "D" and "M". The ground truth network is depicted in Fig. 1. We choose the first m points to be the anchors. The rest of the data generation is same as in Example 4.1.

(b) MC test problems. Molecular conformation has long been an important application of EDM optimization [6]. We collected real data of 12 molecules derived from 12 structures of proteins from the Protein Data Bank (PDB) [43]. They are `1GM2`, `304D`, `1PBM`, `2MSJ`, `1AU6`, `1LFB`, `104D`, `1PHT`, `1POA`, `1AX8`, `1RGS`, `2CLJ`. They provide a good set of test problems in terms of the size n, which ranges from a few hundreds to a few thousands (the smallest $n = 166$ for `1GM` and the largest $n = 4189$ for `2CLJ`). The distance information was obtained in a realistic way as done in [44] and is described in the following example.

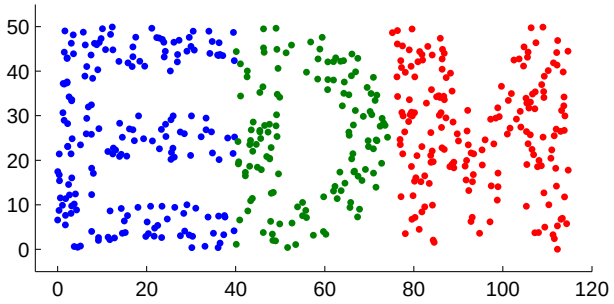


Fig. 1: Ground truth EDM network with $n = 500$ nodes.

Example 4.3: (Real PDB data) Each molecule comprises n atoms $\{\mathbf{x}_1, \ldots \mathbf{x}_n\}$ in $\Re^3$ and its distance information is collected as follows. If the Euclidean distance between two of the atoms is less than R, the distance is chosen; otherwise no distance information about this pair is known. For example, $R = 6\mathring{A}$ ($1\mathring{A} = 10^{-8}$cm) is nearly the maximal distance that the nuclear magnetic resonance (NMR) experiment can measure between two atoms. For realistic molecular conformation problems, not all the distances below R are known from NMR experiments, so one may obtain $c\%$ (e.g., $c = 50\%$) of all the distances below R. Denote $\mathcal{N}_x$ the set formed by indices of those measured distances. Moreover, the distances in $\mathcal{N}_x$ can not be exactly measured. Instead, only lower bounds ℓ_{ij} and upper bounds u_{ij} are provided, that is for $(i,j) \in \mathcal{N}_x$,

$$\begin{aligned} \ell_{ij} &= \max\{1, (1 - |\epsilon_{ij}|)\|\mathbf{x}_i - \mathbf{x}_j\|\}, \\ u_{ij} &= (1 + |\varepsilon_{ij}|)\|\mathbf{x}_i - \mathbf{x}_j\|. \end{aligned}$$

where $\epsilon_{ij} \sim N(0, \mathtt{nf}^2 \times \pi/2)$ are independent normal random variables. In our test, we set the noise factor $\mathtt{nf} = 0.1$ and the parameters $W, \Delta, L, U \in \mathcal{S}^n$ are given as in the table below, where $M > 0$ is the upper bound (e.g., $M := n \max_{ij} \Delta_{ij}$).

(i,j)	W_{ij}	Δ_{ij}	L_{ij}	U_{ij}
$i = j$	0	0	0	0
$(i,j) \in \mathcal{N}_x$	1	$(a_{ij} + b_{ij})/2$	a_{ij}^2	b_{ij}^2
otherwise	0	0	0	M^2

B. Implementation

The `SQREDM` Alg. 1 is easy to implement. For its input, we already defined Δ, L and U matrices for the test problems. For the initial point, we follow the popular choice used in [7], [8] $\sqrt{D^0} := \widehat{\Delta}$, where $\widehat{\Delta}$ is the matrix obtained by the shortest path distances among Δ. If Δ has no missing values, then $\widehat{\Delta} = \Delta$. We now address the remaining issues that are the stopping criterion and choice of the penalty parameter ρ.

(c) Stopping criterion. It follows from Thm. 3.7 that the objective sequence $\{F_\rho(D^k)\}$ is non-increasing. We define the relative progress in F_ρ by

$$\mathtt{Fprog}_k := \frac{F_\rho(D^{k-1}) - F_\rho(D^k)}{1 + F_\rho(D^{k-1})}.$$

Having less progress alone in F_ρ is not enough to terminate the algorithm. We will also need to ensure that the current iterate D^k is close to $\mathcal{K}_+^n(r)$. It follows from (16) that

$$\begin{aligned}\texttt{Kprog}_k &:= \frac{2g(D^k)}{\|JD^kJ\|^2} = \frac{\|D^k + \Pi_{\mathcal{K}_+^n(r)}(-D^k)\|^2}{\|JD^kJ\|^2} \\ &= \frac{\|\mathrm{PCA}_r^+(-JD^kJ) + (JD^kJ)\|^2}{\|JD^kJ\|^2} \\ &= 1 - \frac{\sum_{i=1}^r \left[\lambda_i^2 - (\lambda_i - \max\{\lambda_i, 0\})^2\right]}{\lambda_1^2 + \ldots + \lambda_n^2} \\ &\leq 1,\end{aligned}$$

where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ are the eigenvalues of $(-JD^kJ)$. The smaller $\texttt{Kproj}_k$ is, the closer D^k is to $\mathcal{K}_+^n(r)$. The benefit of using Kprog over $g(D)$ is that the former is independent of any scaling of D. We terminate SQREDM when

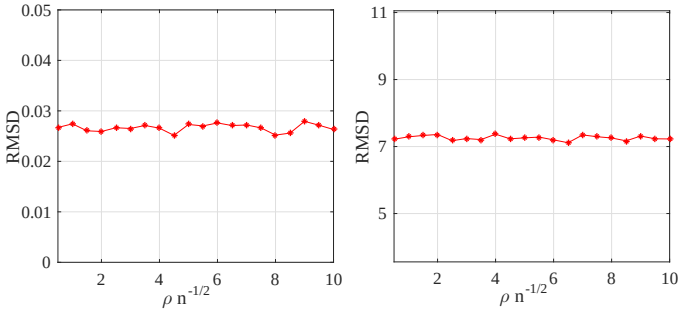
$$\texttt{Fprog}_k \leq \sqrt{n}10^{-5} \quad \text{and} \quad \texttt{Kprog}_k \leq 10^{-3}.$$

(d) Measuring the solution quality. For this purpose, we adopt a widely used measure RMSD (Root of the Mean Squared Deviation) defined by

$$\texttt{RMSD} := \left[\frac{1}{n-m}\sum_{i=m+1}^{n} \|\widehat{\mathbf{x}}_i - \mathbf{x}_i\|^2\right]^{1/2},$$

where $\mathbf{x}_i$'s are the true positions of the sensors in our test problems and $\widehat{\mathbf{x}}_i$'s are their corresponding estimates. The $\widehat{\mathbf{x}}_i$'s were obtained by applying cMDS to the final output of the distance matrix, followed by aligning them to the existing anchors through the well-known Procrustes procedure (see [9], [3, Chp. 20] or [50, Prop. 4.1] for more details). Furthermore, upon obtaining $\widehat{\mathbf{x}}_i$'s, a heuristic gradient method can be applied to improve their accuracy and it is called the refinement step in [41]. We report rRMSD to highlight its contribution. As we will see, all tested methods benefit from this step, but with varying degrees.

(e) Choice of the penalty parameter. In principle, the penalty parameter ρ should start from a small value and is then eventually increased in a way that should depend on the latest progress made (see [34, P. 495]). The optimal choice of ρ is dependent upon the size and geometry of the network and the distance information available.



(a) Example 4.1 (b) Example 4.2

Fig. 2: RMSD error of the estimated positions obtained from SQREDM with penalty parameter ρ: (a) Example 4.1 ($n = 200$, $R = 0.2$). (b) Example 4.2 ($n = 200$, $m = 20$, $R = 10$).

To see the dependence of our method on the penalty parameter ρ, we tested it on Example 4.1 (regular layout) and Example 4.2 (irregular layout) with varying ρ such that $\rho/\sqrt{n} \in \{0.5, 1, 1.5, 2, \ldots, 10\}$. Now under a given ρ, we ran each test instance 20 times and recorded the average RMSD. A plot of RMSD vs $\rho/\sqrt{n}$ for the two examples can be found in Figure 2. A pleasing feature is that the plot closely follows a straight line in both cases. This means that SQREDM is quite robust to the change of ρ when it is in the order of $\sqrt{n}$. In our implementation, we fixed ρ and used $\rho = \sqrt{n}$.

C. Selection of benchmark methods

(f) On some simple majorization methods. We first demonstrate how a simple majorization method (SMM) that falls in the framework of (3) with (4) and (5) works. It is suggested by a referee and is implied by the framework studied in [14]. The quadratic function in (4) and the linear function in (5) are respectively given by

$$\begin{aligned}q(\mathbf{x}_i, \mathbf{x}_j, X^k) &= 2\|\mathbf{x}_i - \mathbf{x}_i^k\|^2 + 2\langle \mathbf{d}_{ij}^k, \mathbf{x}_i - \mathbf{x}_i^k\rangle + \\ &\quad 2\|\mathbf{x}_j - \mathbf{x}_j^k\|^2 - 2\langle \mathbf{d}_{ij}^k, \mathbf{x}_j - \mathbf{x}_j^k\rangle + \|\mathbf{d}_{ij}^k\|^2\end{aligned}$$

where $\mathbf{d}_{ij}^k := \mathbf{x}_i^k - \mathbf{x}_j^k$ and for $\mathbf{x}_i^k \neq \mathbf{x}_j^k$,

$$\begin{aligned}&l(\mathbf{x}_i, \mathbf{x}_j, X^k) \\ &= -\|\mathbf{d}_{ij}^k\|^{-1}(\langle \mathbf{d}_{ij}^k, \mathbf{x}_i - \mathbf{x}_i^k\rangle - \langle \mathbf{d}_{ij}^k, \mathbf{x}_j - \mathbf{x}_j^k\rangle) - \|\mathbf{d}_{ij}^k\|\end{aligned}$$

We note that the quadratic function $q(\mathbf{x}_i, \mathbf{x}_j, X^k)$ does not have any coupled terms between $\mathbf{x}_i$ and $\mathbf{x}_j$. Hence the individual majorization function $m_i(\mathbf{x}_i, X^k)$ in (3) can be constructed through those quadratic and linear functions.

We also note that Soares, Xavier and Gomes developed two other important "simple" majorization methods, respectively referred to as SMLL (Stable Maximum-Likelihood Localization) [18] and diskRelax [15]. As pointed out in [18, Sect. V], SMLL "*receives an initialization from a convex approximation method. The initialization will hopefully hand to nonconvex refinement algorithms a point near the basin of attraction of the true minimum.*" However, our choice of the initialization (if not provided by a package) is the embedding by cMDS, which is cheap to compute and commonly used. It appears that cMDS initialization is not good enough for SMLL for many tested cases in this paper. We therefore will not compare it with SQREDM in our experiments. One common and nice feature of those methods is that they are free from tuning any algorithmic parameters.

The performance of SMM and diskRelax (with comparison to SQREDM) was demonstrated on Examples 4.1 with varying ranges R. For diskRelax, we set MAXITER $= 10^4$ and epsilon $= 10^{-3}$. The setting for SMM was same as those for SMACOF. The initial point for both methods was the cMDS embedding. We ran each test instance 20 times and recorded their average rRMSD. The reason for reporting rRMSD is that the refinement step significantly improved the solution quality for both methods. The results were plotted in Fig. 3. It can be observed that both SMM and diskRelax returned high quality embedding only when the radio range was sufficiently large (e.g., $R \geq 0.8$ for diskRelax and

$R \geq 1.2$ for SMM.) This essentially means that only a small number of dissimilarities δ_{ij} are not known. In contrast, SQREDM worked well also for small ranges (e.g., $R = 0.2$). We also note that the linear function $l(\mathbf{x}_i, \mathbf{x}_j, X^k)$ is poorly scaled when $\mathbf{x}_i^k$ and $\mathbf{x}_j^k$ are close to each other and it is even not well-defined when $\mathbf{x}_i^k = \mathbf{x}_j^k$ because it involves the term $1/\|\mathbf{x}_i^k - \mathbf{x}_j^k\|$. Moreover, this drawback would create difficulties in establishing convergences of SMM on the iterates $\{X^k\}$. On the time consumed, SMM was the fastest, while diskRelax took proportionally significantly more time to terminate. Furthermore, this test problem is moderate in size ($n = 100$) when compared to our tested problems below with n ranging from a few hundreds to a few thousands. Our experiments showed that they only worked for a small number of our tested problems. It was pointed out to us by one of its authors [15] that diskRelax tends to work well for networks whose unknown sensors lie on the convex hull of some anchors. However, both Examples 4.1 and 4.2 do not meet this assumption. Therefore, we will not include the two methods in our further numerical experiments.

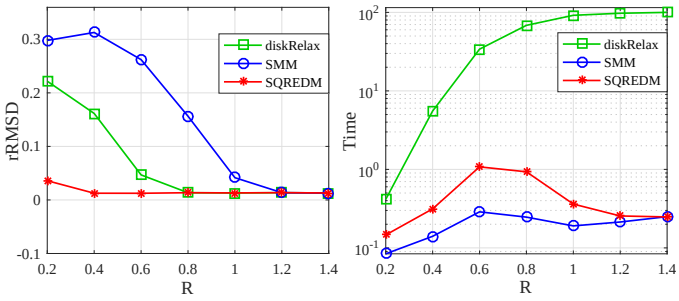


Fig. 3: Performance of three majorization methods: diskRelax, SMM and SQREDM on Example 4.1 with $n = 100$ and varying radio ranges R. Left: rRMSD error of the estimated positions. Right: Time consumed.

(g) Four benchmark methods and their computational complexities. Out of many published methods, we select four representative state-of-the-art methods for comparison due to their high-quality code implementation and availability. Those methods have been shown to be capable of returning satisfactory localization/embedding in many applications. Those methods are SMACOF [4] whose MATLAB implementation is taken from [36]; ARAP [9], ADMMSNL [17], and SFSDP [31]. SMACOF is a traditional method for the stress minimization and has a high reputation in experimental sciences [3]. ARAP can yield satisfactory embedding especially when the noisy factor nf is small. ADMMSNL is motivated by [15] and aims to enhance the package diskRelax of [15] for the SNL problems ($r = 2$). Its current implementation does not support the embedding for $r \geq 3$. All the three methods are for the stress minimization problem (1). However, SFSDP is developed for the problem (7). We include it because SFSDP is a high-level MATLAB implementation of the SDP approach and is capable of solving large scale problems with high-quality embedding. It truly serves as a benchmark method for any embedding algorithms.

In our tests, we used all of their default parameters except one or two in order to achieve the best results. In particular, for ARAP, tol $= 10^{-2}$ and IterNum $= 40$ to speed up the termination. For SFSDP we set pars.SDPsolver = "sedumi" because it returns the best overall performance. For SMACOF, we set rtol $= 10^{-2}$, iter $= 10^3$ and its initial point was the embedding by cMDS on Δ. ADMMSNL used the same setting for SMACOF.

We briefly discuss the computational complexity of those methods. For SMACOF, the update formula [3, Eq. (8.28]) is

$$(X^{k+1})^T = V^- B(X^k)(X^k)^T, \tag{39}$$

where V is an $n \times n$ matrix solely dependent on the weight matrix W and V^- is the Moore-Penrose inverse (only calculated once), $B(X^k)$ can be obtained using about $(3/2)n(n-1)r$ operations (see [3, Eq. (8,24)]). The data matrix X^k is of $r \times n$ and (39) involves matrices of $n \times n$ multiplying an $n \times r$ matrix. Hence, the total complexity of SMACOF is $O(rn^2)$ per iteration. As emphasized in [9, P. 35:14], the overall complexity of ARAP is $O(nk^3)$ with k being the average number of neighbours of the nodes. If k is about $\sqrt{n}$, then the overall complexity would be about $O(n^{2.5})$. This may justify why ARAP used much time to terminate in some of our test problems reported below.

The computational complexity of ADMMSNL at each node i is analysed in [17, Sect. V] and is primarily dominated by solving a nonlinear optimization problem of size $r(1 + N_i)$, where N_i is the size of the neighbourhood of i. This nonlinear optimization can be simplified and solved by standard optimization methods such as Newton's method, which makes use of gradient and Hessian information. SFSDP uses the SDP solver "sedumi" whose complexity is $O(s^2\kappa^{2.5} + \kappa^{3.5})$ where s is the number of decision variables and κ the number of rows of the linear matrix inequality constraints. This is in addition to some computational techniques that exploit the sparsity properties in the linear equations encountered. Since our computation each iteration is dominated by $\Pi_{\mathcal{K}_+^n(r)}(-D)$ in the construction of the majorization function g_m in (18), the overall computational complexity of SQREDM is about $O(rn^2)$ (we used MATLAB's built-in function eigs.m to compute $\mathrm{PCA}_r^+(A)$ in (15)).

D. Numerical Comparison

In this part, we report extensive numerical results on the three examples, which in total have 14 problems. In each test case, we randomly generate 20 samples (set rng('shuffle') in Matlab) and the reported results are the average on them. For instance, if we were to test the case $n = 200, R = 0.2$ in Example 4.1, we would have generated 20 such networks in the way described in the example. This subsection includes three parts, signposted by (h), (i), (j), which are respectively for the three examples with varying size n.

(h) Comparison on Example 4.1 ($200 \leq n \leq 2000$). The quality of the general performance of the five methods can be better appreciated through visualizing their key indicators (RMSD, rRMSD, and the CPU time consumed). For this purpose, we tested Example 4.1 with a moderate size $n = 200$

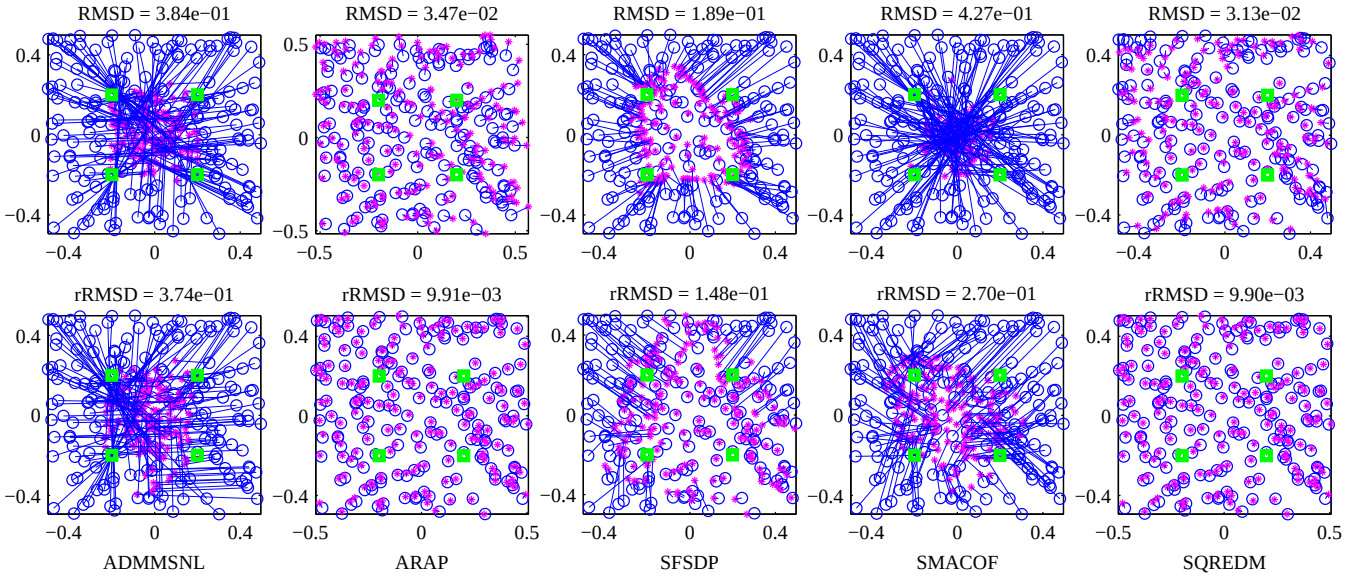


Fig. 4: Localization by the five methods for Example 4.1 with $n = 200, R = 0.2$.

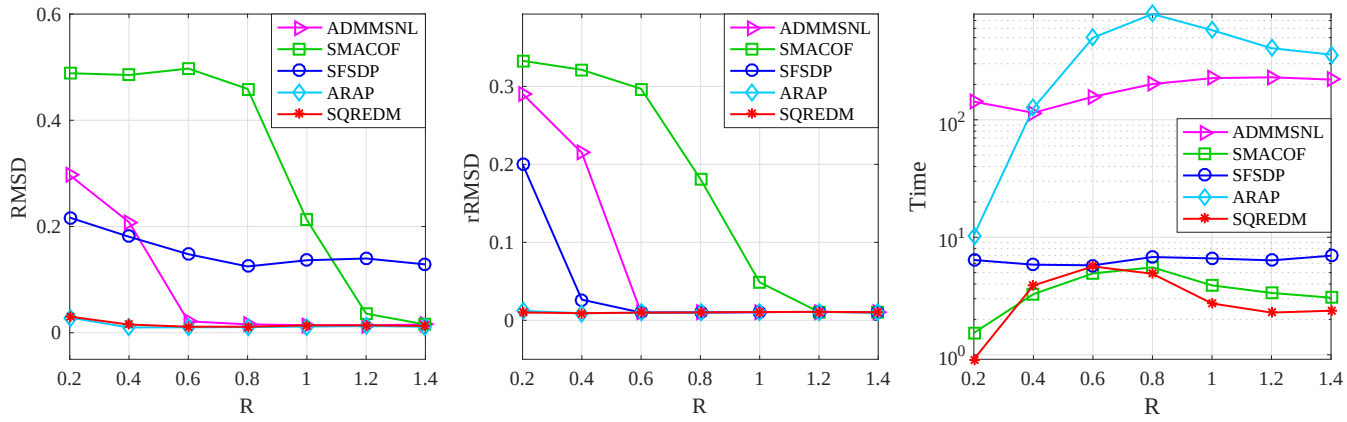


Fig. 5: Comparison of five methods for Example 4.1 with $n = 200, R = 0.2$.

and $R = 0.2$, which rendered many missing values in Δ. The actual embedding by each method was shown in Fig. 4, where the four anchors were plotted in green square and $\widehat{\mathbf{x}}_i$ in pink points were jointed to its true location (blue circle). It can be visibly seen that the clear winners are ARAP and SQREDM, followed by SFSDP, SMACOF and ADMMSNL. Clearly, there exist a number of miss-placed sensors by SMACOF, SFSDP and ADMMSNL both before and after the refinement step.

As expected, the performance for all the methods improves as R increases from 0.2 to 1.4. This is because we have more distance information in Δ as the radio range gets bigger. The results on the three indicators were plotted in Fig. 5. It can be seen that ARAP and SQREDM are again joint winners in terms of both RMSD and rRMSD. However, the time used by ARAP is the longest. This comes as no surprise because its complexity depends on the cubic of the average node degree k. As R increases, k increases as well. When R gets bigger than 0.6, both ADMMSNL and SFSDP produced similar rRMSD as ARAP and SQREDM, while the time consumed by ADMMSNL is significantly larger than that by SFSDP and SQREDM. We note that SQREDM used about 5 seconds in all cases and the time by SFSDP is just below 10 seconds. However, as we will see, SQREDM scales well when n gets larger, while SFSDP scales badly when n reaches a few thousands. This is demonstrated below.

We tested 8 problems with $n = 400, 500, 1000, 2000$ and $R = 0.2, \sqrt{2}$ respectively. We ran each problem 20 times and recorded average results in Table I, where SD is the standard deviation of RMSD. When $R = \sqrt{2}$, Δ has no missing values (since the sensors are restricted to a unit square region). For this case, all methods worked satisfactorily with rRMSD in the order of 10^{-3}. We note that SQREDM and ARAP benefited little from the refinement step because their RMSD are already in the order of 10^{-3}. Furthermore, SQREDM used only a fraction of cpu time consumed by other methods. When $R = 0.2$, Δ has many missing values and hence it is sparse. The picture is significantly different. RMSD by SQREDM and ARAP are in the order of 10^{-2}, while both RMSD and rRMSD by ADMMSNL and SFSDP are in the order of 10^{-1}, which is in the order of the unit region. Therefore, SQREDM generated

the most accurate results and used the least time (e.g., for $n = 2000, R = \sqrt{2}$, 33s (SQREDM) vs 4019s (SFSDP)).

(i) Comparison on Example 4.2 ($400 \leq n \leq 2000$). The purpose of testing this example is to see how those methods behave for networks with irregular layout. In this test, we fix the radio range $R = 10$, which generated Δ with many missing elements (i.e., Δ is sparse). For the visualization purpose, we plotted the results after the refinement step for the case of $n = 500$ and $m = 20, 40, 60$. As shown in Fig. 6, the black points were anchors and the rest were sensors. Compared with the shape of the ground truth EDM network in Fig. 1, the letters 'E', 'D' generated by ADMMSNL, SMACOF and SFSDP became clearer as m increased, but 'M' was still deformed. ARAP well captured the shapes of the three letters when $m = 20$ but got a slightly deformed 'M' for $m = 40$. By contrast, SQREDM was capable of capturing the shapes of the three letters for both cases.

Next, we tested 8 problems with $n = 400, 500, 1000, 2000$ and $m = 20, 40$ respectively. Also, each problem was run 20 times with the averaged results being reported in Table II, where it is easily observed that SQREDM always generated the lowest rRMSD. In terms of computational speed, SQREDM is the fastest and only used a fraction of the cpu consumed by other methods.

(j) Comparison on Example 4.3 ($166 \leq n \leq 4189$). These 12 problems represent a very challenging set of embedding problems in three dimensions ($r = 3$) because of the three reasons. One is that the size n ranges from hundreds to a few thousands. The second reason is that the dissimilarity matrix Δ is very sparse and the third reason is that the lower and upper bounds ℓ_{ij} and u_{ij} for $(i, j) \in \mathcal{N}_x$ have to be physically satisfied due to the properties of the atoms involved. Any violation of such box constraints would lead to certain level of deformation in the final embedding. Our method has a unique advantage in that it always obeys those box constraints, while others may not. Furthermore, both ADMMSNL and ARAP are purposely designed for SNL problems (i.e., $r = 2$). Their current implementations do not support the case $r = 3$. Hence, we have to exclude those two methods from our comparison.

In our test, we fixed $R = 6$, $c = 50\%$ and nf= 0.1. The generated embeddings by the remaining three methods for the two molecules 1GM2 and 1LFB were shown in Fig. 7, where the true and estimated positions of the atoms were plotted by blue circles and pink stars respectively. Each pink star was linked to its corresponding blue circle by a pink line. For both cases, SQREDM almost conformed the shape of the original data. Clearly, the other two failed to conform. The complete numerical results for the 12 problems were reported in Table III. It can be clearly seen that SQREDM performed significantly better in all three indicators: RMSD, rRMSD and Time. In particular, the time used by SQREDM is just a small fraction of that by the other two. For example, SQREDM only used 36.83s for 2CLJ, which is a very large data set with $n = 4189$. We feel that the significance of our proposed method in terms of the solution quality and the speed has been well demonstrated through this class of problems.

V. Conclusion

It is known that existing methods such as SMACOF and SDP relaxations for the stress minimization do not work satisfactorily in the context of SNL problems. In this paper, we considered the stress criterion under box constraints. The key concept used is the EDM cone with rank-r cut, which governs how well a dissimilarity matrix can be approximated by a true EDM with low-embedding dimensions. Based on this geometric concept, we developed a very fast algorithm, whose major computation for each step is from computing a few largest eigenvalues of a symmetric matrix (and the corresponding eigenvectors). Hence, the overall computational complexity of each step is $O(rn^2)$. We further established its theoretical convergence to a stationary point. One significant result is that the algorithm follows a smooth path despite the objective function is not everywhere differentiable. This result has led to a neat and water-tight convergence analysis. The performance of the proposed algorithm has been demonstrated against a few leading algorithms both SNL and MC problems. Based on our extensive numerical experiments, it is safe to say that SQREDM is capable of producing embeddings comparable to the best results by the tested algorithms, but only uses a small fraction of their computing time. In particular, our algorithm is potentially very useful and competitive for large scale embedding problems.

The proposed model and the algorithm has a wider applications other than SNL and MC problems. For example, it could be applied to image data for dimensionality reduction as done in [7] and problems studied in [45], [46]. It also remains to be seen whether the developed techniques can be used for the variants of the stress function considered in [3] and for outlier removal in the robust MDS [47]–[49]. We plan to investigate those problems in near future.

Acknowledgement

We sincerely thank the associate editor and the four referees for their constructive comments, which have significantly improved the quality of the paper. We are also very grateful to Dr Cláudia Soares for her timely support in using the packages SMLL and diskRelax, and to Dr Nicola Piovesan for sharing with us his excellent package ADMMSNL. We thank Dr Shuanghua Bai at NAG (Numerical Algorithms Group) for his help in testing the benchmark methods. This work is supported by the National Natural Science Foundation of China (11728101,71611130218).

References

[1] J.B. Kruskal. "Nonmetric multidimensional scaling: a numerical method", *Psychometrika*, 29, pp. 115-129, 1964.

[2] T.F. Cox and M.A.A. Cox, *Multidimensional Scaling*, 2nd Ed, Chapman and Hall/CRC, 2001.

[3] I. Borg and P.J.F. Groenen, *Modern Multidimensional Scaling: Theory and Applications*, 2nd Ed., Springer Series in Statistics, Springer, 2005.

[4] J. de Leeuw, "Applications of Convex Analysis to Multidimensional Scaling", In J Barra, F Brodeau, G Romier, B van Cutsem (eds.), *Recent Developments in Statistics*, pp. 133–145. North Holland Publishing Company, Amsterdam, The Netherlands, 1977.

[5] J. de Leeuw and P. Mair, "Multidimensional scaling using majorization: Smacof in R", *J. Stat. Software*, 31, pp. 1-30, 2009.

TABLE I: Comparisons of five methods for Example 4.1: rTime: cpu (in seconds) by the refinement step; Time: total cpu by each method including rTime (We omitted the results of ADMMSNL when $R = \sqrt{2}, n \geq 400$ since it made our desktop run out of memory, and omitted the results of ARAP when $R = \sqrt{2}, n \geq 1000$ since it consumed over 10 hours).

		$R = \sqrt{2}$					$R = 0.2$				
n		ADMMSNL	ARAP	SFSDP	SMACOF	SQREDM	ADMMSNL	ARAP	SFSDP	SMACOF	SQREDM
400	RMSD	1.33e-2	8.00e-3	1.16e-1	1.37e-2	8.89e-3	3.64e-1	9.21e-3	1.89e-1	4.84e-1	2.10e-2
	rRMSD	7.65e-3	7.55e-3	7.80e-3	7.54e-3	7.54e-3	3.47e-1	5.94e-3	1.46e-1	3.07e-1	6.02e-3
	SD	5.38e-3	9.07e-4	4.65e-3	1.30e-3	1.00e-3	1.55e-2	1.30e-3	1.37e-2	8.02e-2	2.31e-3
	rTime	5.53	0.84	6.62	0.90	1.67	0.64	0.28	0.51	0.63	0.34
	Time	440.67	2319.24	11.31	24.69	3.96	124.00	83.79	5.89	4.66	0.92
500	RMSD	−−	6.40e-3	1.19e-1	1.26e-2	7.05e-3	3.59e-1	7.47e-3	1.89e-1	4.57e-1	1.96e-2
	rRMSD	−−	5.83e-3	5.82e-3	5.85e-3	5.82e-3	3.38e-1	5.34e-3	1.61e-1	3.12e-1	5.22e-3
	SD	−−	4.42e-4	4.13e-3	8.18e-4	3.68e-4	1.16e-2	1.37e-3	1.57e-2	3.67e-2	2.31e-3
	rTime	−−	1.92	14.64	2.41	0.83	0.94	0.54	0.88	0.94	0.62
	Time	−−	4478.66	20.71	51.94	4.71	161.34	197.44	7.51	8.53	1.47
1000	RMSD	−−	−−	1.18e-1	1.08e-2	5.83e-3	3.43e-1	4.14e-3	1.80e-1	4.44e-1	1.03e-2
	rRMSD	−−	−−	5.68e-3	4.27e-3	4.23e-3	3.21e-1	3.30e-3	1.15e-1	2.99e-1	3.24e-3
	SD	−−	−−	2.87e-3	6.59e-4	2.01e-4	7.11e-3	1.05e-3	1.90e-3	4.11e-3	9.21e-4
	rTime	−−	−−	41.70	4.70	7.23	4.03	1.28	3.71	4.53	4.04
	Time	−−	−−	60.63	496.82	14.56	404.93	3118.40	22.26	7.61	6.48
2000	RMSD	−−	−−	1.24e-1	1.00e-2	4.73e-3	3.47e-1	3.72e-3	1.86e-1	4.35e-1	9.05e-3
	rRMSD	−−	−−	1.17e-2	3.87e-3	3.02e-3	2.87e-1	2.46e-3	1.59e-1	2.98e-1	2.40e-3
	SD	−−	−−	1.71e-3	4.11e-4	1.02e-4	2.11e-3	3.95e-4	1.08e-3	1.01e-3	3.21e-4
	rTime	−−	−−	314.14	20.00	17.71	41.71	6.24	29.85	21.20	17.56
	Time	−−	−−	426.89	4019.35	32.63	1297.91	32192.35	78.10	63.60	36.12

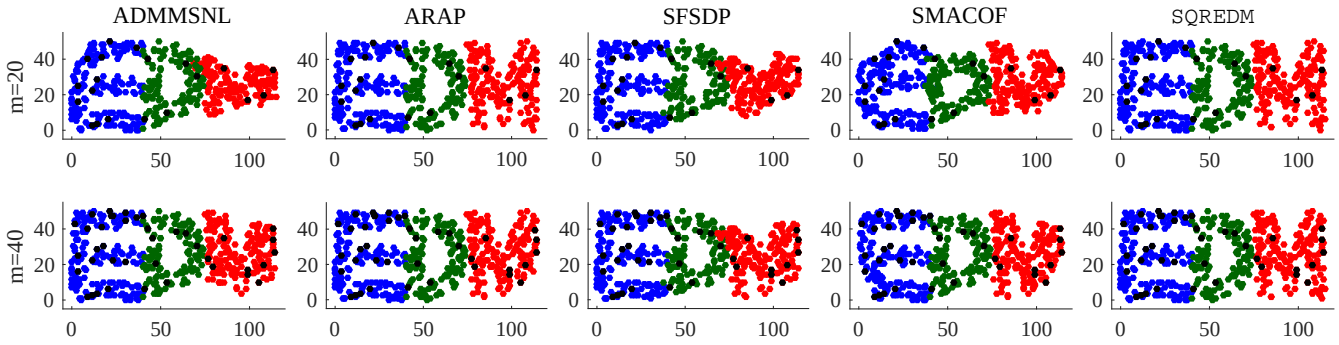


Fig. 6: Localization for Example 4.2 with $n = 500, R = 10$.

[6] W. Glunt, T.L. Hayden and R. Raydan, "Molecular conformations from distance matrices", *J. Comput. Chemistry*, 14, pp. 114-120, 1993.

[7] J.B. Tenenbaum, V. de Silva and J.C. Langford, "A global geometric framework for nonlinear dimensionality reduction", *Science*, 290, pp. 2319-2323, 2000.

[8] Y. Shang, W. Ruml, Y. Zhang and M.P.J. Fromherz, "Localization from mere connectivity", in: *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing*, MobiHoc 03, ACM, New York, NY, USA, pp.201-212, 2003.

[9] L. Zhang, L. Liu, C. Gotsman and S.J. Gortler, "An as-rigid-as-possible approach to sensor network localization", *ACM Trans. Sen. Netw.*, 6(4), pp. 35:1-35:21, 2010.

[10] I.J. Schoenberg, "Remarks to Maurice Frechet's article Sur la definition axiomatque d'une classe d'espaces vectoriels distancies applicbles vectoriellement sur l'espace de Hilbet", *Ann. Math.*, 36, pp. 724-732, 1935.

[11] G. Young and A.S. Householder, "Discussion of a set of points in terms of their mutual distances", *Psychometrika*, 3, pp. 19-22, 1938.

[12] W.S. Torgerson, "Multidimensional scaling: I. Theory and method", *Psychometrika*, 17(4), pp. 401-419, 1952.

[13] A. Karbasi and S. Oh, "Robust localization from incomplete local information", *IEEE/ACM Trans. Netw.*, 21(4), pp. 1131-1144, 2013.

[14] P. Oğuz-Ekim, J.P. Gomes, J. Xavier and P. Oliveira, "Robust localization of nodes and time-recursive tracking in sensor networks using noisy range measurements", *IEEE Trans. Signal Process.*, 59(8), pp. 3930-3942, 2011.

[15] C. Soares, J. Xavier and J. Gomes, "Simple and fast convex relaxation method for cooperative localization in sensor networks using range measurements", *IEEE Trans. Signal Process.*, 63(17), pp. 4532-4543, 2015.

[16] Y. Sun, P. Babu and D.P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning", *IEEE Trans. Signal Process.*, 65(3), pp. 794-816, 2017.

[17] N. Piovesan and T. Erseghe, "Cooperative localization in WSNs: a hybrid convex/non-convex solution", *IEEE Trans. Signal and Information Processing over Networks*, DOI 10.1109/TSIPN.2016.2639442 (IEEE early access article, 2016).

[18] C. Soares, J. Xavier and J. Gomes, "Distributed, simple and stable network localization", In *Signal and Information Processing* (GlobalSIP), 2014 IEEE Global Conference, pp. 764–768.

[19] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization", in *Proceedings of the 3rd IPSN*, Berkeley, CA, pp. 46-54, 2004.

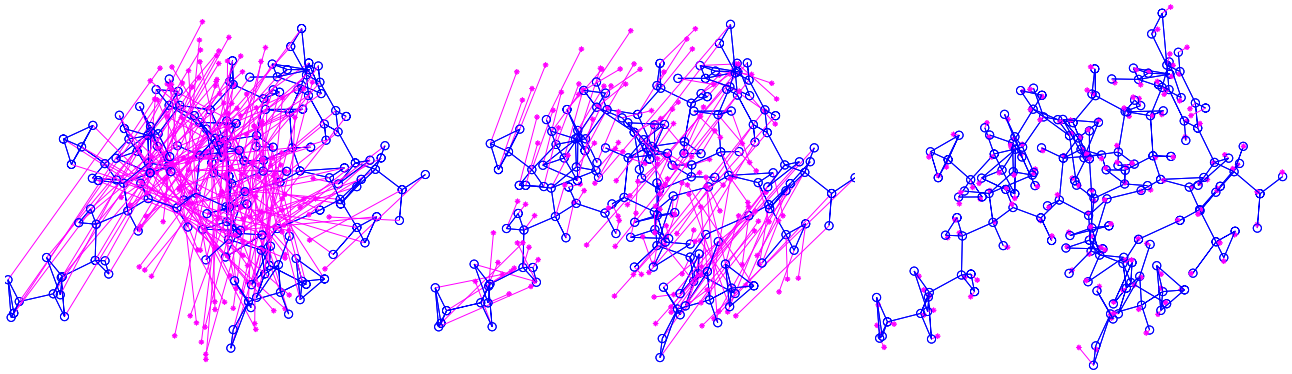
[20] K.C. Toh, M.J. Todd and R.H. Tutuncu, "SDPT3 a matlab software package for semidefinite programming", *Optim. Methods Soft.*, 11(1-4), pp. 545-581, 1999.

[21] Q. Shi, C. He, H. Chen and L. Jiang, "Distributed wireless sensor network localization via sequential greedy optimization algorithm", *IEEE Trans. Signal Process.*, 58(6), pp. 3328–3340, 2010.

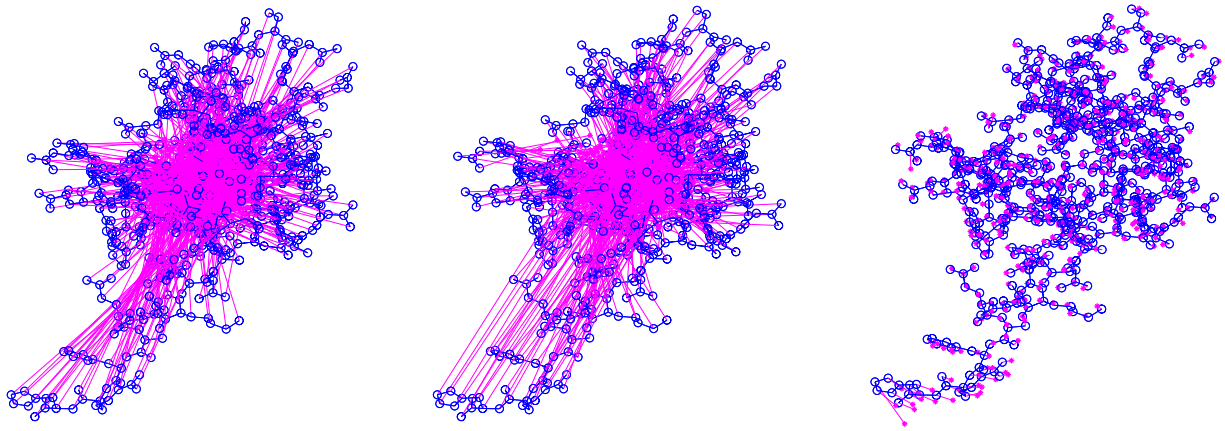
[22] N. Gaffke and R. Mathar, "A cyclic projection algorithm via duality", *Metrika*, 36, pp. 29-54, 1989.

TABLE II: Comparisons of five methods for Example 4.2.

n		$m=20$					$m=40$				
		ADMMSNL	ARAP	SFSDP	SMACOF	SQREDM	ADMMSNL	ARAP	SFSDP	SMACOF	SQREDM
400	RMSD	7.44e+0	1.90e+0	9.77e+0	3.92e+1	4.72e+0	1.82e+0	2.07e+0	4.04e+0	3.90e+1	3.94e+0
	rRMSD	6.49e+0	7.19e-1	8.61e+0	1.27e+1	5.60e-1	1.80e+0	5.40e-1	1.63e+0	4.72e+0	5.04e-1
	SD	1.70e-2	3.79e-2	6.08e-2	6.73e-3	6.29e-2	3.97e-2	2.52e-2	2.26e-2	4.76e-3	1.20e-2
	rTime	0.37	0.27	0.27	0.29	0.29	0.20	0.16	0.18	0.29	0.17
	Time	116.82	58.85	18.25	0.57	0.68	118.11	88.62	17.08	0.75	0.62
500	RMSD	5.76e+0	1.92e+0	7.96e+0	4.01e+1	4.73e+0	3.51e+0	3.40e+0	5.08e+0	3.92e+1	4.76e+0
	rRMSD	5.49e+0	4.09e-1	5.97e+0	7.04e+0	4.55e-1	3.46e+0	2.49e+0	4.07e+0	4.64e+0	5.46e-1
	SD	2.00e-2	2.91e-3	3.55e-2	3.87e-3	2.21e-2	3.60e-2	1.52e-2	1.60e-2	3.70e-3	1.15e-2
	rTime	0.46	0.40	0.40	0.41	0.31	0.44	0.44	0.41	0.46	0.28
	Time	153.17	110.40	18.82	0.78	1.19	143.99	146.38	49.21	1.27	1.17
1000	RMSD	1.00e+0	1.26e+0	7.37e+0	3.92e+1	5.11e+0	1.47e+0	1.87e+0	4.02e+0	3.85e+1	5.43e+0
	rRMSD	3.08e-1	2.45e-1	9.87e-1	6.14e+0	2.40e-1	1.41e+0	2.32e-1	5.86e-1	5.41e+0	2.32e-1
	SD	1.05e-2	3.26e-3	3.74e-2	3.49e-3	1.24e-3	1.36e-2	1.18e-2	1.09e-2	2.40e-3	8.14e-3
	rTime	1.56	1.51	1.77	1.78	1.74	0.81	1.06	1.19	1.88	1.28
	Time	338.61	1096.82	95.27	3.80	3.34	341.62	1196.51	95.25	3.87	3.01
2000	RMSD	3.62e+0	8.22e-1	6.24e+0	3.78e+1	6.49e+0	9.98e-1	1.06e+0	2.79e+0	3.76e+1	6.41e+0
	rRMSD	2.60e+0	1.91e-1	3.23e+0	8.71e+0	1.88e-1	7.02e-1	1.48e-1	1.48e-1	5.32e+0	1.48e-1
	SD	3.46e-3	1.10e-3	1.49e-2	1.41e-4	1.03e-3	5.63e-3	2.89e-3	7.33e-3	1.25e-3	2.06e-3
	rTime	12.53	12.55	13.94	13.81	13.88	10.07	7.15	7.75	11.24	10.38
	Time	910.89	14707.36	493.12	29.46	24.10	899.92	14558.45	603.62	24.06	19.36



(a) SMACOF:rRMSD=7.813 (b) SFSDP:rRMSD=4.582 (c) SQREDM:rRMSD=0.238



(d) SMACOF:rRMSD=13.67 (e) SFSDP:rRMSD=13.54 (f) SQREDM:rRMSD=0.532

Fig. 7: Molecular conformation by SMACOF, SFSDP and SQREDM. Above: 1GM2 ($n = 166$). Below: 1LFB ($n = 641$).

TABLE III: Comparisons of three methods for Example 4.3. For data 1RGS and 2CLJ, our desktop run out of memory with SFSDP and hence we omitted its results.

Data	n	RMSD			rRMSD			Time		
		SMACOF	SFSDP	SQREDM	SMACOF	SFSDP	SQREDM	SMACOF	SFSDP	SQREDM
1GM2	166	6.589	6.674	0.825	7.809	4.585	0.238	0.44	5.93	0.44
304D	237	10.287	10.334	3.454	10.681	9.933	2.581	0.52	7.16	0.48
1PBM	388	8.446	8.475	1.013	9.612	9.239	0.224	3.23	24.07	1.76
2MSJ	480	10.549	10.568	0.872	10.657	11.236	0.255	7.64	162.73	3.2
1AU6	506	9.299	9.321	0.643	9.634	10.215	0.174	9.47	35.28	2.9
1LFB	641	13.355	13.389	1.452	13.673	13.536	0.532	19.68	94.54	4.74
104D	766	12.310	12.337	2.948	12.642	13.266	1.174	34.86	51.97	8.7
1PHT	814	12.282	12.304	1.608	13.147	13.24	1.032	41.83	255.22	1.73
1POA	914	14.178	14.206	1.532	14.523	14.645	0.405	47.37	672.04	1.97
1AX8	1003	14.323	14.334	1.298	14.824	14.82	0.438	62.19	1531.1	2.35
1RGS	2015	20.223	−−	1.998	20.236	−−	0.558	487.97	−−	8.33
2CLJ	4189	22.703	−−	1.564	22.990	−−	0.598	4335.61	−−	36.82

[23] W. Glunt, T.L. Hayden, S. Hong and J. Wells, "An alternating projection algorithm for computing the nearest Euclidean distance matrix", *SIAM J. Matrix Anal. Appl.*, 11, pp. 589-600, 1990.

[24] H.-D. Qi, "A semismooth Newton method for the nearest Euclidean distance matrix problem", *SIAM J. Matrix Anal. Appl.*, 34, pp. 67-93, 2013.

[25] C. Ding and H.-D. Qi, "Convex optimization learning of faithful Euclidean distance representations in nonlinear dimensionality reduction", *Math. Program.*, 164, pp. 341-381, 2017.

[26] A. Alfakih, A. Khandani and H. Wolkowicz, "Solving Euclidean distance matrix completion problems via semidefinite programming", *Comput. Optim. Appl.*, 12, pp. 13-30, 1999.

[27] Y. Ding, N. Krislock, J. Qian and H. Wolkowicz, "Sensor network localization, Euclidean distance matrix completions, and graph realization", *Optim. Engineering*, 11, pp. 45-66, 2010.

[28] N. Krislock and H. Wolkwicz, "Explicit sensor network localization using semidefinite representations and facial reductions", *SIAM J. Optim.*, 20, 2679–2708, 2010.

[29] Z. Wang, S. Zheng, Y. Ye and S. Boyd, "Further relaxations of the semidefinite pro-gramming approach to sensor network localization", *SIAM J. Optim.*, 19(2), pp. 655-673, 2008.

[30] T.K. Pong, "Edge-based semidefinite programming relaxation of sensor network localization with lower bound constraints", *Comput Optim Appl.*, 53, pp. 23-44, 2012.

[31] S. Kim, M. Kojima, H. Waki and M. Yamashita, "Algorithm 920: SFSDP: A sparse version of full semidefinite programming relaxation for sensor network localization problems", *ACM Trans. Math. Softw.*, 38(4), pp. 27:1-27:19, 2012.

[32] I.J. Schoenberg, "Metric spaces and positive definite functions", *Trans. Am. Math. Soc.*, 44, pp. 522-536, 1938.

[33] H.-D. Qi and X.M. Yuan, "Computing the Nearest Euclidean Distance Matrix with Low Embedding Dimensions", *Math. Prog.*, 147, pp. 351-389, 2014.

[34] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd Ed., Springer Series in Operations Research and Financial Engineering, 2006.

[35] Z. Xu, X. Chang, F. Xu and H. Zhang, "$L_{1/2}$ regularization: a thresholding representation theory and a fast solver", *IEEE Trans. Neural Netw. Learn. Sys.*, 23(7), pp. 1013-1027, 2012.

[36] G. Rosman, A.M. Bronstein, M.M. Bronstein, A. Sidi and R. Kimmel, "Fast multidimensional scaling using vector extrapolation", *Tech. Report CIS-2008-01*, Depart. Comput. Sci., Technion, Israel, 2008 (SMACOF code can be obtained from http://tosca.cs.technion.ac.il).

[37] Y. Gao, *Structured Low Rank Matrix Optimization Problems: a Penalty Approach*, PhD Thesis, National University of Singapore, 2010.

[38] D.M. Burton, *The History of Mathematics* (7th Eds), MaGraw-Hill, 2011.

[39] F.C. Xing, "Investigation on solutions of cubic equations with one unknown", *J. Central Univ. Nat. (Natural Sci. Ed.)*, 12(3), pp. 207-218, 2003.

[40] C. Kanzow and H.-D. Qi, "A QP-free constrained Newton-type method for variational inequality problems", *Math. Prog.*, 85, pp. 81-106, 1999.

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

[42] S. Bai and H.-D. Qi, "Tackling the flip ambiguity in wireless sensor network localization and beyond", *Digital Signal Process.*, 55, pp. 85-97, 2016.

[43] H.M. Berman, J. Westbrook, Z. Feng, G. Gillilan, T.N. Bhat, H. Weissig, I.N. Shindyalov and P.E. Bourne, "The protein data bank", Nucleic Acids Res. 28, pp. 235242, 2000.

[44] K.F. Jiang, D.F Sun and K.C. Toh, "Solving nuclear norm regularized and semidefinite matrix least squares problems with linear equality constraints", *Discrete Geometry and Optimization*, Springer International Publishing, pp. 133-162, 2013.

[45] I. Dokmanic, R. Parhizkar, J. Ranieri and M. Vetterli, "Euclidean distance matrices: Essential theory, algorithms, and applications", *IEEE Signal Process. Mag.*, 32(6), pp. 12-30, 2015.

[46] L. Zhang, G. Wahba and M. Yuan, "Distance shrinkage and Euclidean embedding via regularized kernel estimation", *J. Royal Stat. Soc.: Series B*, 78, pp. 849-867, 2016.

[47] P.A. Forero and G.B. Giannakis, "Sparsity-exploiting robust multidimensional scaling", *IEEE Trans. Signal Process.*, 60(8), pp. 4118-4134, 2012.

[48] F.D. Mandanas and C.L. Kotropoulos, "Robust multidimensional scaling using a maximum correntropy criterion", *IEEE Trans. Signal Process.*, 65(4), pp. 919-932, 2017.

[49] I. Spence and S. Lewandowsky, "Robust multidimensional scaling", *Psychometrika*, 54(3), pp. 501-513, 1989.

[50] H.-D. Qi, N.H. Xiu, and X.M. Yuan, "A Lagrangian dual approach to the single source localization problem", *IEEE Trans. Signal Process.*, 61, pp. 3815-3826, 2013.