SIAM J. MATRIX ANAL. APPL.
Vol. 34, No. 1, pp. 67–93
© 2013 Society for Industrial and Applied Mathematics

A SEMISMOOTH NEWTON METHOD FOR THE NEAREST EUCLIDEAN DISTANCE MATRIX PROBLEM*

HOU-DUO QI†

Abstract. The nearest Euclidean distance matrix problem (NEDM) is a fundamental computational problem in applications such as multidimensional scaling and molecular conformation from nuclear magnetic resonance data in computational chemistry. Especially in the latter application, the problem is often large scale with the number of atoms ranging from a few hundreds to a few thousands. In this paper, we introduce a semismooth Newton method that solves the dual problem of NEDM. We prove that the method is quadratically convergent. We then present an application of the Newton method to NEDM with H-weights. We demonstrate the superior performance of the Newton method over existing methods including the latest quadratic semidefinite programming solver. This research also opens a new avenue toward efficient solution methods for the molecular embedding problem.

Key words. Euclidean distance matrix, semismooth Newton method, quadratic convergence

AMS subject classifications. 49M45, 90C25, 90C33

DOI. 10.1137/110849523

1. Introduction.

Finding a Euclidean distance matrix (EDM) that is nearest to a given data matrix is a fundamental computational problem in many applications including multidimensional scaling and molecular conformation from nuclear magnetic resonance data in computational chemistry. We do not intend to give a detailed account of the importance of EDM to the two applications. Instead we simply point to the excellent books [6] by Borg and Groenen and [12] by Cox and Cox for the former application and [13] by Crippen and Havel, and to the review paper [36] (and references therein) by Neumaier for the latter. We also refer to a recent paper [16] by Fang and O'Leary for algorithmic comparisons on different approaches to the EDM completion problem, which is closely related to ours. For its link to the latest development in semidefinite programming, see Dattorro [14], Toh [45], and the recent survey [30] by Krislock and Wolkowicz. The purpose of this paper is to propose an efficient Newton method for large scale problems of this type. Below we describe the problem in detail and review some existing methods that motivate our research.

Let $\mathcal{S}^n$ denote the space of $n \times n$ symmetric matrices equipped with the standard inner product $\langle A, B\rangle = \text{trace}(AB)$ for $A, B \in \mathcal{S}^n$. Let $\|\cdot\|$ denote the induced Frobenius norm. Let $\mathcal{S}^n_+$ denote the cone of positive semidefinite matrices in $\mathcal{S}^n$ (often abbreviated as $X \succeq 0$ for $X \in \mathcal{S}^n_+$). The so-called *hollow* subspace $\mathcal{S}^n_h$ is defined by

$$\mathcal{S}^n_h := \{A \in \mathcal{S}^n : \quad \text{diag}(A) = 0\},$$

where $\text{diag}(A)$ is the vector formed by the diagonal elements of A. A matrix D is an EDM if $D \in \mathcal{S}^n_h$, and there exist points $p_1, \ldots, p_n$ in $\mathbb{R}^r$ ($r \leq n-1$) such that $D_{ij} = \|p_i - p_j\|^2$ for $i, j = 1, \ldots, n$. $\mathbb{R}^r$ is often referred to as the embedding space,

*Received by the editors September 28, 2011; accepted for publication (in revised form) by Dianne P. O'Leary September 21, 2012; published electronically January 29, 2013.
http://www.siam.org/journals/simax/34-1/84952.html

†School of Mathematics, University of Southampton, Highfield, Southampton SO17 1BJ, UK (H.Qi@soton.ac.uk).

and r is the embedding dimension when it is the smallest such r. It is well known that a matrix $D \in \mathcal{S}_h^n$ is an EDM if and only if

$$J(-D)J \succeq 0 \quad \text{and} \quad J := I - ee^T/n, \tag{1}$$

where I (or I_n when the indication of dimension is needed) is the identity matrix in $\mathcal{S}^n$ and e is the vector of all ones in $\mathbb{R}^n$. The origin of this result can be traced back to Schoenberg [42] and an independent work [47] by Young and Householder. See also Gower [23] for a nice derivation of (1). The corresponding embedding dimension $r = \text{rank}(JDJ) \leq n-1$.

It is noted that the matrix J, when treated as an operator, is the orthogonal projection onto the subspace $e^\perp := \{x \in \mathbb{R}^n : \ e^T x = 0\}$. The characterization (1) simply means that D is an EDM if and only if $D \in \mathcal{S}_h^n$ and D is negative semidefinite on the subspace $e^\perp$:

$$-D \in \mathcal{K}_+^n := \left\{A \in \mathcal{S}^n : \ \ x^T A x \geq 0, \ \ x \in e^\perp\right\}. \tag{2}$$

It is easy to check whether a given data matrix D is an EDM via (1). If it is not, then it is often possible to calculate the nearest EDM to D in order to retain as much distance information as possible. This problem can be formulated as the following nearest Euclidean distance matrix problem:

$$v_p := \min \|D - X\|^2/2 \quad \text{s.t. (subject to)} \quad X \in \mathcal{S}_h^n \cap \mathcal{K}_+^n. \tag{3}$$

Given $(-D)$ used in (2), the matrix D in (3) should be $-D$. This change of sign has been widely adopted to reformulate (3) (see, e.g., [19, 17, 1]) and it reminds us that the objective is to minimize a distance. The widely used H-weighted version (see [1]) is defined as

$$\min \|H \circ (D - X)\|^2/2 \quad \text{s.t.} \quad X \in \mathcal{S}_h^n \cap \mathcal{K}_+^n, \tag{4}$$

where $H \in \mathcal{S}^n$ is nonnegative (i.e., $H_{ij} \geq 0$) and $\circ$ denotes the Hadamard product among matrices. In practice, the magnitude of H_{ij} reflects the level of accuracy of the corresponding distance D_{ij}. The H-weighted problem is much more difficult to solve than (3) (note that (4) reduces to (3) when $H = E$, the matrix of all ones in $\mathcal{S}^n$). Our main purpose in this paper is to develop a fast convergent Newton method for (3) and then apply it to (4). Below we conduct a brief literature review on both problems.

Problem (3) has been the main subject of several important papers. We first note that the feasible region is the intersection of the subspace $\mathcal{S}_h^n$ and the closed convex cone $\mathcal{K}_+^n$. Hence, alternating projection methods of Dykstra–Han type [15, 24] are a choice. In fact, one such method, called modified alternating projection (`MAP`), was studied by Glunt et al. [19]. The same method was independently studied by Gaffke and Mathar [17], but based on a different projection formula on $\mathcal{K}_+^n$ (see (15) and (17)). However, `MAP` does not apply to (4) unless $H = E$. Problem (3) (and in general (4)) can also be solved by *semidefinite programming* (SDP) initiated by Alfakih, Khandani, and Wolkowicz [1] (see also [2]). We note that the dimension of $\mathcal{S}_h^n$ is $n(n-1)/2$, as is the dimension of the cone $\mathcal{E}^n$ of the Euclidean distance matrices, where $\mathcal{E}^n := \mathcal{S}_h^n \cap (-\mathcal{K}_+^n)$ (see [26]). Alfakih et al. introduced their interesting linear mapping, $\mathcal{K}_V : \mathcal{S}_+^{n-1} \mapsto \mathcal{E}^n$, defined by

$$\mathcal{K}_V(X) = \text{diag}(VXV^T)e^T + e\text{diag}(VXV^T)^T - 2VXV^T,$$

where $V \in \mathbb{R}^{n\times(n-1)}$ satisfies $V^TV = I_{n-1}$ and $V^Te = 0$. Then (4) is equivalent to the problem

$$\min \|H \circ (\mathcal{K}_V(X) - D)\|^2/2 \qquad \text{s.t.} \quad X \in \mathcal{S}_+^{n-1}. \tag{5}$$

It is also interesting to note that the mapping $\mathcal{K}_V$ also allows for Slater's condition to hold (see [1, Cor. 2]). Alfakih, Khandani, and Wolkowicz studied an interior point method based on the Gauss–Newton direction. This method can deal only with problems with size up to 100. Problem (5) (possibly with more linear equalities) was one of the major convex quadratic semidefinite programs studied by Toh [45], where problem size n can be up to a few thousand. Other linear mappings instead of $\mathcal{K}_V$ can be used; see [30]. Generally speaking, specific transformations must take place before SDP can be applied to (3) and (4). Such transformations aim to shift the difficulty of handling the cone $\mathcal{K}_+^n$ to new objective functions defined on $\mathcal{S}_+^n$. The cost is that the new objective function is more complicated than the original distance function. Those transformations tend to destroy the nice geometric properties of $\mathcal{K}_+^n$, which we will take advantage of to develop our Newton method.

Problem (3) also plays a very important role in solving the *embedding problem*. In multidimensional scaling, the given matrix D is often called dissimilarity and the embedding dimension r should be small, whereas in molecular conformation D is often called a predistance matrix (i.e., $D \in \mathcal{S}_h^n$ with $D_{ij} \geq 0$ for all i, j) and the embedding dimension $r = 3$. Therefore, the well-known embedding problem [20] is to find the nearest EDM with a low embedding dimension r:

$$\min \|D - X\|^2/2 \quad \text{s.t.} \quad X \in \mathcal{S}_h^n \cap \mathcal{K}_+^n \quad \text{and} \quad \operatorname{rank}(JXJ) \leq r. \tag{6}$$

Compared to (3), problem (6) is nonconvex and hence is extremely difficult to solve. Phase I of the two-phase methods in [20] for (6) is to find the optimal solution of (3) and modify it as a starting point for the Phase II method. Further developments can be found in [21, 22]. Fast algorithms for (3) are crucial in those applications.

Problem (3) also bears a remarkable resemblance to the *nearest correlation matrix* problem:

$$\min \ \|C - X\|^2/2 \quad \text{s.t.} \quad \operatorname{diag}(X) = e, \quad X \in \mathcal{S}_+^n, \tag{7}$$

where $C \in \mathcal{S}^n$ is given. The constraints define the set of all $n \times n$ correlation matrices. Higham [28] studied this problem and made it widely accessible to the community of numerical analysis and optimization. The subsequent research, all trying to design efficient algorithms for (7), includes (to name just a few) [31, 8, 38, 45, 7]. An important approach which emerged from those studies is the Lagrangian dual approach, which was first applied to (7) by Malick [31] and Boyd and Xiao [8]. The dual approach was then studied by Qi and Sun [38] to design what is now known as one of the most efficient methods for (7): the *semismooth Newton method*. The link between the dual approach and that used in classical computational mathematics (see e.g., [33, 34]) was well discussed in [38]. This class of research also motivated our research in this paper.

When applied to problem (3), the Lagrangian dual problem (in the form of minimization) becomes (see [34, Thm. 2.2] and also [41, 31, 8])

$$v_d := \min_{y\in\mathbb{R}^n} \ \theta(y) := \|\Pi_{\mathcal{K}_+^n}(D + \operatorname{Diag}(y))\|^2/2 - \|D\|^2/2, \tag{8}$$

where $\Pi_{\mathcal{K}_+^n}(\cdot)$ denotes the orthogonal projection onto the closed convex cone $\mathcal{K}_+^n$, and $\operatorname{Diag}(y)$ is the diagonal matrix with y being its diagonal. Function $\theta(\cdot)$ is just

once continuously differentiable, but convex. Furthermore, (8) must have an optimal solution (see Proposition 2.1), which can be found through the first-order optimality condition

$$F(y) := \nabla\theta(y) = \mathrm{diag}\Big(\Pi_{\mathcal{K}^n_+}(D + \mathrm{Diag}(y))\Big) = 0. \tag{9}$$

If y is a solution of (9), then

$$X := \Pi_{\mathcal{K}^n_+}(D + \mathrm{Diag}(y)) \tag{10}$$

is the optimal solution of (3). This follows from the *zero* duality gap result (i.e., $v_p = -v_d$), which can be easily proved via writing down the Lagrangian function of (3) and using Proposition 2.1. Hence, it is enough to solve the dual problem and it is relatively easy to solve as it is defined in $\mathbb{R}^n$ rather than in the significantly larger space $\mathcal{S}^n$.

It follows from the projection formula (17) of Gaffke and Mathar that $F(y)$ is *strongly semismooth*[1] because it is a composition of linear mappings and $\Pi_{\mathcal{S}^n_+}(\cdot)$ (the orthogonal projection onto $\mathcal{S}^n_+$), and $\Pi_{\mathcal{S}^n_+}(\cdot)$ is known to be strongly semismooth [44, 9]. Now it becomes natural to develop the semismooth Newton method. Given $y^0 \in \mathbb{R}^n$ and letting $k := 0$, compute $V_k \in \partial F(y^k)$ and

$$y^{k+1} = y^k - V_k^{-1}F(y^k), \qquad k = 0, 1, 2, \ldots. \tag{11}$$

Since F is the gradient of θ, ∂F is often called the generalized Hessian of θ, denoted by $\partial^2\theta(y)$. We refer to [38, sect. 3] for a detailed development of (11) for (7). The above arguments leading to the Newton method (11) for (3) fail to hold for the H-weighted problem (4) because the projection onto $\mathcal{K}^n_+$ under the H-weights does not have an analytical formula. It is already very difficult to calculate the projection under the H-weights, let alone compute its generalized Jacobian.

Therefore, our main tasks in this paper are (i) to address the quadratic convergence of (11); (ii) to demonstrate its superior numerical performance, especially on large scale problems; and (iii) to apply it to the H-weighted problem (4).

The paper is organized as follows. Sections 2 and 2.1 are devoted to the Newton method (11). In section 2.2, we include some notation and technical results. One of the results states that problem (3) is constraint nondegenerate (Theorem 2.3). A characterization of the constraint nondegeneracy (Proposition 2.4) generalizes the corresponding result in SDP of Alizadeh, Haeberly, and Overton [3]. In section 3, we conduct quadratic convergence analysis of the Newton method (11). The main result is Theorem 3.3, which says that every matrix in the generalized Jacobian of F at the optimal solution is positive definite. This result then leads to the quadratic convergence result Theorem 3.5. Section 4 includes an application of the Newton method to the H-weighted problem (4). We report our numerical results in section 5, and we conclude the paper in section 6 by discussing the use of the Newton method in future research.

2. Preliminaries. In this section, we first list most of the notation used in this paper and review two formulae of $\Pi_{\mathcal{K}^n_+}$. We finish this section by establishing two results on the existence of an optimal dual solution for (8) and constraint nondegeneracy of (3).

[1] A (locally) Lipschitz function $\Phi : \mathbb{R}^m \mapsto \mathbb{R}^\ell$ is said to be strongly semismooth at $x \in \mathbb{R}^m$ if (i) Φ is directionally differentiable at x; and (ii) for any $V \in \partial\Phi(x + h)$, $\Phi(x + h) - \Phi(x) - Vh = o(\|h\|^2), h \in \mathbb{R}^m$, where $\partial\Phi(x)$ denotes the generalized Jacobian of Φ at x in the sense of Clarke [11, sect. 2.6].

2.1. Notation and two formulae of $\Pi_{\mathcal{K}^n_+}$. Apart from $\mathcal{S}^n$, $\mathcal{S}^n_+$, $\mathcal{S}^n_h$, $\mathcal{K}^n_+$, J, diag, and Diag mentioned in the introduction, we also need the following (":=" means "define"): e_i is the ith unit basis vector in $\mathbb{R}^n$ and e is the vector of all ones. E is the matrix of all ones. Define the Householder matrix Q by

$$Q := I - \frac{2}{v^T v} v v^T, \qquad v = [1, \ldots, 1, \sqrt{n}+1]^T \in \mathbb{R}^n, \tag{12}$$

where v^T is the transpose of v. We note that Q is symmetric and orthogonal: $Q^2 = I$. We often partition a matrix $X \in \mathcal{S}^n$ into blocks

$$X = \begin{bmatrix} X_1 & x \\ x^T & x_0 \end{bmatrix}, \qquad \text{with} \quad X_1 \in \mathcal{S}^{n-1},\ x \in \mathbb{R}^{n-1},\ x_0 \in \mathbb{R}.$$

$\Pi_{\mathcal{S}^n_+}(X)$ is the orthogonal projection of X onto $\mathcal{S}^n_+$. For a close convex cone $\mathcal{C}$ in $\mathcal{S}^n$, its polar cone $\mathcal{C}^*$ is defined by

$$\mathcal{C}^* := \{X \in \mathcal{S}^n :\ \langle X,\ A\rangle \le 0 \ \ \forall\ A \in \mathcal{C}\}.$$

The normal cone $\mathcal{N}_{\mathcal{K}^n_+}(\overline{A})$ of $\mathcal{K}^n_+$ at $\overline{A} \in \mathcal{K}^n_+$ is defined by

$$\mathcal{N}_{\mathcal{K}^n_+}(\overline{A}) := \left\{X \in \mathcal{S}^n :\ \langle X,\ A - \overline{A}\rangle \le 0 \ \ \forall\ A \in \mathcal{K}^n_+\right\}.$$

Since $\mathcal{K}^n_+$ is convex, the tangent cone $\mathcal{T}_{\mathcal{K}^n_+}(\overline{A})$ of $\mathcal{K}^n_+$ at $\overline{A} \in \mathcal{K}^n_+$ can be conveniently defined as the polar cone of $\mathcal{N}_{\mathcal{K}^n_+}(\overline{A})$:

$$\mathcal{T}_{\mathcal{K}^n_+}(\overline{A}) := \left(\mathcal{N}_{\mathcal{K}^n_+}(\overline{A})\right)^*. \tag{13}$$

We let $\mathrm{lin}(\mathcal{T}_{\mathcal{K}^n_+}(\overline{A}))$ denote the largest linear space contained in $\mathcal{T}_{\mathcal{K}^n_+}(\overline{A})$. $A \circ B := [A_{ij}B_{ij}]$ is the Hadamard product between two matrices A and B of the same size. For subsets α, β of $\{1, \ldots, n\}$, denote $B_{\alpha\beta}$ as the submatrix of B indexed by α and β (α for rows and β for columns). B_α denotes the submatrix consisting of columns of B indexed by α, and $|\alpha|$ is the cardinality of α.

There are two known formulae for computing $\Pi_{\mathcal{K}^n_+}$. One is due to Hayden and Wells [25, Thm. 2.1]:

$$A \in \mathcal{K}^n_+ \qquad \Longleftrightarrow \qquad QAQ =: \begin{bmatrix} \widehat{A} & \hat{a} \\ \hat{a}^T & \hat{a}_0 \end{bmatrix} \qquad \text{and} \qquad \widehat{A} \in \mathcal{S}^{n-1}_+ \tag{14}$$

and

$$\Pi_{\mathcal{K}^n_+}(A) = Q \begin{bmatrix} \Pi_{\mathcal{S}^{n-1}_+}(\widehat{A}) & \hat{a} \\ \hat{a}^T & \hat{a}_0 \end{bmatrix} Q, \qquad \forall\ A \in \mathcal{S}^n, \tag{15}$$

where (and throughout this paper) Q is the Householder matrix defined in (12). Because of (15), the cone $\mathcal{K}^n_+$ can be described as follows:

$$\mathcal{K}^n_+ = \left\{ Q \begin{bmatrix} Z & z \\ z^T & z_0 \end{bmatrix} Q : \quad \begin{array}{l} Z \in \mathcal{S}^{n-1}_+ \\ z \in \mathbb{R}^{n-1},\ z_0 \in \mathbb{R} \end{array} \right\}.$$

Its polar cone $(\mathcal{K}^n_+)^*$ is then given by

$$\left(\mathcal{K}^n_+\right)^* = \left\{ Q \begin{bmatrix} Z & 0 \\ 0 & 0 \end{bmatrix} Q : \quad Z \in -\mathcal{S}^{n-1}_+ \right\}. \tag{16}$$

The other projection formula is due to Gaffke and Mathar [17, eq. (29)]:

$$\Pi_{\mathcal{K}^n_+}(A) = A + \Pi_{\mathcal{S}^n_+}(-JAJ) \qquad \forall\ A \in \mathcal{S}^n. \tag{17}$$

We note that the original projection formula of Gaffke and Mathar is onto $(-\mathcal{K}^n_+)$. Each formula has its own advantage. Formula (15) states that the projection is in fact carried out onto $\mathcal{S}^{n-1}_+$, while (17) brings the formula to the defining space $\mathcal{S}^n$. We will use the Gaffke–Mathar formula in our numerical implementation and the Hayden–Wells formula for our analysis because it brings out the rich structures that exist in $\mathcal{T}_{\mathcal{K}^n_+}(\overline{A})$.

2.2. Existence of optimal dual solutions and constraint nondegeneracy. The following result on the coerciveness of the dual problem (8) ensures that it must have an optimal solution.

PROPOSITION 2.1. *The function $\theta(\cdot)$ in* (8) *is coercive, i.e., $\theta(y) \to +\infty$ as $\|y\| \to +\infty$. Consequently, the dual problem* (8) *must have an optimal solution.*

Proof. Suppose to the contrary that $\theta(\cdot)$ is not coercive. Then there must exist a sequence $\{y^k\}$ such that $\|y^k\| \to +\infty$ and $\theta(y^k) \le c$ for some constant $c > 0$. We consider the sequence $\{y^k/\|y^k\|\}$, which, without loss of generality, is assumed to converge to y^*. Because $\mathcal{K}^n_+$ is a cone, we have

$$\frac{c}{\|y^k\|^2} \ge \frac{\theta(y^k)}{\|y^k\|^2} = \frac{1}{2}\left\|\Pi_{\mathcal{K}^n_+}\left(\frac{D}{\|y^k\|} + \frac{\mathrm{Diag}(y^k)}{\|y^k\|}\right)\right\|^2 - \frac{1}{2}\frac{\|D\|^2}{\|y^k\|^2}.$$

Taking the limit on both sides of the above inequality, we have (due to the continuity of the projection operator $\Pi_{\mathcal{K}^n_+}(\cdot)$)

$$\|\Pi_{\mathcal{K}^n_+}(\mathrm{Diag}(y^*))\| \le 0,$$

which means $\Pi_{\mathcal{K}^n_+}(\mathrm{Diag}(y^*)) = 0$. Consequently, $\mathrm{Diag}(y^*) \in (\mathcal{K}^n_+)^*$.

It follows from (16) that there exists $Z \in -\mathcal{S}^{n-1}_+$ such that

$$\mathrm{Diag}(y^*) = Q\begin{bmatrix} Z & 0 \\ 0 & 0 \end{bmatrix}Q \qquad \text{or, equivalently,} \qquad Q\mathrm{Diag}(y^*)Q = \begin{bmatrix} Z & 0 \\ 0 & 0 \end{bmatrix}.$$

Obviously, the last column of $Q\mathrm{Diag}(y^*)Q$ is zero:

$$0 = Q\mathrm{Diag}(y^*)Qe_n = -\frac{1}{\sqrt{n}}Q\mathrm{Diag}(y^*)e = -\frac{1}{\sqrt{n}}Qy^*,$$

where we used $Qe_n = -e/\sqrt{n}$. The nonsingularity of Q implies $y^* = 0$, contradicting $\|y^*\| = 1$. This proves that $\theta(\cdot)$ is coercive. □

Constraint nondegeneracy plays a very important role in optimization; see [3, Def. 5], [10, Def. 9], and [37, sect. 2] for its use in SDP. Generally speaking, it ensures a certain regularity of optimal solutions. For our problem (3), the constraint nondegeneracy is defined as follows.

DEFINITION 2.2. *We say that the constraint nondegeneracy holds at $\overline{A} \in \mathcal{S}^n_h \cap \mathcal{K}^n_+$ if*

$$\mathrm{diag}\Big(\mathrm{lin}\,(\mathcal{T}_{\mathcal{K}^n_+}(\overline{A}))\Big) = \mathbb{R}^n. \tag{18}$$

Let $\overline{A} \in \mathcal{K}_+^n$ and

$$\overline{A} = Q \begin{bmatrix} Z & z \\ z^T & z_0 \end{bmatrix} Q, \qquad Z \in \mathcal{S}^{n-1}. \tag{19}$$

Then $Z \succeq 0$ by (14). We assume that $\operatorname{rank}(Z) = r$ and let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0$ be the r positive eigenvalues of Z in nonincreasing order. Let $\Lambda = \operatorname{Diag}(\lambda_1, \ldots, \lambda_r)$. We assume that Z takes the following spectral decomposition:

$$Z = U \begin{bmatrix} \Lambda & \\ & 0 \end{bmatrix} U^T, \tag{20}$$

where $U \in \mathbb{R}^{(n-1)\times(n-1)}$ and $U^T U = I_{n-1}$. The normal cone $\mathcal{N}_{\mathcal{K}_+^n}(\overline{A})$ is given by [19, Thm. 3.1]:

$$\mathcal{N}_{\mathcal{K}_+^n}(\overline{A}) = \left\{ Q \begin{bmatrix} U \begin{bmatrix} 0 & 0 \\ 0 & M \end{bmatrix} U^T & 0 \\ 0 & 0 \end{bmatrix} Q : \quad -M \in \mathcal{S}_+^{n-r-1} \right\}.$$

Let

$$\overline{U} := \begin{bmatrix} U & 0 \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{n\times n}. \tag{21}$$

Then $\overline{U}^T \overline{U} = I$, and the normal cone can be equivalently written as

$$\mathcal{N}_{\mathcal{K}_+^n}(\overline{A}) = \left\{ Q\overline{U} \begin{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & M \end{bmatrix} & 0 \\ 0 & 0 \end{bmatrix} \overline{U}^T Q : \quad -M \in \mathcal{S}_+^{n-r-1} \right\}.$$

By definition (13) of the tangent cone in terms of $\mathcal{N}_{\mathcal{K}_+^n}(\overline{A})$, we have

$$\mathcal{T}_{\mathcal{K}_+^n}(\overline{A}) = \left\{ Q\overline{U} \begin{bmatrix} \begin{bmatrix} \Sigma_1 & \Sigma_{12} \\ \Sigma_{12}^T & \Sigma_2 \end{bmatrix} & a \\ a^T & a_0 \end{bmatrix} \overline{U}^T Q : \begin{array}{l} \Sigma_1 \in \mathcal{S}^r, \Sigma_2 \in \mathcal{S}_+^{n-r-1} \\ \Sigma_{12} \in \mathbb{R}^{r\times(n-r-1)} \\ a \in \mathbb{R}^{n-1}, \quad a_0 \in \mathbb{R} \end{array} \right\} \tag{22}$$

$$= \left\{ Q \begin{bmatrix} U \begin{bmatrix} \Sigma_1 & \Sigma_{12} \\ \Sigma_{12}^T & \Sigma_2 \end{bmatrix} U^T & a \\ a^T & a_0 \end{bmatrix} Q : \begin{array}{l} \Sigma_1 \in \mathcal{S}^r, \Sigma_2 \in \mathcal{S}_+^{n-r-1} \\ \Sigma_{12} \in \mathbb{R}^{r\times(n-r-1)} \\ a \in \mathbb{R}^{n-1}, \quad a_0 \in \mathbb{R} \end{array} \right\}. \tag{23}$$

The last equality used the facts that U is nonsingular and $[a^T, a_0]$ is not restricted. Now we are ready to prove the following result.

THEOREM 2.3. *Constraint nondegeneracy holds at each feasible point $\overline{A}$ of problem* (3).

Proof. We need only prove condition (18). It follows from (22) that

$$\operatorname{lin}(\mathcal{T}_{\mathcal{K}_+^n}(\overline{A})) = \left\{ Q\overline{U} \begin{bmatrix} \begin{bmatrix} \Sigma_1 & \Sigma_{12} \\ \Sigma_{12}^T & 0 \end{bmatrix} & a \\ a^T & a_0 \end{bmatrix} \overline{U}^T Q : \begin{array}{l} \Sigma_1 \in \mathcal{S}^r \\ \Sigma_{12} \in \mathbb{R}^{r\times(n-r-1)} \\ a \in \mathbb{R}^{n-1}, a_0 \in \mathbb{R} \end{array} \right\}. \tag{24}$$

It is obvious from (23) that

$$A = Q \begin{bmatrix} 0_{(n-1)\times(n-1)} & a \\ a^T & a_0 \end{bmatrix} Q \in \operatorname{lin}(\mathcal{T}_{\mathcal{K}_+^n}(\overline{A})) \qquad \forall\ [a^T, a_0] \in \mathbb{R}^n.$$

Let $b \in \mathbb{R}^n$ be arbitrary. We will find $[a^T, a_0] \in \mathbb{R}^n$ such that

$$\mathrm{diag}(A) = b. \tag{25}$$

We calculate the diagonal of A. For $i = 1, \ldots, n$, we have

$$\begin{aligned} A_{ii} &= e_i^T Q \begin{bmatrix} 0_{(n-1)\times(n-1)} & a \\ a^T & a_0 \end{bmatrix} Q e_i = \mathrm{trace}\left(Q e_i (e_i^T Q) \begin{bmatrix} 0_{(n-1)\times(n-1)} & a \\ a^T & a_0 \end{bmatrix} \right) \\ &= e_n^T (Q e_i e_i^T Q) \begin{bmatrix} 2a \\ a_0 \end{bmatrix} \\ &= -\frac{1}{\sqrt{n}} e^T (e_i e_i^T Q) \begin{bmatrix} 2a \\ a_0 \end{bmatrix} \qquad \left(\text{using } Q e_n = -\frac{1}{\sqrt{n}} e \right) \\ &= -\frac{1}{\sqrt{n}} (e_i^T Q) \begin{bmatrix} 2a \\ a_0 \end{bmatrix}. \end{aligned}$$

We therefore have

$$\mathrm{diag}(A) = -\frac{1}{\sqrt{n}} Q \begin{bmatrix} 2a \\ a_0 \end{bmatrix}.$$

Substituting this into (25) to solve for a and a_0, we obtain

$$\begin{bmatrix} 2a \\ a_0 \end{bmatrix} = -\sqrt{n} Q b.$$

With such choice of a and a_0 in A, we have $b = \mathrm{diag}(A) \in \mathrm{diag}(\mathrm{lin}(\mathcal{T}_{\mathcal{K}_+^n}(\overline{A})))$. This proves (18) and hence the constraint nondegeneracy at $\overline{A}$. □

Theorem 2.3 is not practical enough for our use. We now develop a result for later use. Let $E_i := e_i e_i^T$ for $i = 1, \ldots, n$, and let

$$B_i := Q E_i Q =: \begin{bmatrix} B_1^i & b^i \\ (b^i)^T & b_0^i \end{bmatrix} \qquad \text{with } \; B_1^i \in \mathcal{S}^{n-1}.$$

Let U be defined as in (20). We define the corresponding index sets

$$\alpha(Z) := \{ i : \; \lambda_i > 0 \} \qquad \text{and} \qquad \overline{\alpha}(Z) := \{1, 2, \ldots, n-1\} \setminus \alpha(Z). \tag{26}$$

Whenever no confusion arises, we abbreviate $\alpha(Z)$, $\overline{\alpha}(Z)$ as α and $\overline{\alpha}$, respectively. We write

$$U = [U_\alpha, \; U_{\overline{\alpha}}].$$

We further define

$$C_i := \begin{bmatrix} \begin{bmatrix} U_\alpha^T B_1^i U_\alpha & U_\alpha^T B_1^i U_{\overline{\alpha}} \\ U_{\overline{\alpha}}^T B_1^i U_\alpha & 0 \end{bmatrix} & U^T b^i \\ (b^i)^T U & b_0^i \end{bmatrix} = \overline{U}^T B_i \overline{U} - \begin{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & U_{\overline{\alpha}}^T B_1^i U_{\overline{\alpha}} \end{bmatrix} & 0 \\ 0 & 0 \end{bmatrix}, \tag{27}$$

where $\overline{U}$ is defined by (21).

Proposition 2.4. *The matrices $\{C_i\}_{i=1}^n$ are linearly independent at each feasible point $\overline{A}$ of (3).*

Proof. We note that the constraint nondegeneracy condition (18) is equivalent to

$$\mathcal{S}_h^n + \mathrm{lin}(\mathcal{T}_{\mathcal{K}_+^n}(\overline{A})) = \mathcal{S}^n,$$

which in turn is equivalent to

$$\Big(S_h^n\Big)^{\perp} \cap \Big(\mathrm{lin}(\mathcal{T}_{\mathcal{K}_+^n}(\overline{A}))\Big)^{\perp} = \{0\}, \tag{28}$$

where $(\mathcal{S}_h^n)^{\perp}$ denotes the subspace orthogonal to $\mathcal{S}_h^n$.

Because of Theorem 2.3, (28) holds. We prove the claim by contradiction. Assume that $\{C_i\}_{i=1}^n$ are linearly dependent. Then there exists $0 \neq h \in \mathbb{R}^n$ such that $\sum_{i=1}^n h_i C_i = 0$. It follows from (27) that

$$\begin{aligned}\sum_{i=1}^n h_i C_i &= \overline{U}^T\Big(\sum(h_i B_i)\Big)\overline{U} - \begin{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & U_{\overline{\alpha}}^T \sum(h_i B_1^i) U_{\overline{\alpha}} \end{bmatrix} & 0 \\ 0 & 0 \end{bmatrix} \\ &= \overline{U}^T\Big(Q(\mathrm{Diag}(h))Q\Big)\overline{U} - \begin{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & U_{\overline{\alpha}}^T \sum(h_i B_1^i) U_{\overline{\alpha}} \end{bmatrix} & 0 \\ 0 & 0 \end{bmatrix}, \end{aligned} \tag{29}$$

where we used the trivial identity

$$\sum_{i=1}^n h_i B_i = Q(\mathrm{Diag}(h))Q. \tag{30}$$

Hence, $\sum_{i=1}^n h_i C_i = 0$ implies

$$\overline{U}^T\Big(Q(\mathrm{Diag}(h))Q\Big)\overline{U} = \begin{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & U_{\overline{\alpha}}^T \sum(h_i B_1^i) U_{\overline{\alpha}} \end{bmatrix} & 0 \\ 0 & 0 \end{bmatrix}. \tag{31}$$

It is obvious that $\mathrm{Diag}(h) \in (\mathcal{S}_h^n)^{\perp}$. For any $A \in \mathrm{lin}(\mathcal{T}_{\mathcal{K}_+^n}(\overline{A}))$, we have

$$\langle \mathrm{Diag}(h),\ A\rangle = \langle \overline{U}^T Q(\mathrm{Diag}(h))Q\overline{U},\ \overline{U}^T Q A Q \overline{U}\rangle = \Big\langle U_{\overline{\alpha}}^T \sum(h_i B_1^i) U_{\overline{\alpha}},\ 0\Big\rangle = 0.$$

The first equality used the fact that $Q\overline{U}$ is orthogonal because both Q and $\overline{U}$ are orthogonal. The second equality used (31) and the structure of $\mathrm{lin}(\mathcal{T}_{\mathcal{K}_+^n}(\overline{A}))$ in (24). Hence

$$0 \neq \mathrm{Diag}(h) \in \Big(\mathcal{S}_h^n\Big)^{\perp} \cap \Big(\mathrm{lin}(\mathcal{T}_{\mathcal{K}_+^n}(\overline{A}))\Big)^{\perp},$$

which contradicts (28). This proves the linear independence of $\{C_i\}_{i=1}^n$. ∎

As a matter of fact, it is not hard to derive the constraint nondegeneracy at $\overline{A}$ from the linear independence of $\{C_i\}_{i=1}^n$. In other words, linear independence of $\{C_i\}_{i=1}^n$ is equivalent to constraint nondegeneracy at $\overline{A}$. It is interesting to note that this equivalent characterization is a natural extension of a result of Alizadeh, Haeberly, and Overton [3, Thm. 6] on primal nondegeneracy in SDP from $\mathcal{S}_+^n$ to $\mathcal{K}_+^n$.

3. Quadratic convergence. This section is mainly concerned with the quadratic convergence of the Newton method (11). Globalizing the Newton method is straightforward, as the dual problem (8) is convex (see section 5). Our key result is that every matrix in $\partial^2\theta(y)$ is positive definite when y is an optimal solution of (8). This result will lead to the desired quadratic convergence. To facilitate our analysis, we need to study the structure of $\partial^2\theta(y) = \partial F(y)$.

It follows from the definition of $F(y)$ in (9) and the Gaffke–Mathar formula (17) that (also using $\mathrm{diag}(D) = 0$)

$$F(y) = y + \mathrm{diag}(\Pi_{\mathcal{S}^n_+}(-J(D + \mathrm{Diag}(y))J).$$

The Jacobian chain rule of Clarke [11, Thm. 2.6.6] implies

$$\partial^2\theta(y)h \subseteq h - \mathrm{diag}\Big(\partial\Pi_{\mathcal{S}^n_+}(Y)(J(\mathrm{Diag}(h))J)\Big), \tag{32}$$

where $Y := -J(D + \mathrm{Diag}(y))J$. We will reveal the rich structures in $\partial\Pi_{\mathcal{S}^n_+}(Y)(J\mathrm{Diag}(h)J)$ step by step to prove our ultimate result on quadratic convergence of (11).

3.1. Generalized Jacobian of $\Pi_{\mathcal{S}^n_+}(\cdot)$. Let

$$Y := -J(D + \mathrm{Diag}(y))J \qquad \text{and} \qquad Y = P\Lambda P^T, \tag{33}$$

where $P^TP = I$ and $\Lambda := \mathrm{Diag}(\lambda_1, \ldots, \lambda_n)$, with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ being eigenvalues of Y in nonincreasing order. For those eigenvalues, define the corresponding symmetric matrix $\Omega \in \mathcal{S}^n$ with entries

$$\Omega_{ij} := \frac{\max\{\lambda_i, 0\} + \max\{\lambda_j, 0\}}{|\lambda_i| + |\lambda_j|}, \qquad i, j = 1, \ldots, n, \tag{34}$$

where $0/0$ is defined to be 1.

We further define three index sets

$$\alpha(Y) := \{i : \ \lambda_i > 0\}, \quad \beta(Y) := \{i : \ \lambda_i = 0\}, \quad \gamma(Y) := \{i : \ \lambda_i < 0\}. \tag{35}$$

We will drop the dependence of those indices on Y whenever no confusion arises. We have the following formula describing $\partial\Pi_{\mathcal{S}^n_+}(Y)$.

PROPOSITION 3.1 (see [43, Prop. 2.2]). *Suppose that $Y \in \mathcal{S}^n$ has the spectral decomposition as in* (33). *Then $V \in \partial\Pi_{\mathcal{S}^n_+}(Y)$ if and only if there exists $\widetilde{V} \in \partial\Pi_{\mathcal{S}^{|\beta|}_+}(0)$ such that*

$$V(H) = P\begin{bmatrix} \widetilde{H}_{\alpha\alpha} & \widetilde{H}_{\alpha\beta} & \Omega_{\alpha\gamma} \circ \widetilde{H}_{\alpha\gamma} \\ \widetilde{H}^T_{\alpha\beta} & \widetilde{V}(\widetilde{H}_{\beta\beta}) & 0 \\ \Omega^T_{\alpha\gamma} \circ \widetilde{H}^T_{\alpha\gamma} & 0 & 0 \end{bmatrix} P^T \qquad \forall\ H \in \mathcal{S}^n, \tag{36}$$

where $\widetilde{H} := P^THP$.

Therefore, to specify an element $V \in \partial\Pi_{\mathcal{S}^n_+}(Y)$ one needs to specify the corresponding $\widetilde{V}$ from $\partial\Pi_{\mathcal{S}^{|\beta|}_+}(0)$. It is usually complicated to specify all elements in $\partial\Pi_{\mathcal{S}^{|\beta|}_+}(0)$ (see [32], which is solely devoted to a detailed characterization). But for us we need only the following property on $\widetilde{V}$:

$$\langle Z_1,\ \widetilde{V}(Z_2)\rangle \leq \|Z_1\|\|Z_2\| \qquad \forall\ Z_1, Z_2 \in \mathcal{S}^{|\beta|}. \tag{37}$$

This can be easily proved by using [10, eq. (17)].

The general description in (36) is not adequate for our further analysis. We need to break it down to reveal the structure of our problem. Next we establish a useful relationship between P and Q.

3.2. Relationship between P and Q. The following identity has been used by Glunt et al. [19, p. 591]:

$$Q \begin{bmatrix} I_{n-1} & 0 \\ 0 & 0 \end{bmatrix} Q = J. \tag{38}$$

It follows from (33) that

$$Y = Q \begin{bmatrix} \widehat{Y}_1 & 0 \\ 0 & 0 \end{bmatrix} Q, \tag{39}$$

where we denote

$$-Q(D + \mathrm{Diag}(y))Q =: \begin{bmatrix} \widehat{Y}_1 & \widehat{y} \\ \widehat{y}^T & \widehat{y}_0 \end{bmatrix} \quad \text{with } \widehat{Y}_1 \in \mathcal{S}^{n-1}. \tag{40}$$

Let $\widehat{Y}_1 \in \mathcal{S}^{n-1}$ take the spectral decomposition

$$\widehat{Y}_1 = W \widehat{\Lambda} W^T, \tag{41}$$

where $\widehat{\Lambda} := \mathrm{Diag}(\widehat{\lambda}_1, \ldots, \widehat{\lambda}_{n-1})$, with $\widehat{\lambda}_1 \geq \cdots \geq \widehat{\lambda}_{n-1}$ being eigenvalues of $\widehat{Y}_1$ and $W \in \mathbb{R}^{(n-1)\times(n-1)}$, $W^T W = I_{n-1}$. Define

$$\widehat{\alpha} := \{i : \ \widehat{\lambda}_i > 0\}, \quad \widehat{\beta} := \{i : \ \widehat{\lambda}_i = 0\}, \quad \widehat{\gamma} := \{i : \ \widehat{\lambda}_i < 0\}, \tag{42}$$

and

$$W = [W_{\widehat{\alpha}}, \ W_{\widehat{\beta}}, \ W_{\widehat{\gamma}}]. \tag{43}$$

Then we have

$$\overline{Y} := \begin{bmatrix} \widehat{Y}_1 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \widehat{\Lambda} & \\ & 0 \end{bmatrix} \begin{bmatrix} W^T & 0 \\ 0 & 1 \end{bmatrix}.$$

This means that in addition to $\{\widehat{\lambda}_1, \ldots, \widehat{\lambda}_{n-1}\}$, 0 is the last eigenvalue of $\overline{Y}$ and e_n is the corresponding eigenvector. It follows from (39) that Y and $\overline{Y}$ share the same set of eigenvalues because Q is orthogonal. The relationship between index sets α, β, and γ in (35) and $\widehat{\alpha}$, $\widehat{\beta}$, and $\widehat{\gamma}$ is

$$\alpha = \widehat{\alpha}, \quad \beta = \widehat{\beta} \cup \{|\widehat{\alpha}| + |\widehat{\beta}| + 1\}, \quad \text{and} \quad \gamma = \{i + 1 : \ i \in \widehat{\gamma}\}.$$

We define $\overline{W}$ by

$$\overline{W} := \begin{bmatrix} W_{\widehat{\alpha}} & W_{\widehat{\beta}} & 0 & W_{\widehat{\gamma}} \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} \overline{W}_\alpha & \overline{W}_\beta & \overline{W}_\gamma \end{bmatrix}, \tag{44}$$

where

$$\overline{W}_\alpha := \begin{bmatrix} W_{\widehat{\alpha}} \\ 0 \end{bmatrix}, \quad \overline{W}_\beta := \begin{bmatrix} W_{\widehat{\beta}} & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad \overline{W}_\gamma := \begin{bmatrix} W_{\widehat{\gamma}} \\ 0 \end{bmatrix}.$$

We then arrive at

$$Y = Q\overline{Y}Q = Q\overline{W}\Lambda\overline{W}^T Q.$$

Therefore, the matrix P in (33) can be chosen to satisfy

$$P = Q\overline{W} \quad \text{and} \quad \overline{W} \text{ is defined by (44).} \tag{45}$$

3.3. Structure of $\partial\Pi_{\mathcal{S}^n_+}(Y)(J\mathrm{Diag}(h)J)$. We let

$$H := J\mathrm{Diag}(h)J \qquad \text{and} \qquad \underline{H} := Q\mathrm{Diag}(h)Q =: \begin{bmatrix} \underline{H}_1 & \underline{h} \\ \underline{h}^T & \underline{h}_0 \end{bmatrix}, \tag{46}$$

where $\underline{H}_1 \in \mathcal{S}^{n-1}$. By the identity in (38), we have

$$H = Q \begin{bmatrix} \underline{H}_1 & 0 \\ 0 & 0 \end{bmatrix} Q, \qquad \text{and hence} \qquad QHQ = \begin{bmatrix} \underline{H}_1 & 0 \\ 0 & 0 \end{bmatrix}.$$

We also note from (45) that

$$P_\alpha = Q\overline{W}_\alpha, \quad P_\beta = Q\overline{W}_\beta, \quad \text{and} \quad P_\gamma = Q\overline{W}_\gamma.$$

We also recall from Proposition 3.1 that $\widetilde{H} = P^T H P$. It follows that

$$\widetilde{H}_{\alpha\alpha} = P_\alpha^T H P_\alpha = \overline{W}_\alpha^T Q H Q \overline{W}_\alpha = W_{\widehat{\alpha}}^T \underline{H}_1 W_{\widehat{\alpha}}.$$

Similarly, we can calculate

$$\widetilde{H}_{\alpha\beta} = \begin{bmatrix} W_{\widehat{\alpha}}^T \underline{H}_1 W_{\widehat{\beta}} & 0 \end{bmatrix}, \qquad \widetilde{H}_{\alpha\gamma} = W_{\widehat{\alpha}}^T \underline{H}_1 W_{\widehat{\gamma}}$$

and

$$\widetilde{H}_{\beta\beta} = \begin{bmatrix} W_{\widehat{\beta}}^T \underline{H}_1 W_{\widehat{\beta}} & 0 \\ 0 & 0 \end{bmatrix}, \qquad \widetilde{H}_{\gamma\gamma} = W_{\widehat{\gamma}}^T \underline{H}_1 W_{\widehat{\gamma}}.$$

We have now completed our preparation to describe any element in $\partial\Pi_{\mathcal{S}^n_+}(Y)(J\mathrm{Diag}(h)J)$ for $h \in \mathbb{R}^n$. The description uses only the spectral information of $\widehat{Y}_1$ in (41) and $\underline{H}_1$ defined in (46). We note that $\underline{h}$ and $\underline{h}_0$ in $\underline{H}$ of (46) do not appear in our description, which we state as a proposition.

PROPOSITION 3.2. *For any $y \in \mathbb{R}^n$, let $Y := -J(D + \mathrm{Diag}(y))J$, which assumes the spectral decomposition* (33)*. Let matrix $\Omega \in \mathcal{S}^n$ be as defined in* (34)*. Let $H := J\mathrm{Diag}(h)J$ for a given $h \in \mathbb{R}^n$. Then a matrix $L \in \partial\Pi_{\mathcal{S}^n_+}(Y)(H)$ if and only if there exists $\widetilde{V} \in \partial\Pi_{\mathcal{S}^{|\beta|}_+}(0)$ such that*

$$L = P\mathcal{W}(H)P^T, \tag{47}$$

where P is defined by (45) *and*

(48)

$$\mathcal{W}(H) := \begin{bmatrix} W_{\widehat{\alpha}}^T \underline{H}_1 W_{\widehat{\alpha}} & \begin{bmatrix} W_{\widehat{\alpha}}^T \underline{H}_1 W_{\widehat{\beta}} & 0 \end{bmatrix} & \Omega_{\alpha\gamma} \circ W_{\widehat{\alpha}}^T \underline{H}_1 W_{\widehat{\gamma}} \\ \begin{bmatrix} W_{\widehat{\beta}}^T \underline{H}_1 W_{\widehat{\alpha}} \\ 0 \end{bmatrix} & \widetilde{V}\left(\begin{bmatrix} W_{\widehat{\beta}}^T \underline{H}_1 W_{\widehat{\beta}} & 0 \\ 0 & 0 \end{bmatrix}\right) & 0 \\ \Omega_{\alpha\gamma}^T \circ W_{\widehat{\gamma}}^T \underline{H}_1 W_{\widehat{\alpha}} & 0 & 0 \end{bmatrix}.$$

Proof. This result is just a new interpretation of the formula in Proposition 3.1 in terms of the above calculations. □

This proposition will be used to study the nonsingularity of $\partial^2\theta(y)$ in the next section. Let us first list two facts that will be used there.

The first fact is a simple observation on $\|h\|$ for $h \in \mathbb{R}^n$:

$$\|h\|^2 = \|\mathrm{Diag}(h)\|^2 = \|P^T\mathrm{Diag}(h)P\|^2 = \|\overline{W}^T Q\mathrm{Diag}(h)Q\overline{W}\|^2 = \|\overline{W}^T \underline{H}\overline{W}\|^2. \tag{49}$$

The second fact is about an inequality. Let

$$G_\beta := \begin{bmatrix} W_{\widehat{\beta}}^T \underline{H}_1 W_{\widehat{\beta}} & W_{\widehat{\beta}}^T \underline{h} \\ \underline{h}^T W_{\widehat{\beta}} & \underline{h}_0 \end{bmatrix}, \qquad G_{\widehat{\beta}} := \begin{bmatrix} W_{\widehat{\beta}}^T \underline{H}_1 W_{\widehat{\beta}} & 0 \\ 0 & 0 \end{bmatrix}.$$

It is easy to see that $\|G_{\widehat{\beta}}\| \le \|G_\beta\|$ and

$$\|G_\beta\|^2 - \|G_{\widehat{\beta}}\|^2 = 2\|W_{\widehat{\beta}}^T \underline{h}\|^2 + \underline{h}_0^2.$$

Hence we have

$$\|G_\beta\|(\|G_\beta\| - \|G_{\widehat{\beta}}\|) \ge \|W_{\widehat{\beta}}^T \underline{h}\|^2 + \frac{1}{2}\underline{h}_0^2. \tag{50}$$

3.4. Nonsingularity of $\partial^2\theta(y)$. We are ready to prove the following result.

THEOREM 3.3. *Let y be an optimal solution of the dual problem* (8). *Then every matrix $M \in \partial^2\theta(y)$ is positive definite.*

Proof. We continue to use the notation developed so far. Let $M \in \partial^2\theta(y)$. Our purpose is to prove $\langle h,\ Mh\rangle > 0$ for all $0 \ne h \in \mathbb{R}^n$. For such h, we recall that $H := J\mathrm{Diag}(h)J$ and $Y := -J(D + \mathrm{Diag}(y))J$. It follows from (32) and (47) that there exists $\widetilde{V} \in \partial\Pi_{\mathcal{S}_+^{|\beta|}}(Y)$ satisfying

$$Mh = h - \mathrm{diag}\Big(P\mathcal{W}(H)P^T\Big),$$

where $\mathcal{W}(H)$ is defined by (48).

We now calculate $\langle h,\ Mh\rangle$:

$$\begin{aligned}
\langle h,\ Mh\rangle &= \|h\|^2 - \langle \mathrm{Diag}(h),\ P\mathcal{W}(H)P^T\rangle = \|h\|^2 - \langle P^T\mathrm{Diag}(h)P,\ \mathcal{W}(H)\rangle \\
&= \|h\|^2 - \langle \overline{W}^T Q\mathrm{Diag}(h)Q\overline{W},\ \mathcal{W}(H)\rangle \qquad \text{(by (45))} \\
&= \|\overline{W}^T \underline{H}\overline{W}\|^2 - \langle \overline{W}^T \underline{H}\overline{W},\ \mathcal{W}(H)\rangle \qquad \text{(by (49), (46))} \\
&= 2\left\{\|W_{\widehat{\alpha}}^T \underline{h}\|^2 + \|W_{\widehat{\alpha}}^T \underline{H}_1 W_{\widehat{\gamma}}\|^2 - \langle W_{\widehat{\alpha}}^T \underline{H}_1 W_{\widehat{\gamma}},\ \Omega_{\alpha\gamma} \circ (W_{\widehat{\alpha}}^T \underline{H}_1 W_{\widehat{\gamma}})\rangle\right\} \\
&\quad + 2\left\{\|W_{\widehat{\beta}}^T \underline{H}_1 W_{\widehat{\gamma}}\|^2 + \|W_{\widehat{\gamma}}^T \underline{h}\|^2 + \|W_{\widehat{\gamma}}^T \underline{H}_1 W_{\widehat{\gamma}}\|^2/2\right\} \\
&\quad + \|G_\beta\|^2 - \langle G_\beta,\ \widetilde{V}(G_{\widehat{\beta}})\rangle.
\end{aligned}$$

The last equality made use of (48) and the structure of $\overline{W}^T \underline{H}\overline{W}$:

$$\overline{W}^T \underline{H}\overline{W} = \begin{bmatrix} W_{\widehat{\alpha}}^T \underline{H}_1 W_{\widehat{\alpha}} & W_{\widehat{\alpha}}^T \underline{H}_1 W_{\widehat{\beta}} & W_{\widehat{\alpha}}^T \underline{h} & W_{\widehat{\alpha}}^T \underline{H}_1 W_{\widehat{\gamma}} \\ W_{\widehat{\beta}}^T \underline{H}_1 W_{\widehat{\alpha}} & W_{\widehat{\beta}}^T \underline{H}_1 W_{\widehat{\beta}} & W_{\widehat{\beta}}^T \underline{h} & W_{\widehat{\beta}}^T \underline{H}_1 W_{\widehat{\gamma}} \\ \underline{h}^T W_{\widehat{\alpha}} & \underline{h}^T W_{\widehat{\beta}} & \underline{h}_0 & \underline{h}^T W_{\widehat{\gamma}} \\ W_{\widehat{\gamma}}^T \underline{H}_1 W_{\widehat{\alpha}} & W_{\widehat{\gamma}}^T \underline{H}_1 W_{\widehat{\beta}} & W_{\widehat{\gamma}}^T \underline{h} & W_{\widehat{\gamma}}^T \underline{H}_1 W_{\widehat{\gamma}} \end{bmatrix}.$$

Define $\tau_{\max} := \max_{i\in\alpha, j\in\gamma}\ \Omega_{ij}$. By (34), $0 < \tau_{\max} < 1$. We continue to simplify $\langle h, Mh\rangle$:

$$\begin{aligned}\langle h,\ Mh\rangle &\geq 2\left\{\|W_{\widehat{\alpha}}^T\underline{h}\|^2 + \|W_{\widehat{\gamma}}^T\underline{h}\|^2 + \|W_{\widehat{\beta}}^T\underline{H}_1W_{\widehat{\gamma}}\|^2 + (1-\tau_{\max})\|W_{\widehat{\alpha}}^T\underline{H}_1W_{\widehat{\gamma}}\|^2\right\}\\ &\quad + \|W_{\widehat{\gamma}}^T\underline{H}_1W_{\widehat{\gamma}}\|^2 + \|G_\beta\|^2 - \|G_\beta\|\|G_{\widehat{\beta}}\| \qquad \text{(by (37))}\\ &\geq 2\left\{\|W_{\widehat{\alpha}}^T\underline{h}\|^2 + \|W_{\widehat{\gamma}}^T\underline{h}\|^2 + \frac{1}{2}\|W_{\widehat{\beta}}^T\underline{h}\|^2\right\} + \|W_{\widehat{\gamma}}^T\underline{H}_1W_{\widehat{\gamma}}\|^2\\ &\quad + 2\left\{(1-\tau_{\max})\|W_{\widehat{\alpha}}^T\underline{H}_1W_{\widehat{\gamma}}\|^2 + \|W_{\widehat{\beta}}^T\underline{H}_1W_{\widehat{\gamma}}\|^2\right\} + \frac{1}{2}\underline{h}_0^2 \qquad \text{(by (50))}\\ &\geq 0. \qquad (51)\end{aligned}$$

Hence, the assumption $\langle h,\ Mh\rangle = 0$ would imply

$$W_{\widehat{\alpha}}^T\underline{h} = 0, \quad W_{\widehat{\beta}}^T\underline{h} = 0, \quad W_{\widehat{\gamma}}^T\underline{h} = 0, \quad \underline{h}_0 = 0 \qquad (52)$$

and

$$W_{\widehat{\alpha}}^T\underline{H}_1W_{\widehat{\gamma}} = 0, \quad W_{\widehat{\beta}}^T\underline{H}_1W_{\widehat{\gamma}} = 0, \quad W_{\widehat{\gamma}}^T\underline{H}_1W_{\widehat{\gamma}} = 0. \qquad (53)$$

Because of (43) and the nonsingularity of W, (52) implies

$$\underline{h} = 0 \qquad \text{and} \qquad \underline{h}_0 = 0. \qquad (54)$$

Since y is an optimal solution of (8), $\overline{A} := \Pi_{\mathcal{K}_+^n}(D + \mathrm{Diag}(y))$ is the optimal solution of (3) by (10). Obviously, $\overline{A}$ is feasible with respect to the constraints of (3). Constraint nondegeneracy holds at $\overline{A}$ due to Theorem 2.3. We assume $\overline{A}$ is decomposed as in (19). We now clarify the structure of matrix Z. We recall the decomposition (40) and Hayden–Wells formula (15) for $\Pi_{\mathcal{K}_+^n}$. It follows that

$$\overline{A} = Q\begin{bmatrix}\Pi_{\mathcal{S}_+^{n-1}}(-\widehat{Y}_1) & -\widehat{y}\\ -\widehat{y}^T & -\widehat{y}_0\end{bmatrix}Q = Q\begin{bmatrix}W\Pi_{\mathcal{S}_+^{n-1}}(-\widehat{\Lambda})W^T & -\widehat{y}\\ -\widehat{y}^T & -\widehat{y}_0\end{bmatrix}Q.$$

Hence, the matrix Z in (19) has the form

$$Z = W\Pi_{\mathcal{S}_+^{n-1}}(-\widehat{\Lambda})W^T = W\begin{bmatrix}0_{\widehat{\alpha}} & & \\ & 0_{\widehat{\beta}} & \\ & & -\widehat{\Lambda}_{\widehat{\gamma}}\end{bmatrix}W^T. \qquad (55)$$

Now we recall the definitions of $\alpha(Z)$ and $\overline{\alpha}(Z)$ in (26). The matrix Z as decomposed in (20) also takes the following form:

$$Z = U\begin{bmatrix}\Lambda_\alpha & & \\ & 0 & \\ & & 0\end{bmatrix}U^T. \qquad (56)$$

We note that the indices in $\widehat{\gamma}$ are

$$\widehat{\gamma} = \left\{|\widehat{\alpha}| + |\widehat{\beta}| + 1, \ldots, |\widehat{\alpha}| + |\widehat{\beta}| + |\widehat{\gamma}|\right\} \qquad \text{with} \qquad |\widehat{\alpha}| + |\widehat{\beta}| + |\widehat{\gamma}| = n - 1.$$

By comparing (55) with (56), we have the following correspondence between eigenvalues indexed by α and those by $\widehat{\gamma}$:

$$|\alpha| = |\widehat{\gamma}| \qquad \text{and} \qquad \lambda_i = -\widehat{\lambda}_{n-i} \ \text{ for } \ i \in \alpha = \{1, 2, \ldots, |\alpha|\}.$$

Let $\widetilde{P}$ be the permutation matrix that maps the order sequence $\{1, 2, \ldots, |\widehat{\gamma}|\}$ to its reverse order sequence $\{|\widehat{\gamma}|, \ldots, 2, 1\}$. Then we have the following correspondence between eigenvectors in U and those in W:

$$U_{\alpha(Z)} = W_{\widehat{\gamma}}\widetilde{P} \qquad \text{and} \qquad U_{\overline{\alpha}(Z)} = W_{\overline{\gamma}} \qquad \text{with} \qquad \overline{\gamma} := \widehat{\alpha} \cup \widehat{\beta}.$$

Then the matrices $\{C_i\}_{i=1}^n$ defined in (27) are linearly independent by Proposition 2.4. It follows from (46), (54), and (30) that

$$\underline{H} = Q\mathrm{Diag}(h)Q = \begin{bmatrix} \underline{H}_1 & \underline{h} \\ \underline{h}^T & \underline{h}_0 \end{bmatrix} = \begin{bmatrix} \underline{H}_1 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \sum_{i=1}^n h_i B_1^i = \underline{H}_1.$$

We recall that $\overline{U}$ is defined in (21). Now we consider the linear combination $\sum h_i C_i$, which has been derived in (29):

$$\begin{aligned} \sum_{i=1}^n h_i C_i &= \overline{U}^T \Big(Q\mathrm{Diag}(h)Q \Big) \overline{U} - \begin{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & U_{\overline{\alpha}}^T \sum (h_i B_1^i) U_{\overline{\alpha}} \end{bmatrix} & 0 \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} U_\alpha^T \underline{H}_1 U_\alpha & U_\alpha^T \underline{H}_1 U_{\overline{\alpha}} & 0 \\ U_{\overline{\alpha}}^T \underline{H}_1 U_\alpha & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \widetilde{P}^T W_{\widehat{\gamma}}^T \underline{H}_1 W_{\widehat{\gamma}} \widetilde{P} & \widetilde{P}^T W_{\widehat{\gamma}}^T \underline{H}_1 W_{\overline{\gamma}} & 0 \\ W_{\overline{\gamma}}^T \underline{H}_1 W_{\widehat{\gamma}} \widetilde{P} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \end{aligned}$$

Consequently, (53) forces $\sum h_i C_i = 0$. The linear independence of $\{C_i\}$ in turn forces $h = 0$. Therefore, $\langle h,\ Mh \rangle = 0$ if and only if $h = 0$. In other words, $\langle h,\ Mh \rangle > 0$ for $0 \neq h \in \mathbb{R}^n$ by (51). This proves that M is positive definite. □

We now state some consequences of Theorem 3.3. The first is on the uniqueness of the optimal solution of the dual problem (8). Let us regard the gradient function $F(y) = \nabla\theta(y)$ as a mapping from $\mathbb{R}^n$ to $\mathbb{R}^n$. The generalized Jacobian $\partial F(y)$ is said to be of *maximal rank* provided that every matrix M in $\partial F(y)$ is of maximal rank (i.e., nonsingular) [11, p. 253]. It follows from Theorem 3.3 that $\partial F(y^*)$ is of maximal rank provided that y^* is an optimal solution of (8). Then, the inverse function theorem of Clarke [11, Thm. 7.1.1] and the convexity of (8) lead to the following result.

COROLLARY 3.4. *The dual problem* (8) *has a unique optimal solution.*

The second consequence of Theorem 3.3 is about the quadratic convergence of the Newton method (11). We state it as a theorem.

THEOREM 3.5. *The Newton method* (11) *is quadratically convergent provided that* y^0 *is sufficiently close to the unique optimal solution* y^* *of* (8).

Proof. In the quadratic convergence-rate theorem of Qi and Sun [39, Thm. 3.2] for general semismooth Newton methods, there are three conditions: (i) The function F is strongly semismooth, which is true for our case because it is a composition of linear mappings and the strongly semismooth mapping $\Pi_{\mathcal{S}_+^n}(\cdot)$ [44]. (ii) Every matrix in the generalized Jacobian of $\partial F(y^*)$ is nonsingular, which has been proved in Theorem 3.3. The last condition is that the initial point y^0 stays close to y^*. This proves our result. □

Since (8) is convex, globalization of the Newton method (11) is an easy task. We simply use one of the well-developed globalization method studied by Qi and Sun [38] in our numerical experiment.

4. Application to the *H*-weighted problem. As briefly mentioned in introduction, the H-weighted problem (4) is much more difficult to solve than the unweighted case (3). In this section, we develop a global method for this difficult problem. The most important feature of this method is that each subproblem is a diagonally weighted problem of (3), and this subproblem can be efficiently solved by a Newton method similar to (11). The bridge that links the H-weighted problem and the diagonally weighted problem is the *majorization* approach introduced by Gao and Sun [18] for the H-weighted nearest correlation matrix problem. We refer to [18] for more information about the majorization approach initially used in multidimensional scaling. We will first demonstrate how this approach works for (4).

4.1. The majorization approach. Denote the objective function in (4) by

$$f(X) = 0.5\|H \circ (X - D)\|^2.$$

Obviously, $f(\cdot)$ is quadratic and its Taylor expansion at a given point $Y^k \in \mathcal{S}^n$ is

$$f(X) = f(Y^k) + \langle H \circ H \circ (Y^k - D), X - Y^k\rangle + 0.5\|H \circ (X - Y^k)\|^2.$$

We replace the quadratic term by a simpler function $\|W^{1/2}(X - Y^k)W^{1/2}\|^2$, which satisfies

$$\|W^{1/2}(X - Y^k)W^{1/2}\| \geq \|H \circ (X - Y^k)\| \qquad \forall\ X \in \mathcal{S}^n,$$

where $W := \mathrm{Diag}(w)$ and $0 < w \in \mathbb{R}^n$. A particular choice recommended by Gao and Sun [18] is

$$w_i := \max\{\tau,\ \max\{H_{ij} :\ j = 1, \ldots, n\}\}, \qquad i = 1, \ldots, n, \tag{57}$$

where $\tau > 0$ is a constant. Define

$$f_k(X) := f(Y^k) + \langle H \circ H \circ (Y^k - D), X - Y^k\rangle + 0.5\|W^{1/2}(X - Y^k)W^{1/2}\|^2. \tag{58}$$

We certainly have the property

$$f_k(X) \geq f(X) \qquad \forall\ X \in \mathcal{S}^n \qquad \text{and} \qquad f_k(Y^k) = f(Y^k). \tag{59}$$

Because of this property, $f_k(X)$ is called a majorization of f at Y^k. The majorization approach aims to solve the following problem:

$$\min\ f_k(X) \qquad \text{s.t.}\ \ X \in \mathcal{S}^n_h \cap \mathcal{K}^n_+. \tag{60}$$

We note that problem (60) is strictly convex, and it has a unique solution, denoted by X^{k+1}. We then have (because of (59))

$$f(X^{k+1}) \leq f_k(X^{k+1}) \leq f_k(Y^k) = f(Y^k). \tag{61}$$

In other words, the solution of (60) provides a better point X^{k+1} than Y^k in terms of the original objective function. When Y^k is chosen to be X^k, property (61) is known as the *sandwich* property, and the majorization approach produces a sequence $\{X^k\}$ satisfying $f(X^{k+1}) \leq f(X^k)$.

Numerical implication of the majorization approach is then to solve a sequence of the problem (60) starting from X^0. Theoretically, we get a sequence of $\{X^k\}$ with decreasing function values if we choose $Y^k = X^k$. As a matter of fact, there are other (better) choices for Y^k. We will describe one in section 4.3. Numerically, this approach is sensible only if the new problem (60) is much easier to solve than the original problem. We demonstrate below that this is the case.

4.2. Solving subproblem (60). It is observed that problem (60) is actually a diagonally weighted problem of (3). To see this, we note that

$$f_k(X) = \frac{1}{2}\|W^{1/2}(X-(Y^k-D^k))W^{1/2}\|^2 \\ +f(Y^k)-\frac{1}{2}\|W^{-1/2}(H\circ H\circ(Y^k-D))W^{-1/2}\|^2,$$

where $D^k := W^{-1}(H\circ H\circ(Y^k-D))W^{-1}$. Ignoring the constant term in f_k, problem (60) is equivalent to

$$\min\ \frac{1}{2}\|W^{1/2}(X-\overline{D}^k)W^{1/2}\|^2 \qquad \text{s.t.}\ \ X\in\mathcal{S}_h^n\cap\mathcal{K}_+^n, \tag{62}$$

where $\overline{D}^k := Y^k - D^k$. Because of $W=\mathrm{Diag}(w)$, we call this problem a diagonally weighted version of problem (3).

Let

$$\widetilde{X} := W^{1/2}XW^{1/2} \qquad \text{and} \qquad \widetilde{D}^k := W^{1/2}\overline{D}^kW^{1/2}.$$

Then problem (62) is equivalent to

$$\min\frac{1}{2}\|\widetilde{X}-\widetilde{D}^k\|^2 \qquad \text{s.t.}\quad W^{-1/2}\widetilde{X}W^{-1/2}\in\mathcal{S}_h^n\cap\mathcal{K}_+^n. \tag{63}$$

It is easy to verify that (because W is diagonal)

$$W^{-1/2}\widetilde{X}W^{-1/2}\in\mathcal{S}_h^n \qquad \text{if and only if} \qquad \widetilde{X}\in\mathcal{S}_h^n$$

and

$$W^{-1/2}\widetilde{X}W^{-1/2}\succeq 0\ \ \text{on}\ \ \{e\}^\perp \qquad \text{if and only if} \qquad \widetilde{X}\succeq 0\ \ \text{on}\ \ \left\{W^{1/2}e\right\}^\perp.$$

Define the closed convex cone as

$$\mathcal{K}_w^n := \left\{X\in\mathcal{S}^n:\ \ X\succeq 0\ \ \text{on}\ \ \left\{W^{1/2}e\right\}^\perp\right\}.$$

It follows from (63) that (62) is equivalent to

$$\min\frac{1}{2}\|\widetilde{X}-\widetilde{D}^k\|^2 \qquad \text{s.t.}\quad \widetilde{X}\in\mathcal{S}_h^n\cap\mathcal{K}_w^n. \tag{64}$$

This problem is almost the same as (3) except that $\mathcal{K}_+^n$ is being replaced by $\mathcal{K}_w^n$. We can develop the Newton method for this problem just as we have done for problem (3). We summarize this procedure below.

The corresponding dual problem and its first-order optimality condition are (see (8) and (9), respectively, for problem (3)):

$$\min_{y\in\mathbb{R}^n}\ \theta_w(y) := \frac{1}{2}\|\Pi_{\mathcal{K}_w^n}(\widetilde{D}^k+\mathrm{Diag}(y))\|^2-\frac{1}{2}\|\widetilde{D}^k\|^2 \tag{65}$$

and

$$F_w(y) := \nabla\theta_w(y) = \mathrm{diag}\Big(\Pi_{\mathcal{K}_w^n}(\widetilde{D}^k+\mathrm{Diag}(y))\Big) = 0.$$

The Newton method therefore takes the following form (see (11)):

$$y^{j+1} = y^j - V_j^{-1} F_w(y^j), \qquad j = 0, 1, 2, \ldots, \tag{66}$$

where $V_j \in \partial F_w(y^j)$.

In order to implement the Newton method (66), we need to characterize the projection $\Pi_{\mathcal{K}_w^n}(A)$ for any $A \in \mathcal{S}^n$. This can be done as follows. Let Q be the Householder transformation that maps the vector $W^{1/2}e$ to $[0, \ldots, 0, -\|W^{1/2}e\|]^T$. Let

$$v := \left[\sqrt{w_1}, \ldots, \sqrt{w_{n-1}}, \sqrt{w_n} + \sqrt{\sum_{i=1}^n w_i} \right]^T.$$

Then

$$Q = I - \frac{2}{v^T v} v v^T.$$

According to [25, Thm. 2.1] (which takes $S = W^{1/2}e$), we have

$$\Pi_{\mathcal{K}_w^n}(A) = Q \begin{bmatrix} \Pi_{\mathcal{S}_+^{n-1}}(\widehat{A}_1) & \widehat{a} \\ \widehat{a}^T & \widehat{a}_0 \end{bmatrix} Q \qquad \forall\ A \in \mathcal{S}^n,$$

where

$$QAQ =: \begin{bmatrix} \widehat{A}_1 & \widehat{a} \\ \widehat{a}^T & \widehat{a}_0 \end{bmatrix} \qquad \text{with} \quad \widehat{A}_1 \in \mathcal{S}^{n-1},\ \widehat{a} \in \mathbb{R}^{n-1},\ \widehat{a}_0 \in \mathbb{R}.$$

Let

$$J := Q \begin{bmatrix} I_{n-1} & 0 \\ 0 & 0 \end{bmatrix} Q = I - \frac{1}{\sum w_i} \sqrt{w}\sqrt{w}^T \qquad \text{with} \quad \sqrt{w} := \begin{bmatrix} \sqrt{w_1} \\ \vdots \\ \sqrt{w_n} \end{bmatrix}.$$

Then the corresponding formula (17) of Gaffke and Mathar is

$$\Pi_{\mathcal{K}_w^n}(A) = A + \Pi_{\mathcal{S}_+^n}(-JAJ).$$

We can repeat the analysis conducted in sections 2.2 and 3 to conclude that the Newton method (66) is quadratically convergent (see Theorem 3.5). We omit the details.

4.3. Global method for the *H*-weighted problem. Having addressed the quadratic convergence of the Newton method (66), we are ready to formally state our global method for the H-weighted problem (4). There exist a couple of globalization strategies. One is to follow the algorithmic framework of Gao and Sun [18] based on the majorization argument (e.g., choose $Y^k = X^k$). Another is to cast the majorization function f_k as a proximal gradient approximation to the objective function f, and hence the resulting proximal gradient method can be applied. Our numerical results (not reported here) suggest that the second strategy works better. In this paper, we adopt the accelerated proximal gradient (APG) method (e.g., choose Y^k to be an extrapolation of the two iterates X^k and X^{k-1}), recently studied by Jiang et al.

[29] for large scale linearly constrained convex quadratic semidefinite programming (QSDP). We describe this method below.

ALGORITHM 4.1. Inexact accelerated proximal gradient method (IAPG).

Step 1. *Choose* $X^0 \in \mathcal{S}^n$. *Let* $Y^0 := X^0$, $t_0 := 1$. *Set* $k := 0$.

Step 2. *Define the function* f_k *by* (58). *Find an approximate minimizer* X^{k+1} *of the problem* (60).

Step 3. *Compute* $t_{k+1} := (1 + \sqrt{1 + 4t_k^2})/2$.

Step 4. *Compute*

$$Y^{k+1} := X^{k+1} + \left(\frac{t_k - 1}{t_{k+1}}\right)(X^{k+1} - X^k).$$

Let $k := k + 1$ *and return to Step* 2.

We have the following remarks regarding this algorithm.

(R1) An obvious choice of the initial point X^0 is obtained by solving the unweighted problem (3). If we keep $t_k = 1$ instead of updating t_k in Step 3, Algorithm 4.1 becomes the majorization method of Gao and Sun [18].

(R2) The subproblem (60) is solved by the Newton method (66). Let y^{k+1} be the approximate solution obtained by this Newton method. Then X^{k+1} is given by

$$X^{k+1} := W^{-1/2}\widetilde{X}^{k+1}W^{-1/2} \quad \text{and} \quad \widetilde{X}^{k+1} := \Pi_{\mathcal{K}_w^n}(\widetilde{D}^k + \mathrm{Diag}(y^{k+1})).$$

We also calculate

$$Z^{k+1} := W^{1/2}(\widetilde{X}^{k+1} - (\widetilde{D}^k + \mathrm{Diag}(y^{k+1})))W^{1/2}.$$

If y^{k+1} is the exact solution of the dual problem (65), then X^{k+1} is the optimal solution of (62), and (Wy^{k+1}, Z^{k+1}) are the Lagrange multipliers corresponding to the two constraints in (62). Consequently, $\nabla f_k(X^{k+1}) - \mathrm{Diag}(Wy^{k+1}) - Z^{k+1} = 0$. Since the dual problem (65) is solved approximately, X^{k+1} is only an approximate solution of (62), and (Wy^{k+1}, Z^{k+1}) are approximate multipliers. This is why the method is called the inexact APG (see [29] for more detailed justification of why only solving the subproblem to certain accuracy is still adequate to obtain the desired complexity result described in (R4)).

(R3) The level of the accuracy for each subproblem can be specified, but with involved formulations. The interested reader can refer to [29, eq. (32)] for those formulations. We simply use the stopping criterion, proposed by [29], to terminate the Newton method (66) for the dual problem (65) when

$$\|\nabla\theta_w(y^{k+1})\| \le \min\left\{1/t_k^{3.1},\ 0.2\|\nabla f_k(X^k) - \mathrm{Diag}(Wy^k) - Z^k\|\right\}.$$

Because t_k increases to ∞, the approximate solution y^{k+1} (and hence X^{k+1}) becomes more and more accurate as the algorithm progresses.

(R4) It has become well known (because of the work [35, 4, 46] and many others) that APGs possess the following complexity result:

$$f(X^k) - f(X^*) = O(1/k^2),$$

where X^* is an optimal solution of (4). For the inexact APG described in Algorithm 4.1, the above complexity still holds (see [29, Thm. 3.1]) subject

to the subproblem being solved to certain accuracy. We omit the details. Our main contribution in applying IAPG is that the subproblem (60) can be efficiently solved by applying the Newton method (66) to its dual problem (65).

(R5) Without a proper convergence check, Algorithm 4.1 may lead to an infinite loop. In our implementation, we used the stopping criterion (69) to terminate Algorithm 4.1.

5. Numerical results. In this section, we conduct numerical tests on both problem (3) and the H-weighted problem (4). For the former we use the Newton method (11), and for the latter we use Algorithm 4.1. At each of its iterations, Algorithm 4.1 uses the Newton method (66) to solve its subproblem (62). Since both Newton methods in their current forms are only locally quadratically convergent, we used a globalized version of each of the Newton methods in our implementation. The globalized version we used is taken from [38, Alg. 5.1]. This globalized Newton method is globally and quadratically convergent (see the convergence analysis in [38, Sect. 5]).

We just like to make three remarks about this globalized Newton method. We use (11) as an example. The first remark is about calculating the matrix V_k. This can be done by adapting the computing procedure of [38, sect. 5(a)] to our function F. We summarize the calculation in a lemma. For simplicity, we drop the iteration index k on y. We also recall that E is the matrix of all ones.

LEMMA 5.1 (computing $V_y \in \partial F(y)$). *Let $Y := -J(D + \mathrm{Diag}(y))J$ have the spectral decomposition* (33)*, with index sets α, β, and γ being defined by* (35)*. Then a matrix $V_y \in \partial F(y)$ can be computed as*

$$V_y h = h - \mathrm{diag}\big(P(M_y \circ (P^T H P))P^T\big) \qquad \forall\ h \in \mathbb{R}^n,$$

where $H := J\mathrm{Diag}(h)J$ and M_y is defined by

$$M_y := \begin{pmatrix} E_{\alpha\alpha} & E_{\alpha\beta} & (\tau_{ij}(y))_{\substack{i\in\alpha\\ j\in\gamma}} \\ E_{\beta\alpha} & 0 & 0 \\ (\tau_{ji}(y))_{\substack{i\in\alpha\\ j\in\gamma}} & 0 & 0 \end{pmatrix}, \qquad \tau_{ij}(y) := \frac{\lambda_i}{\lambda_i - \lambda_j}, \quad i \in \alpha, j \in \gamma.$$

Evaluating the explicit form of V_y costs a prohibitive $O(n^4)$ operations. We therefore choose the conjugate gradient (CG) method [27], which requires matrix-vector products only and avoids computing the explicit form of V_y, to solve the Newton equation (11). The second remark is about preconditioning CG by the diagonal preconditioner of V_y. The preconditioner can be calculated by adapting the computing procedure of [7, sect. 3.2] for the problem (7) to our case. The computational complexity is about $2n^3$, similar to that of [7].

Our last remark is about extending the Newton method to handle additional fixed distance constraints:

$$X_{ij} = D_{ij} \qquad \text{for} \quad (i,j) \in \mathcal{B}, \tag{67}$$

where $\mathcal{B}$ is the index set that fixes those known distances D_{ij}. Toh [45] included such constraints in solving (5). Our test Example 5.6 considers such additional constraints. Our methodology and computation can be extended to this case in a natural way. To see this, let $\mathcal{A}^o : \mathcal{S}^n \mapsto \mathbb{R}^{|\mathcal{B}|}$ denote the linear mapping that fixes the off-diagonal distances in (67) indexed by $\mathcal{B}$. We further let

$$b := \big(D_{ij}\big)_{(i,j)\in\mathcal{B}} \in \mathbb{R}^{|\mathcal{B}|} \qquad \text{and} \qquad \mathcal{A}(X) := \begin{pmatrix} \mathrm{diag}(X) \\ \mathcal{A}^o(X) \end{pmatrix}.$$

For two column vectors y and z, we use MATLAB notation $(y; z)$ to denote the column vector $(y^T, z^T)^T$. The nearest EDM problem (see (3)) that has the extra constraints (67) becomes

$$\min \ \|D - X\|^2/2 \qquad \text{s.t.} \quad \mathcal{A}(X) = (0;\ b) \quad \text{and} \quad X \in \mathcal{K}_+^n.$$

The corresponding dual problem becomes (see (8))

$$\min_{(y;z)\in\mathbb{R}^{n+|\mathcal{B}|}} \ \theta(y, z) := \|\Pi_{\mathcal{K}_+^n}(D + \mathcal{A}^*(y; z))\|^2/2 - \langle b,\ z\rangle - \|D\|^2/2,$$

with $\mathcal{A}^*$ being the adjoint of $\mathcal{A}$. The first-order optimality condition becomes (see (9))

$$F(y; z) := \nabla\theta(y; z) = \mathcal{A}\Big(\Pi_{\mathcal{K}_+^n}(D + \mathcal{A}^*(y; z))\Big) - (0;\ b) = (0;\ 0).$$

The Newton method becomes (see (11))

$$(y^{k+1}; z^{k+1}) = (y^k; z^k) - V_k^{-1}F(y^k; z^k), \qquad k = 1, 2, \ldots,$$

where $V_k \in \partial F(y^k; z^k)$. The new dual problem, as well as the function $F(y; z)$, is structurally similar to what we have in (8) and (9). As a consequence, the calculation of the generalized Jacobian matrix V_k, which is essential to our Newton method, can be done via Lemma 5.1 with diag being replaced by $\mathcal{A}$ and Diag by $\mathcal{A}^*$. Moreover, the preconditioning CG used to solve the Newton equation goes through without any difficulties. However, if we have too many extra constraints of the type (67), we may lose the property of constraint nondegeneracy, which in turn may destroy the quadratic convergence of the Newton method. We like to point out that it is a very complicated issue to know which constraints enjoy the constraint nondegeneracy and which do not.

We will test the following problems. The first two problems are of dense type, i.e., $D_{ij} \neq 0$ when $i \neq j$, while the remaining three enjoy certain sparsity patterns. The first three problems are the type of unweighted problem (3), and the last two are of H-weighted problem (4). Examples 5.3–5.6 refer to the EDM1 problem of Toh [45].

Example 5.2 (see [19]). The predistance matrix D is randomly generated with values uniformly distributed between 10^{-5} and 10.

Example 5.3. This problem is a slight modification of the EDM1 problem of Toh [45]. First, we generate n random points, $x_1, \ldots, x_n$, in the unit cube centered at the origin in $\mathbb{R}^3$. We calculate $D_{ij} = \|x_i - x_j\|^2$ (the squared distance between x_i and x_j.) We then add to D an $n \times n$ random symmetric matrix with entries in $[-\alpha, \alpha]$, where $\alpha = 0.3$ in our test.

Example 5.4. This is the EDM1 problem of [45] except that the H-weight matrix is taken to be $H = E$. First, we generate n random points, $x_1, \ldots, x_n$, in the unit cube centered at the origin in $\mathbb{R}^3$. Then we set $D_{ij} = \|x_i - x_j\|^2$ if the distance is less than a certain cut-off distance R; otherwise, set $D_{ij} = 0$. $R = 1$ in our test.

Example 5.5. This is the EDM1 problem of [45] except that we do not have fixed distances. Generate matrix D as in Example 5.4 with various choices of R. The weight matrix H is chosen to be the 0-1 matrix having the same sparsity pattern as D. Density is calculated by `nnz(D)/numel(D)`.

Example 5.6. This is the EDM1 problem of [45]. Generate D and H as in Example 5.5. The set of indices where additional distances of the type (67) are fixed is given by $\mathcal{B} := \{(1, j) :\ D_{1j} \neq 0,\ j = 1, \ldots, n\}$.

All tests were carried out using the 64-bit version of MATLAB R2011b on a Windows 7 desktop with a 64-bit operating system having Intel Core 2 Duo CPU of 3.16 GHz, and 4.0 GB of RAM. In Table 1, we compare the Newton method with `MAP` [19] and the `QSDP` solver of Toh [45]. It follows from [17] (see also [31, Thm. 5.1]) that the alternating projection method is actually the gradient method for the dual problem with step size 1. Therefore, the error measured between successive iterates by `MAP` is the norm of the gradient $\|\nabla\theta(y)\|$. Therefore, we terminate `MAP` when

$$\text{Res} := \|\nabla\theta(y^k)\| \le \texttt{tol}, \tag{68}$$

with $\texttt{tol} = 10^{-5}$, and we stop the Newton method when (68) is satisfied with $\texttt{tol} = 10^{-6}$. The reason why we chose 10^{-5} for `MAP` is that it run into difficulties in some cases for higher accuracy (e.g., took too many iterations to have one more digit of accuracy). On the contrary, the Newton method can quickly reach a higher accuracy. This is well reflected by the CPU time (in hh:mm:ss format) and the number of iterations (`Iter` columns) used by the two methods. The starting point for both methods was set to 0, and the maximum iterations of `MAP` are capped at 2000. As for the `QSDP` solver, we used the default parameter settings. The `Obj` column contains returned objective function values by each method. The results reported below are the average on 10 randomly generated instances of each test problem.

The performance of the Newton method on unweighted problems in Table 1 is outstanding. It took under 1 minute to solve problems with $n = 2000$, which is equivalent to about 2 million independent variables in each problem. An interesting observation is that once it reached the level $\|\nabla\theta(y^k)\| \le 10^{-1}$, the Newton method converged at a quadratic rate, taking just a few more steps to reach the required accuracy of 10^{-6}. This observation seems independent of problem size and probably justifies why it took only about 4–8 steps to terminate for all the problems. On the contrary, `MAP` used an increasing number of iterations as n increased. We note that the complexity of one iteration of `MAP` is about one full eigenvalue-eigenvector decomposition. The large number of iterations needed by `MAP` slows its convergence and takes a long time to terminate. For example, `Newton` took about 8 seconds to solve Example 5.4 ($n = 1000$) while `MAP` used about 13 minutes. When $n = 2000$, the numbers are 51 seconds for `Newton` versus nearly 2 hours for `MAP`, which reached the maximum iteration with $\|\theta(y^k)\| \approx 10^{-3}$. This lower accuracy of the final iterate is reflected by the corresponding (slightly) lower objective function value of 84996 compared to 84998 returned by the Newton method. This is because the final matrix returned by `MAP` is not yet (but close to) an EDM due to the low accuracy. `QSDP` suffers difficulties similar to `MAP` when n gets bigger than 1000. When $n = 2000$, `QSDP` took more than 9 hours to terminate. As QSDP is a general solver for quadratic semidefinite programs, we feel that it has much room to improve on our test problems by taking advantage of the problem structure. An encouraging observation is that all methods were able to return almost the same objective function value on each test problem.

One important issue that was brought up by a referee is the scaling when using the Newton method. The following test problem was suggested by a referee. First, the n points $X = [x_1, x_2, \ldots, x_n]$ are generated by $X =$ `rand(n-1,n)`. We then calculate $D_{ij} = \|x_i - x_j\|^2$. Finally, D is perturbed by small quantities via $D = D + 0.01 *$ `randn(n,n)`. For example, when $n = 800$, the Newton method, without scaling, would lose its quadratic convergence. This is because the step length was too small after a couple of iterations. The culprit is that D contains large distances.

TABLE 1
Comparison between Newton, *MAP, and QSDP on unweighted problems.*

		Newton				MAP				QSDP			
	n	Iter	cpu	Res	Obj	Iter	cpu	Res	Obj	Iter	cpu	gap	Obj
E5.2	100	5	0.1	6.58E-08	10804.01	42	0.1	8.27E-06	10804.01	12	17	2.75E-07	10804.04
	200	6	0.2	8.62E-10	50738.29	60	0.6	8.61E-06	50738.29	12	58	7.03E-07	50738.54
	500	5	1.2	9.38E-07	377674.26	85	7	9.94E-06	377674.26	12	8:05	2.73E-07	377674.84
	1000	6	7	3.61E-09	1644287.62	116	54	9.58E-06	1644287.62	12	45:38	7.29E-07	1644294.55
	1500	6	22	6.92E-09	3856026.55	141	3:20	9.83E-06	3856026.54	13	2:25:24	9.87E-08	3856028.46
	2000	6	55	1.69E-08	7023779.98	161	9:54	9.18E-06	7023779.97	13	5:14:33	8.82E-08	7023783.17
E5.3	100	4	0.1	1.48E-07	8.27	27	0.1	7.91E-06	8.27	15	18	2.94E-07	8.27
	200	5	0.2	1.37E-08	43.21	46	0.4	9.88E-06	43.21	16	50	4.34E-07	43.21
	500	5	1.3	1.59E-08	333.57	61	5	9.81E-06	333.57	23	6:18	3.93E-07	333.59
	1000	5	6	9.46E-08	1464.24	86	40	9.63E-06	1464.24	36	1:15:31	1.02E-07	1464.26
	1500	5	18	2.53E-07	3438.71	104	2:31	9.39E-06	3438.71	47	4:07:54	6.99E-07	3439.08
	2000	6	54	6.26E-09	6280.59	118	7:12	9.83E-06	6280.59	59	12:24:12	4.18E-07	6280.98
E5.4	100	6	0.1	4.65E-09	156.92	162	0.4	9.94E-06	156.92	14	15	3.41E-07	156.92
	200	6	0.3	1.36E-07	718.24	342	3	9.98E-06	718.24	16	50	6.25E-07	718.24
	500	7	1.7	3.35E-07	5119.41	861	1:05	9.96E-06	5119.41	20	6:58	8.51E-07	5119.44
	1000	7	8	5.54E-07	21367.14	1783	13:05	9.97E-06	21367.14	24	50:40	8.36E-07	21367.22
	1500	8	24	2.98E-08	47629.30	2000	48:04	2.71E-04	47629.17	29	3:25:24	1.08E-07	47629.33
	2000	8	51	1.91E-07	84998.87	2000	1:57:03	2.64E-03	84996.76	35	9:32:38	5.19E-07	84999.08

When D is scaled to $D/\max(D_{ij})$ to bring all distances between 0 and 1, the quadratic convergence returns for this scaled matrix. Of course, the obtained solution should be scaled back by multiplying $\max(D_{ij})$. For this particular problem ($n = 800$), we have `Iter` = 1, `cpu` = 0.7s, `Res` = 1.43E-5, and `Obj` = 2.12E-3. The corresponding result for `MAP` (no scaling was used) are `Iter` = 2, `cpu` = 0.8s, `Res` = 4.53E-7, and `Obj` = 2.12E-3. For other values of n, we obtained similar results (e.g., `Iter` =1 for the Newton method and 2 for `MAP`). The reason why `Newton` and `MAP` took just one or two iterations is that the starting point is very close to the true solution, bearing in mind that the true distance matrix is only perturbed by a small amount. For a general discussion on the need of scaling in optimization methods, we refer the reader to [5, sect. 1.10, Scaling].

In Tables 2 and 3, we report our numerical experience with Algorithm 4.1 and the `QSDP` solver on H-weighted problems with (e.g., Example 5.6) and without (e.g., Example 5.5) additional fixed distances (`MAP` is not applicable to these kind of problems).

TABLE 2

Comparison between Algorithm 4.1 *and* `QSDP` *on H-weighted problems. "*" means that* `psqmr` *in* `QSDP` *reached the maximum number of steps and the algorithm terminated before reaching the accuracy.*

			Algorithm 4.1				`QSDP`			
$n = 500$	Density	R	Iter	cpu	f_{prog}	Obj	Iter	cpu	gap	Obj
	99.79%	1.5	2	4	1.44E-06	1755.9480	23	8:57	1.72E-06	1756.0345
	90.80%	1	18	26	5.79E-06	1088.3068	26	7:44	8.83E-06	1088.4785
	69.14%	0.8	30	38	1.95E-06	428.3699	24	11:13	1.31E-06	427.6617
E5.5	40.68%	0.6	93	1:33	9.00E-06	85.3167	21	11:10	1.80E-06	85.1006
	27.27%	0.5	144	2:08	9.87E-06	26.9529	20	13:46	2.53E-06	26.5405
	16.16%	0.4	101	1:30	9.83E-06	6.4204	20	19:15	4.39E-06	6.0031
	2.63%	0.2	40	36	9.28E-06	0.0192	20	24:04	1.34E-02*	0.0313
	99.79%	1.5	4	1:01	3.19E-07	2666.3489	50	9:03	5.82E-06	2666.6126
	90.80%	1	15	1:50	8.81E-06	1551.5751	48	8:34	5.08E-06	1551.6498
	69.14%	0.8	27	2:25	3.74E-06	527.9511	43	11:02	2.08E-06	527.2400
E5.6	40.68%	0.6	86	5:10	9.92E-06	95.4337	35	12:57	2.10E-06	95.2293
	27.27%	0.5	141	6:54	9.94E-06	28.3609	29	14:59	1.70E-06	27.9579
	16.16%	0.4	101	2:43	9.85E-06	6.6095	29	19:52	3.64E-06	6.1862
	2.63%	0.2	39	45	9.91E-06	0.0197	21	3:41	6.76E-02*	0.0874

TABLE 3

Comparison between Algorithm 4.1 *and* `QSDP` *on H-weighted problems. "*" means that* `psqmr` *in* `QSDP` *reached the maximum number of steps and the algorithm terminated before reaching the accuracy.*

				Algorithm 4.1				`QSDP`			
	n	Density	R	Iter	cpu	f_{prog}	Obj	Iter	cpu	gap	Obj
	100	91.98%	1	11	1	4.47E-06	34.1656	14	14	2.27E-06	34.1322
	200	71.9%	0.8	24	5	3.41E-07	62.0968	16	31	8.62E-06	61.9664
E5.5	500	4.78%	0.25	51	44	9.80E-06	0.16880	21	32:42	2.63E-04*	0.13504
	1000	2.56%	0.2	56	4:11	9.59E-06	0.15303	21	3:52:29	1.05E-03*	0.12348
	1500	2.57%	0.2	68	13:34	9.82E-06	0.45769	23	13:21:57	1.20E-03*	0.38668
	2000	2.59%	0.2	80	17:55	9.77E-06	0.96547	27	36:58:21	4.76E-04*	0.81649
	100	91.98%	1	10	5	5.81E-06	50.8218	21	13	7.96E-06	50.7989
	200	71.9%	0.8	23	21	2.35E-07	84.1753	29	57	1.97E-06	84.0205
E5.6	500	4.78%	0.25	51	1:04	9.83E-06	0.17051	28	40:57	1.35E-04*	0.13508
	1000	2.56%	0.2	56	4:33	9.56E-06	0.15484	30	4:11:52	8.39E-04*	0.12346
	1500	2.57%	0.2	68	14:38	9.81E-06	0.46239	33	12:54:59	1.22E-03*	0.39156
	2000	2.59%	0.2	80	34:55	9.80E-06	0.97490	36	36:58:49	8.22E-04*	0.83625

In our implementation, we used $\tau = 0.1$ (see (57)). We terminate Algorithm 4.1 when

$$f_{\text{prog}} := \frac{|\sqrt{f(X^{k-1})} - \sqrt{f(X^k)}|}{\max\{100, \sqrt{f(X^{k-1})}\}} \leq 10^{-5}. \tag{69}$$

In other words, whenever there is a lack of relative progress on the successive objective function values, we stop the algorithm. This stopping criterion was suggested by Gao and Sun [18] for their majorization method. We once again used the default parameter values for `QSDP`. In particular, it was terminated when the relative gap defined in [45] was less than 10^{-5}.

It is observed that the H-weighted problem is much more difficult to solve than the unweighted one. The difficulty level seems to increase as the density of H decreases. In Table 2, we tested Examples 5.5 and 5.6 with fixed dimension $n = 500$, but with varying densities (ranging from 99.79% to 2.63%). In Table 3, we tested the two examples with varying dimensions (from $n = 100$ to 2000) and varying densities (from 91.98% to 2.59%). It is evident that our algorithm performed significantly faster than `QSDP` for all the problems. An important observation from those tables as well as from our extensive numerical experiments not reported here is that when the density is above about 10%, both Algorithm 4.1 and the `QSDP` solver returned almost the same objective function value. However, when it is below 10%, `QSDP` often terminated early as the `psqmr` solver used in `QSDP` reached the default maximum number of steps. This observation can be clearly seen from Table 3 for $n \geq 500$ with density less than 5%. It is also worth mentioning that one can stop Algorithm 4.1 at any iteration once a satisfactory approximate solution is obtained. This is because at each iteration of Algorithm 4.1, the solution of the subproblem solved by Newton method (66) already provides a Euclidean distance matrix.

6. Conclusion and future research. In this paper, we studied the Newton method for computing the nearest Euclidean distance matrix from a given predistance matrix or dissimilarity. Our theoretical analysis is mainly of the unweighted case (3). The main result is that the Newton method is quadratically convergent. This main result also holds for the diagonally weighted problem (62), which naturally arises from a majorization approach for the H-weighted problem (4). Our numerical experiments showed that the Newton method is extremely efficient even for large scale problems. This research also provides a solid foundation for other important problems.

One such problem is the embedding problem (6) and its H-weighted version:

$$\min \frac{1}{2}\|H \circ (X - D)\|^2 \qquad \text{s.t.} \quad X \in \mathcal{S}_h^n \cap \mathcal{K}_+^n \quad \text{and} \quad \text{rank}(JXJ) \leq r. \tag{70}$$

In distance geometry models for molecular conformation, distances are often known to be contained in a box, i.e., $l_{ij} \leq D_{ij} \leq u_{ij}$, $1 \leq i, j \leq n$. "*The difference between the upper bound and lower bound reflects the accuracy with which the data is known. To reflect this accuracy in the algorithms, it is important that weighted models be considered.*" For more explanation on the above statement, see [21, p. 114], which recommends $H_{ij} = 1/(1 + 10(u_{ij} - l_{ij}))$. The findings in our paper open a new avenue for using the Newton method to (70) through a penalty approach (i.e., penalizing the rank constraint). As correctly pointed out by one referee, the rank constraint would "break the convexity and duality gap could arise." The latest research shows that the duality gap vanishes under reasonable conditions, and the Newton method developed in this paper plays a very important role in solving (70) (see [40] for details).

For the H-weighted problem (4), we proposed a majorization approach, which at each iteration solves a diagonally weighted problem. As seen from Tables 2 and 3, this approach sometimes took a good number of iterations to reach the required accuracy. Given that the inner problem can be efficiently solved, we plan to investigate strategies for improving the efficiency of the majorization approach as well as other approaches.

Acknowledgments. The author would like to thank Prof. K.-C. Toh of the National University of Singapore for his help on using the `QSDP` solver on the test problems. The author also wishes to thank the two referees for their valuable comments and constructive suggestions, which have significantly improved the quality of the paper.

REFERENCES

[1] A. Y. Alfakih, A. Khandani, and H. Wolkowicz, *Solving Euclidean distance matrix completion problems via semidefinite programming*, Comput. Optim. Appl., 12 (1999), pp. 13–30.

[2] S. Al-Homidan and H. Wolkowicz, *Approximate and exact completion problems for Euclidean distance matrices using semidefinite programming*, Linear Algebra Appl., 406 (2005), pp. 109–141.

[3] F. Alizadeh, J.-P. A. Haeberly, M. L. Overton, *Complementarity and nondegeneracy in semidefinite programming*, Math. Program., 77 (1997), pp. 111–128.

[4] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sciences, 2 (2009), pp. 183–202.

[5] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed., Athena Scientific, Nashua, NH, 1999.

[6] I. Borg and P. J. F. Groenen, *Modern Multidimensional Scaling: Theory and Applications*, 2nd ed., Springer Ser. Statist., Springer, New York, 2005.

[7] R. Borsdorf and N. J. Higham, *A preconditioned Newton algorithm for the nearest correlation matrix*, IMA J. Numer. Anal., 94 (2010), pp. 94–107.

[8] S. Boyd and L. Xiao, *Least-squares covariance matrix adjustment*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 532–546.

[9] X. Chen, H. Qi, and P. Tseng, *Analysis of nonsmooth symmetric-matrix-valued functions with applications to semidefinite complementarity problems*, SIAM J. Optim., 13 (2003), pp. 960–985.

[10] Z. X. Chan and D. Sun, *Constraint nondegeneracy, strong regularity, and nonsingularity in semidefinite programming*, SIAM J. Optim., 19 (2008), pp. 370–396.

[11] F. H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley & Sons, New York, 1983.

[12] T. F. Cox and M. A. A. Cox, *Multidimensional Scaling*, 2nd ed., Chapman and Hall/CRC, Boca Raton, FL, 2001.

[13] G. Crippen and T. Havel, *Distance Geometry and Molecular Conformation*, John Wiley & Sons, New York, 1988

[14] J. Dattorro, *Convex Optimization and Euclidean Distance Geometry*, Meboo Publishing USA, Palo Alto, CA, 2005.

[15] R. L. Dykstra, *An algorithm for restricted least squares regression*, J. Amer. Statist. Assoc., 78 (1983), pp. 839–842.

[16] H.-R. Fang and D. P. O'Leary, *Euclidean distance matrix completion problems*, Optim. Methods Softw., 27 (2012), pp. 695–717.

[17] N. Gaffke and R. Mathar, *A cyclic projection algorithm via duality*, Metrika, 36 (1989), pp. 29–54.

[18] Y. Gao and D. F. Sun, *A Majorized Penalty Approach for Calibrating Rank Constrained Correlation Matrix Problems*, Technical Report, Department of Mathematics, National University of Singapore, Singapore, 2010.

[19] 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 (1990), pp. 589–600.

[20] W. Glunt, T. L. Hayden, and W.-M. Liu, *The embedding problem for predistance matrices*, Bull. Math. Biol., 53 (1991), pp. 769–796.

[21] W. Glunt, T. L. Hayden, and R. Raydan, *Molecular conformations from distance matrices*, J. Comput. Chem., 14 (1993), pp. 114–120.

[22] W. Glunt, T. L. Hayden, and R. Raydan, *Preconditioners for distance matrix algorithms*, J. Comput. Chem., 15 (1994), pp. 227–232.
[23] J. C. Gower, *Properties of Euclidean and non-Euclidean distance matrices*, Linear Algebra Appl., 67 (1985), pp. 81–97.
[24] S. P. Han, *A successive projection method*, Math. Programming, 40 (1988), pp. 1–14.
[25] T. L. Hayden and J. Wells, *Approximation by matrices positive semidefinite on a subspace*, Linear Algebra Appl., 109 (1988), pp. 115–130.
[26] T. L. Hayden, J. Wells, W.-M. Liu, and P. Tarazaga, *The cone of distance matrices*, Linear Algebra Appl., 144 (1991), pp. 153–169.
[27] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Stand., 49 (1952), pp. 409–436.
[28] N. J. Higham, *Computing the nearest correlation matrix—a problem from finance*, IMA J. Numer. Anal., 22 (2002), pp. 329–343.
[29] K. Jiang, D. Sun, and K.-C. Toh, *An inexact accelerated proximal gradient method for large scale linearly constrained convex SDP*, SIAM J. Optim., 22 (2012), pp. 1042–1064.
[30] N. Krislock and H. Wolkowicz, *Euclidean distance matrices and applications*, in Handbook on Semidefinite, Conic and Polynomial Optimization, M. Anjos and J. Lasserre, eds., Springer, New York, 2012, pp. 879–914.
[31] J. Malick, *A dual approach to semidefinite least-squares problems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 272–284.
[32] J. Malick and H. S. Sendov, *Clarke generalized Jacobian of the projection onto the cone of positive semidefinite matrices*, Set-Valued Anal., 14 (2006), pp. 273–293.
[33] C. A. Micchelli, P. W. Smith, J. Swetits, and J. D. Ward, *Constrained ℓ_p approximation*, J. Construct. Approx., 9 (1985), pp. 93–102.
[34] C. A. Micchelli and F. I. Utreras, *Smoothing and interpolation in a convex subset of a Hilbert space*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 728–746.
[35] Y. Nesterov, *A method of solving a convex programming problem with convergence rate $O(1/k^2)$*, Soviet Math. Dokl., 27 (1983), pp. 372–376.
[36] A. Neumaier, *Molecular modeling of proteins and mathematical prediction of protein structure*, SIAM Rev., 39 (1997), pp. 407–460.
[37] H.-D. Qi, *Positive semidefinite matrix completions on chordal graphs and constraint nondegeneracy in semidefinite programming*, Linear Algebra Appl., 430 (2009), pp. 1151–1164.
[38] H. Qi and D. Sun, *A quadratically convergent Newton method for computing the nearest correlation matrix*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 360–385.
[39] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Programming, 58 (1993), pp. 353–367.
[40] H.-D. Qi and X. M. Yuan, *Computing the Nearest Euclidean Distance Matrix with Low Embedding Dimensions*, Tech. Report, School of Mathematics, University of Southampton, Southampton, UK, 2012.
[41] R. T. Rockafellar, *Conjugate Duality and Optimization*, CBMS-NSF Reg. Conf. Ser. Appl. Math. 16, SIAM, Philadelphia, 1974.
[42] I. J. Schoenberg, *Remarks to Maurice Fréchet's article "Sur la définition axiomatique d'une classe d'espace vectoriels distanciés vectoriellement applicables sur l'espace de Hilbert,"* Ann. Math., 36 (1935), pp. 724–732.
[43] D. F. Sun, *The strong second-order sufficient condition and constraint nondegeneracy in nonlinear semidefinite programming and their implications*, Math. Oper. Res., 31 (2006), pp. 761–776.
[44] D. F. Sun and J. Sun, *Semismooth matrix valued functions*, Math. Oper. Res., 27 (2002), pp. 150–169.
[45] K. C. Toh, *An inexact path-following algorithm for convex quadratic SDP*, Math. Programming, 112 (2008), pp. 221–254.
[46] P. Tseng, *On Accelerated Proximal Gradient Methods for Convex-Concave Optimization*, Tech. report, University of Washington, Seattle, WA, 2008.
[47] G. Young and A. S. Householder, *Discussion of a set of points in terms of their mutual distances*, Psychometrika, 3 (1938), pp. 19–22.