Explicit Sensor Network Localization using Semidefinite Representations and Clique Reductions *

Nathan Krislock Henry Wolkowicz

May 17, 2009

University of Waterloo
Department of Combinatorics and Optimization
Waterloo, Ontario N2L 3G1, Canada
Research Report CORR 2009-04

Key Words: Sensor Network Localization, Euclidean Distance Matrix Completions, Semidefinite Programming, loss of the Slater constraint qualification.
AMS Subject Classification:

Abstract

The sensor network localization, ***SNL***, problem in embedding dimension r, consists of locating the positions of wireless sensors, given only the distances between sensors that are within radio range and the positions of a subset of the sensors (called anchors). Current solution techniques relax this problem to a weighted, nearest, (positive) semidefinite programming, ***SDP***, completion problem, by using the linear mapping between Euclidean distance matrices, ***EDM***, and semidefinite matrices. The resulting ***SDP*** is solved using primal-dual interior point solvers, yielding an expensive and inexact solution.

This relaxation is highly degenerate in the sense that the feasble set is restricted to a low dimensional face of the ***SDP*** cone, implying that the Slater constraint qualification fails. The degeneracy in the ***SDP*** arises from cliques in the graph of the ***SNL*** problem. In this paper, we take advantage of the absence of the Slater constraint qualification and derive a technique for the ***SNL*** problem, with exact data, that explicitly solves the corresponding rank restricted ***SDP*** problem. No ***SDP*** solvers are used. We are able to efficiently solve this NP-HARD problem with high probability, by finding a representation of the minimal face of the ***SDP*** cone that contains the ***SDP*** matrix representation of the ***EDM***. The main work of our algorithm consists in repeatedly finding the intersection of subspaces that represent the faces of the ***SDP*** cone that correspond to cliques of the ***SNL*** problem.

Contents

*Research supported by Natural Sciences Engineering Research Council Canada and a grant from AFOSR.

1 Introduction

The sensor network localization problem, $\boldsymbol{SNL}$, consists in locating the positions of n ad hoc wireless sensors, $p_i \in \mathbb{R}^r$, $i = 1, \ldots, n$, given only the (squared) Euclidean distances $D_{ij} = \|p_i - p_j\|_2^2$ between sensors that are within a given radio range, $R > 0$, and given the positions of a subset of the sensors, p_i, $i = n-m+1, \ldots, n$ (called anchors); r is the *embedding dimension* of the problem. Currently, many solution techniques for this problem use a relaxation to a nearest, weighted, semidefinite approximation problem

$$\min_{Y \succeq 0,\, Y \in \Omega} \|W \circ (\mathcal{K}(Y) - D)\|\,, \tag{1.1}$$

where $Y \succeq 0$ denotes positive semidefiniteness, $Y \in \Omega$ denotes additional linear constraints, $\mathcal{K}$ is a specific linear mapping, and $\circ$ denotes the *Hadamard (elementwise) product*. This approach requires semidefinite programming, $\boldsymbol{SDP}$, primal-dual interior point (p-d i-p) techniques; see, for example, [2, 3, 5, 9, 10, 13, 22]. This yields an expensive and inexact solution.

The $\boldsymbol{SNL}$ problem is a special case of the Euclidean Distance Matrix, $\boldsymbol{EDM}$, completion problem, $\boldsymbol{EDMC}$. If D is a *partial* $\boldsymbol{EDM}$, then the completion problem consists in finding the missing elements (squared distances) of D. It is shown in [14], that there are advantages for handling the $\boldsymbol{SNL}$ problem as an $\boldsymbol{EDMC}$, and ignoring the distinction between the anchors and the other sensors until after the $\boldsymbol{EDMC}$ is solved. In this paper we use this framework and derive an algorithm that locates the sensors by exploiting the structure and implicit degeneracy in the $\boldsymbol{SNL}$ problem.

In particular, we solve the ***SDP*** problems *explicitly* (exactly) without using programming (p-d i-p solvers) techniques. We do so by repeatedly viewing ***SNL*** in three equivalent forms: as a graph realization problem, as a ***EDMC***, and as a rank restricted ***SDP***.

A common approach to solving the ***EDMC*** is to relax the rank constraint and solve a weighted, nearest, positive semidefinite completion problem using semidefinite programming, ***SDP***, e.g. (1.1). The resulting ***SDP*** is, implicitly, highly degenerate in the sense that the feasible semidefinite matrices have low rank. The low rank arises from cliques in the graph of the ***SNL***. This means that the Slater constraint qualification (strict feasibility) implicitly fails for the ***SDP***. Our algorithm is based on exploiting this degeneracy. We characterize the face of the ***SDP*** cone that corresponds to a given clique in the graph, thus reducing the size of the ***SDP*** problem. Then, we characterize the intersection of two faces that correspond to overlapping cliques. This allows us to explicitly *grow/increase* the size of the cliques by repeatedly finding the intersection of subspaces that represent the faces of the ***SDP*** cone that correspond to these cliques. Equivalently, this corresponds to completing overlapping blocks of the ***EDM***. In this way, we further reduce the dimension of the faces until we get a completion of the entire ***EDM***. The intersection of the subspaces can be found using a singular value decomposition (SVD) or by exploiting the special structure of the subspaces. No ***SDP*** solver is used. Thus we solve the ***SDP*** problem in a finite number of steps, where the work of each step is to find the intersection of two subspaces. (Equivalently the intersection of two faces of the ***SDP*** cone.)

Though our results hold for general embedding dimension r, our preliminary numerical tests involve sensors with embedding dimension $r = 2$, i.e. in the plane. The sensors are in a square with sides of length b units. There are n sensors, m of which are anchors. The radio range is R units.

1.1 Related Work/Applications

The number of applications for distance geometry problems is large and increasing in number and importance. The particular case of ***SNL*** has applications to environmental monitoring of geographical regions, as well as tracking of animals and machinery; see, for example, [5, 13]. There have been many algorithms published recently that solve the ***SNL*** problem. Many of these involve ***SDP*** relaxations and use ***SDP*** solvers; see, for example, [5, 6, 7, 8, 9, 10, 14] and more recently [26]. Heuristics are presented in, for example, [12]. ***SNL*** is closely related to the ***EDMC*** problem; see, for example, [3, 13] and the survey in [2].

Jin et al [11, 21] propose the *SPASELOC* heuristic. It is limited to $r = 2$ and uses an ***SDP*** solver for small localized subproblems. They then *sew* these subproblems together. So & Ye [24] show that the problem of solving a noiseless ***SNL*** with a unique solution can be phrased as an ***SDP*** and thus can be solved in polynomial time. They also give an efficient criterion for checking whether a given instance has a unique solution for $r = 2$.

Two contributions of this paper are: we do not use iterative programming techniques to solve our SDPs, but rather, we solve them with a finite number of explicit solutions; we start with local cliques and expand the cliques. Our algorithm has four different basic steps. The first basic step takes two cliques for which the intersection contains at least $r + 1$ nodes and implicitly completes the corresponding ***EDM*** to form the union of the cliques. The second step does this when one of the cliques is a single element. Therefore, this provides an extension of the algorithm in [16], where Eren et al have shown that the family of *trilateration graphs* admit a polynomial time algorithm

for computing a realization in a required dimension.[1] Our first basic step also provides an explicit form for finding a realization of a *uniquely localizable graph*[2] Our algorithm repeatedly finds explicit solutions of an $\boldsymbol{SDP}$. Other examples of finding explicit solutions of an $\boldsymbol{SDP}$ are given in [25, 27].

The $\boldsymbol{SNL}$ problem with given embedding dimension r is NP-HARD [19, 20, 23]. However, from our numerical tests it appears that random problems that have a unique solution can be solved in polynomial time. This phenomenon fits into the results in [4, 17].

1.2 Outline

We continue in Section 1.3 to present notation and preliminary results. The clique reduction process is based on the results in Section 2. The single clique reduction is given in Theorem 2.3; the reduction of two overlapping cliques in the rigid and nonrigid cases is presented in Theorem 2.10 and Theorem 2.14, respectively; absorbing nodes into cliques in the rigid and nonrigid cases is given in Corollaries 2.17 and 2.18, respectively. These results are then used in our algorithm in Section 3. The numerical tests appear in Section 3.1. Our concluding remarks are given in Section 4.

1.3 Preliminaries

We work in the vector space of *real symmetric* $k \times k$ *matrices*, $\mathcal{S}^k$, equipped with the *trace inner product*, $\langle A, B \rangle = \operatorname{trace} AB$. We let $\mathcal{S}^k_+$ and $\mathcal{S}^k_{++}$ denote the cone of positive semidefinite and positive definite matrices, respectively; $A \succeq B$ and $A \succ B$ denote the Löwner partial order, $A - B \in \mathcal{S}^k_+$ and $A - B \in \mathcal{S}^k_{++}$, respectively; e denote the vector of ones of appropriate dimension; $\mathcal{R}(\mathcal{L})$ and $\mathcal{N}(\mathcal{L})$ denote the range space and null space of the linear transformation $\mathcal{L}$, respectively; $\operatorname{cone}(S)$ denote the convex cone generated by the set S. We use the MATLAB notation $1:n = \{1, \ldots, n\}$.

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

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

If $F \trianglelefteq K$, but is not equal to K, we write $F \triangleleft K$. If $\{0\} \neq F \triangleleft K$, then F is a *proper face* of K. For $S \subseteq K$, we let face(S) denote the smallest face of K that contains S. A face $F \trianglelefteq K$ is an *exposed face* if it is the intersection of K with a hyperplane. The cone K is *facially exposed* if every face $F \trianglelefteq K$ is exposed.

The cone $\mathcal{S}^n_+$ is facially exposed. Moreover, each face $F \trianglelefteq \mathcal{S}^n_+$ is determined by the range of any matrix S in the relative interior of the face, $S \in \operatorname{relint} F$: if $S = U \Gamma U^T$ is the compact spectral decomposition of S with the diagonal matrix of eigenvalues $\Gamma \in \mathcal{S}^t_{++}$, then

$$F = U \mathcal{S}^t_+ U^T. \tag{1.2}$$

A matrix $D = (D_{ij}) \in \mathcal{S}^n$ with nonnegative elements and zero diagonal is called a *pre-distance matrix*. In addition, if there exist points $p_1, \ldots, p_n \in \mathbb{R}^r$ such that

$$D_{ij} = \|p_i - p_j\|_2^2, \quad i, j = 1, \ldots, n, \tag{1.3}$$

[1] A graph is a trilateration graph in dimension r if there exists an ordering of the nodes $1, \ldots, r+1, r+2, \ldots, n$ such that: the first $r+1$ nodes form a clique, and each node $j > r+1$ has at least $r+1$ edges to nodes earlier in the sequence.

[2] A graph is uniquely localizable in dimension r if it has a unique realization in $\mathbb{R}^r$ and it does not have any realization whose affine span is $\mathbb{R}^h$, where $h > r$; see [24].

then D is called a *Euclidean distance matrix*, denoted $\boldsymbol{EDM}$. Note that we work with *squared distances*. The smallest value of r such that (1.3) holds is called the *embedding dimension* of D. Throughout the paper, we assume that r is given and *fixed*. The set of $\boldsymbol{EDM}$ matrices forms a closed convex cone in $\mathcal{S}^n$, denoted $\mathcal{E}^n$. If we are given a partial $\boldsymbol{EDM}$, $D_p \in \mathcal{E}^n$, let $\mathcal{G} = (V, E, \omega)$ be the corresponding simple graph on the vertices $V = 1\!:\!n$ whose edges E correspond to the known entries of D_p, with $(D_p)_{ij} = \omega_{ij}^2$, for all $(i, j) \in E$.

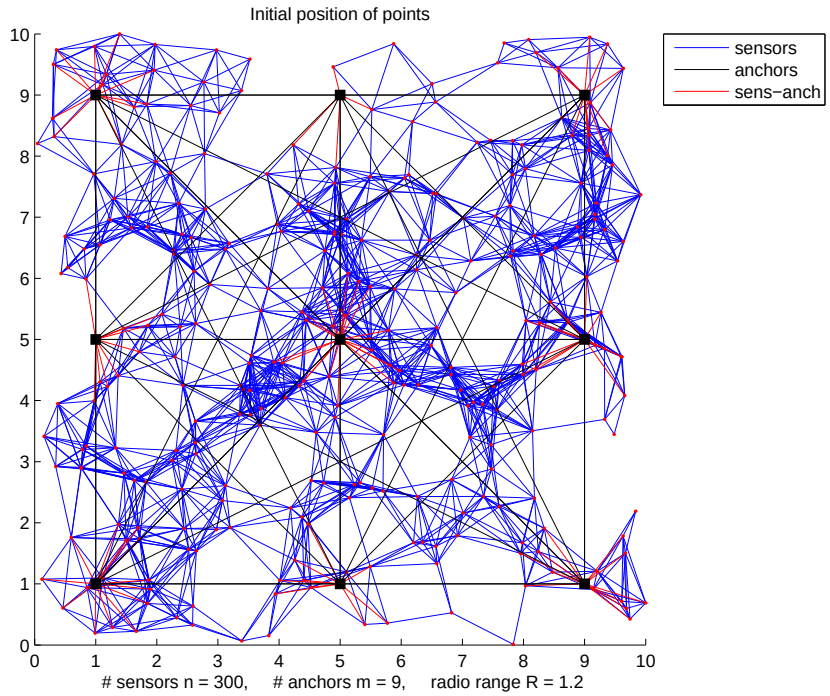

Figure 1.1: Graph of partial $\boldsymbol{EDM}$ with sensors ○ and anchors ■

Definition 1.1. *For $Y \in \mathcal{S}^n$ and $\alpha \subseteq 1:n$, we let $Y[\alpha]$ denote the corresponding* principal submatrix *formed from the rows and columns with indices α. If, in addition, $|\alpha| = k$ and $\bar{Y} \in \mathcal{S}^k$ is given, then we define*

$$\mathcal{S}^n(\alpha, \bar{Y}) := \left\{ Y \in \mathcal{S}^n : Y[\alpha] = \bar{Y} \right\}, \quad \mathcal{S}^n_+(\alpha, \bar{Y}) := \left\{ Y \in \mathcal{S}^n_+ : Y[\alpha] = \bar{Y} \right\},$$

i.e. the subset of matrices $Y \in \mathcal{S}^n$ ($Y \in \mathcal{S}^n_+$) with principal submatrix $Y[\alpha]$ fixed to $\bar{Y}$.

For example, the subset of matrices in $\mathcal{S}^n$ with the top left $k \times k$ block fixed is

$$\mathcal{S}^n(1\!:\!k, \bar{Y}) = \left\{ Y \in \mathcal{S}^n : Y = \left[\begin{array}{c|c} \bar{Y} & \cdot \\ \hline \cdot & \cdot \end{array}\right] \right\}. \tag{1.4}$$

A clique $\gamma \subseteq 1:n$ in the graph $\mathcal{G}$ corresponds to a subset of sensors for which the distances $\omega_{ij} = \|p_i - p_j\|_2$ are known, for all $i, j \in \gamma$; equivalently, the clique corresponds to the principal submatrix $D_p[\gamma]$ of the partial $\boldsymbol{EDM}$ matrix D_p, where all the elements of $D_p[\gamma]$ are known.

Suppose that we are given a subset of the (squared) distances from (1.3) in the form of a partial $\boldsymbol{EDM}$, D_p. The $\boldsymbol{EDM}$ *completion problem* consists of finding the missing entries of D_p

to complete the ***EDM***; see Figure 1.1. This completion problem can be solved by finding a set of points $p_1, \ldots, p_n \in \mathbb{R}^r$ satisfying (1.3), where r is the embedding dimension of the partial ***EDM***, D_p. This problem corresponds to the graph realizability problem with dimension r, which is the problem of finding positions in $\mathbb{R}^r$ for the vertices of a graph such that the inter-distances of these positions satisfy the given edge lengths of the graph.

Let $Y \in \mathcal{M}^n$ be an $n \times n$ real matrix and $y \in \mathbb{R}^n$ a vector. We let $\mathrm{diag}(Y)$ denote the vector in $\mathbb{R}^n$ formed from the diagonal of Y and we let $\mathrm{Diag}(y)$ denote the diagonal matrix in $\mathcal{M}^n$ with the vector y along its diagonal. Note that diag and Diag are the adjoint linear transformations of each other: $\mathrm{Diag} = \mathrm{diag}^*$. The operator offDiag can then be defined as $\mathrm{offDiag}(Y) := Y - \mathrm{Diag}(\mathrm{diag}\, Y)$. For

$$P = \begin{bmatrix} p_1^T \\ p_2^T \\ \vdots \\ p_n^T \end{bmatrix} \in \mathcal{M}^{n \times r},$$

where p_j, $j = 1, \ldots, n$, are the points used in (1.3), let $Y := PP^T$, and let D be the corresponding ***EDM*** satisfying (1.3). Defining the linear operators $\mathcal{K}$ and $\mathcal{D}_e$ on $\mathcal{S}^n$ as follows, we see that

$$\begin{array}{rcl} \mathcal{K}(Y) & := & \mathcal{D}_e(Y) - 2Y \\ & := & \mathrm{diag}(Y)\, e^T + e\, \mathrm{diag}(Y)^T - 2Y \\ & = & \left(p_i^T p_i + p_j^T p_j - 2 p_i^T p_j \right)_{i,j=1}^n \\ & = & \left(\|p_i - p_j\|_2^2 \right)_{i,j=1}^n \\ & = & D. \end{array} \tag{1.5}$$

That is, $\mathcal{K}$ maps the positive semidefinite matrix Y onto the ***EDM*** D. More generally, we can allow for a general vector v to replace e, and define $\mathcal{D}_v(Y) := \mathrm{diag}(Y)\, v^T + v\, \mathrm{diag}(Y)^T$. By abuse of notation, we also allow $\mathcal{D}_v$ to act on a vector; that is, $\mathcal{D}_v(y) := yv^T + vy^T$. The adjoint of $\mathcal{K}$ is

$$\mathcal{K}^*(D) \;=\; 2(\mathrm{Diag}(De) - D). \tag{1.6}$$

The linear operator $\mathcal{K}$ is one-one and onto between the *centered* and *hollow* subspaces of $\mathcal{S}^n$, which are defined as

$$\begin{array}{rcll} \mathcal{S}_C & := & \{Y \in \mathcal{S}^n : Ye = 0\} & \text{(zero row sums)}, \\ \mathcal{S}_H & := & \{D \in \mathcal{S}^n : \mathrm{diag}(D) = 0\} & = \mathcal{R}(\mathrm{offDiag}). \end{array} \tag{1.7}$$

Let $J := I - \frac{1}{n} ee^T$ denote the orthogonal projection onto the subspace $\{e\}^\perp$ and define the linear operator $\mathcal{T}(D) := -\frac{1}{2} J\, \mathrm{offDiag}(D) J$. Then we have the following relationships.

Proposition 1.2. *([1]) The linear operator $\mathcal{T}$ is the generalized inverse of the linear operator $\mathcal{K}$; that is, $\mathcal{K}^\dagger = \mathcal{T}$. Moreover:*

$$\begin{array}{rlrl} \mathcal{R}(\mathcal{K}) = \mathcal{S}_H; & & \mathcal{N}(\mathcal{K}) = \mathcal{R}(\mathcal{D}_e); \\ \mathcal{R}(\mathcal{K}^*) = \mathcal{R}(\mathcal{T}) = \mathcal{S}_C; & & \mathcal{N}(\mathcal{K}^*) = \mathcal{N}(\mathcal{T}) = \mathcal{R}(\mathrm{Diag}); \end{array} \tag{1.8}$$

$$\mathcal{S}^n = \mathcal{S}_H \oplus \mathcal{R}(\mathrm{Diag}) = \mathcal{S}_C \oplus \mathcal{R}(\mathcal{D}_e). \tag{1.9}$$

Theorem 1.3. *([1]) The linear operators $\mathcal{T}$ and $\mathcal{K}$ are one-to-one and onto mappings between the cone $\mathcal{E}^n \subset \mathcal{S}_H$ and the face of the semidefinite cone $\mathcal{S}^n_+ \cap \mathcal{S}_C$. That is,*

$$\mathcal{T}(\mathcal{E}^n) = \mathcal{S}^n_+ \cap \mathcal{S}_C \quad \text{and} \quad \mathcal{K}(\mathcal{S}^n_+ \cap \mathcal{S}_C) = \mathcal{E}^n.$$

Remark 1.4. *$D \in \mathcal{E}^n$ has embedding dimension r if and only if $\mathcal{K}^\dagger(D) \succeq 0$ and $\operatorname{rank}(\mathcal{K}^\dagger(D)) = r$. In addition, we get $\mathcal{K}^\dagger(D)e = 0$. Therefore, we can factor $\mathcal{K}^\dagger(D) = PP^T$, for some $P \in \mathcal{M}^{n\times r}$, to recover the (centered) sensors in $\mathbb{R}^r$ from the rows in P. Note that rotations of the points in the rows of P do not change the value $Y = PP^T$, i.e. $PP^T = PQ^TQP$, if Q is orthogonal. However, the nullspace of $\mathcal{K}$ is related to translations of the points in P. Let $D \in \mathcal{E}^n$ with embedding dimension r and let $Y := \mathcal{K}^\dagger(D)$ have full rank factorization $Y = PP^T$, with $P \in \mathcal{M}^{n\times r}$. Then the translation of points in the rows of P to $\bar{P} := P + ew^T$, for some $w \in \mathbb{R}^r$, results in $\bar{Y} := \bar{P}\bar{P}^T = Y + \mathcal{D}_e(y)$, with $y := Pw + \frac{w^Tw}{2}e$, and $\mathcal{K}(\bar{Y}) = \mathcal{K}(Y) = D$, since $\mathcal{D}_e(y) \in \mathcal{N}(\mathcal{K})$. Note that $\mathcal{R}(Y) = \mathcal{R}(P)$, therefore $y = Pw + \frac{w^Tw}{2}e \in \mathcal{R}(Y) + \operatorname{cone}\{e\}$, as we will also see in more generality in Lemma 2.1 below.*

Let $D_p \in \mathcal{S}^n$ be a *partial* ***EDM*** with embedding dimension r and let $W \in \mathcal{S}^n$ be the 0–1 matrix corresponding to the known entries of D_p. One can use the substitution $D = \mathcal{K}(Y)$, where $Y \in \mathcal{S}^n_+ \cap \mathcal{S}_C$, in the ***EDM*** completion problem

$$\begin{array}{ll} \text{Find} & D \in \mathcal{E}^n \\ \text{s.t.} & W \circ D = D_p \end{array}$$

to obtain the ***SDP*** relaxation

$$\begin{array}{ll} \text{Find} & Y \in \mathcal{S}^n_+ \cap \mathcal{S}_C \\ \text{s.t.} & W \circ \mathcal{K}(Y) = D_p \end{array}.$$

This relaxation does not restrict the rank of Y and may yield a solution with embedding dimension that is too large, if $\operatorname{rank}(Y) > r$. Moreover, solving ***SDP*** problems with rank restrictions is NP-HARD. However, we work on faces of $\mathcal{S}^n_+$ described by $U\mathcal{S}^t_+U^T$, with $t \le n$. In order to find the face with the smallest dimension t, we must have the correct knowledge of the matrix U. In this paper, we obtain information on U using the cliques in the graph of the partial ***EDM***.

2 Clique Reduction

We now present several techniques for reducing an ***EDM*** completion problem when one or more (possibly intersecting) cliques are known. This extends the reduction using disjoint cliques presented in [14, 15]. In each case, we take advantage of the loss of Slater's constraint qualification and project the problem to a lower dimensional ***SDP*** cone.

We first need the following two technical lemmas that exploit the structure of the ***SDP*** cone.

Lemma 2.1. *Let $B \in \mathcal{S}^n$, $Bv = 0$, $v \neq 0$, $y \in \mathbb{R}^n$ and $\bar{Y} := B + \mathcal{D}_v(y)$. If $\bar{Y} \succeq 0$, then*

$$y \in \mathcal{R}(B) + \operatorname{cone}\{v\}.$$

Proof. First we will show that $y \in \mathcal{R}(B) + \operatorname{span}\{v\} = \mathcal{R}\left(\begin{bmatrix} B & v \end{bmatrix}\right)$. If this is not the case, then y can be written as the orthogonal decomposition

$$y = Bu + \beta v + \bar{y},$$

where $0 \neq \bar{y} \in \mathcal{R}\left(\begin{bmatrix} B & v \end{bmatrix}\right)^{\perp} = \mathcal{N}\left(\begin{bmatrix} B & v \end{bmatrix}^T\right)$. Note that $\bar{y}$ satisfies $B\bar{y} = 0$ and $v^T\bar{y} = 0$. To get a contradiction with the assumption that $\bar{Y} \succeq 0$, we let

$$z := \frac{1}{2}\frac{v}{\|v\|^2} - (1 + |\beta|)\frac{\bar{y}}{\|\bar{y}\|^2},$$

and observe that $Bz = 0$ and $v^T z = 1/2$. Then,

$$\begin{array}{rcl} z^T\bar{Y}z & = & z^T\mathcal{D}_v(y)z \\ & = & z^T\left(yv^T + vy^T\right)z \\ & = & y^Tz \\ & = & \frac{1}{2}\beta + \bar{y}^Tz \\ & < & \frac{1}{2}(1 + |\beta|) + \bar{y}^Tz \\ & = & -\frac{1}{2}(1 + |\beta|) \\ & < & 0, \end{array}$$

which gives us the desired contradiction. Therefore, $y \in \mathcal{R}(B) + \operatorname{span}\{v\}$, so to show that $y \in \mathcal{R}(B) + \operatorname{cone}\{v\}$, we only need to show that if $y = Bu + \beta v$, then $\beta \geq 0$. First note that $v^Ty = \beta v^Tv$. Then,

$$\begin{array}{rcl} v^T\bar{Y}v & = & v^T\left(yv^T + vy^T\right)v \\ & = & 2v^Tyv^Tv \\ & = & 2\beta(v^Tv)^2. \end{array}$$

Since $\bar{Y} \succeq 0$, we have $2\beta(v^Tv)^2 \geq 0$. This implies that $\beta \geq 0$, since $v \neq 0$. □

If $\bar{Y} \in \mathcal{S}_+^k$, then we can use the minimal face of $\mathcal{S}_+^k$ containing $\bar{Y}$ to find an expression for the minimal face of $\mathcal{S}_+^n$ that contains $\mathcal{S}_+^n(1\!:\!k, \bar{Y})$.

Lemma 2.2. *Let $\bar{U} \in \mathcal{M}^{k\times t}$ with $\bar{U}^T\bar{U} = I_t$. If $\operatorname{face}\{\bar{Y}\} \trianglelefteq \bar{U}\mathcal{S}_+^t\bar{U}^T$, then*

$$\operatorname{face}\mathcal{S}_+^n(1\!:\!k, \bar{Y}) \trianglelefteq \begin{bmatrix} \bar{U} & 0 \\ 0 & I_{n-k} \end{bmatrix} \mathcal{S}_+^{n-k+t} \begin{bmatrix} \bar{U} & 0 \\ 0 & I_{n-k} \end{bmatrix}^T. \tag{2.1}$$

Furthermore, if $\operatorname{face}\{\bar{Y}\} = \bar{U}\mathcal{S}_+^t\bar{U}^T$, then

$$\operatorname{face}\mathcal{S}_+^n(1\!:\!k, \bar{Y}) = \begin{bmatrix} \bar{U} & 0 \\ 0 & I_{n-k} \end{bmatrix} \mathcal{S}_+^{n-k+t} \begin{bmatrix} \bar{U} & 0 \\ 0 & I_{n-k} \end{bmatrix}^T. \tag{2.2}$$

Proof. Since $\bar{Y} \in \bar{U}\mathcal{S}_+^t\bar{U}^T$, then $\bar{Y} = \bar{U}S\bar{U}^T$, for some $S \in \mathcal{S}_+^t$. Let $Y \in \mathcal{S}_+^n(1\!:\!k, \bar{Y})$ and choose $\bar{V}$ so that $\begin{bmatrix} \bar{U} & \bar{V} \end{bmatrix}$ is an orthogonal matrix. Then, with Y blocked appropriately, we evaluate the congruence

$$0 \preceq \begin{bmatrix} \bar{V} & 0 \\ 0 & I_{n-k} \end{bmatrix}^T Y \begin{bmatrix} \bar{V} & 0 \\ 0 & I_{n-k} \end{bmatrix} = \begin{bmatrix} 0 & \bar{V}^TY_{21}^T \\ Y_{21}\bar{V} & Y_{22} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & Y_{22} \end{bmatrix}.$$

Therefore, $Y \succeq 0$ implies that $\bar{V}^TY_{21}^T = 0$. Since $\mathcal{N}(\bar{V}^T) = \mathcal{R}(\bar{U})$, we get $Y_{21}^T = \bar{U}X$, for some X. Therefore, we can write

$$Y = \begin{bmatrix} \bar{U} & 0 \\ 0 & I_{n-k} \end{bmatrix} \begin{bmatrix} S & X \\ X^T & Y_{22} \end{bmatrix} \begin{bmatrix} \bar{U} & 0 \\ 0 & I_{n-k} \end{bmatrix}^T.$$

This implies that face $\mathcal{S}_+^n(1\!:\!k,\bar{Y}) \trianglelefteq U\mathcal{S}_+^{n-k+t}U^T$, where

$$U := \begin{bmatrix} \bar{U} & 0 \\ 0 & I_{n-k} \end{bmatrix}.$$

This proves (2.1). To prove (2.2), note that if face $\{\bar{Y}\} = \bar{U}\mathcal{S}_+^t\,\bar{U}^T$ then $\bar{Y} \in \text{relint}\left(\bar{U}\mathcal{S}_+^t\,\bar{U}^T\right)$, so $\bar{Y} = \bar{U}S\bar{U}^T$, for some $S \in \mathcal{S}_{++}^t$. Letting

$$\hat{Y} := \begin{bmatrix} \bar{U} & 0 \\ 0 & I_{n-k} \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & I_{n-k} \end{bmatrix} \begin{bmatrix} \bar{U} & 0 \\ 0 & I_{n-k} \end{bmatrix}^T,$$

we see that $\hat{Y} \in \mathcal{S}_+^n(1\!:\!k,\bar{Y}) \cap \text{relint}\left(U\mathcal{S}_+^{n-k+t}U^T\right)$. This implies that there is no smaller face of $\mathcal{S}_+^n$ containing $\mathcal{S}_+^n(1\!:\!k,\bar{Y})$, completing the proof. □

2.1 Single Clique Reductions

If the principal submatrix $\bar{D} \in \mathcal{E}^k$ is given, for index set $\alpha \subseteq 1\!:\!n$, with $|\alpha| = k$, we define

$$\mathcal{E}^n(\alpha,\bar{D}) := \left\{D \in \mathcal{E}^n : D[\alpha] = \bar{D}\right\}. \tag{2.3}$$

Similarly, the subset of matrices in $\mathcal{E}^n$ with the top left $k \times k$ block fixed is

$$\mathcal{E}^n(1\!:\!k,\bar{D}) = \left\{D \in \mathcal{E}^n : D = \left[\begin{array}{c|c} \bar{D} & \cdot \\ \hline \cdot & \cdot \end{array}\right]\right\}. \tag{2.4}$$

A fixed principal submatrix $\bar{D}$ in a partial ***EDM*** D corresponds to a clique α in the graph $\mathcal{G} = (V,E,\omega)$ of the partial ***EDM*** D. Given such a fixed clique defined by the submatrix $\bar{D}$, the following theorem shows that the following set, containing the feasible set of the corresponding ***SDP*** relaxation,

$$\left\{Y \in \mathcal{S}_+^n \cap \mathcal{S}_C : \mathcal{K}(Y[\alpha]) = \bar{D}\right\} = \mathcal{K}^\dagger\left(\mathcal{E}^n(\alpha,\bar{D})\right),$$

is contained in a proper face of $\mathcal{S}_+^n$. This means that the Slater constraint qualification (strict feasibility) fails, and we can reduce the size of the ***SDP*** problem; see [14]. We expand on this and find an explicit expression for face $\mathcal{K}^\dagger\left(\mathcal{E}^n(\alpha,\bar{D})\right)$ in the following Theorem 2.3. For simplicity, here and below, we often work with ordered sets of integers for the two cliques. This simplification can always be obtained by a permutation of the indices of the sensors.

Theorem 2.3. *Let $D \in \mathcal{E}^n$, with embedding dimension r. Let $\bar{D} := D[1\!:\!k] \in \mathcal{E}^k$ with embedding dimension t, and $B := \mathcal{K}^\dagger(\bar{D}) = \bar{U}_B S \bar{U}_B^T$, where $\bar{U}_B \in \mathcal{M}^{k\times t}$, $\bar{U}_B^T\bar{U}_B = I_t$, and $S \in \mathcal{S}_{++}^t$. Furthermore, let $U_B := \begin{bmatrix} \bar{U}_B & \frac{1}{\sqrt{k}}e \end{bmatrix} \in \mathcal{M}^{k\times(t+1)}$, $U := \begin{bmatrix} U_B & 0 \\ 0 & I_{n-k} \end{bmatrix}$, and let $\begin{bmatrix} V & \frac{U^Te}{\|U^Te\|} \end{bmatrix} \in \mathcal{M}^{n-k+t+1}$ be orthogonal. Then*

$$\text{face}\ \mathcal{K}^\dagger\left(\mathcal{E}^n(1\!:\!k,\bar{D})\right) = \left(U\mathcal{S}_+^{n-k+t+1}U^T\right) \cap \mathcal{S}_C = (UV)\mathcal{S}_+^{n-k+t}(UV)^T. \tag{2.5}$$

Proof. Let $Y \in \mathcal{K}^\dagger\left(\mathcal{E}^n(1\!:\!k,\bar{D})\right)$ and $\bar{Y} := Y[1\!:\!k]$. Then there exists $D \in \mathcal{E}^n(1\!:\!k,\bar{D})$ such that $Y = \mathcal{K}^\dagger(D)$, implying that $\mathcal{K}(Y) = D$, and that $\mathcal{K}(\bar{Y}) = \bar{D} = \mathcal{K}(B)$. Thus, $\bar{Y} \in B + \mathcal{N}(\mathcal{K}) = B + \mathcal{R}(\mathcal{D}_e)$, where the last equality follows from Proposition 1.2. This implies that $\bar{Y} = B + \mathcal{D}_e(y)$,

for some $y \in \mathbb{R}^k$. From Theorem 1.3, we get $\bar{Y} \succeq 0$ and $Be = 0$. Therefore, Lemma 2.1 implies that $y = Bu + \beta e$, for some $u \in \mathbb{R}^k$ and $\beta \geq 0$. This further implies

$$\bar{Y} = B + Bue^T + eu^T B + 2\beta ee^T.$$

From this expression for $\bar{Y}$, we can see that $\mathcal{R}(\bar{Y}) \subseteq \mathcal{R}\left(\begin{bmatrix} B & e \end{bmatrix}\right) = \mathcal{R}(U_B)$, where the last equality follows from the fact that $Be = 0$. Therefore, $\bar{Y} \in U_B \mathcal{S}_+^{t+1} U_B^T$, implying, by Lemma 2.2, that $\operatorname{face} \mathcal{S}_+^n(1:k, \bar{Y}) \trianglelefteq U\mathcal{S}_+^{n-k+t+1}U^T$. Since $Y \in \mathcal{S}_+^n(1:k, \bar{Y})$ and $Ye = 0$, we have that $Y \in \left(U\mathcal{S}_+^{n-k+t+1}U^T\right) \cap \mathcal{S}_C$. Therefore, $\operatorname{face} \mathcal{K}^\dagger\left(\mathcal{E}^n(1{:}k, \bar{D})\right) \trianglelefteq \left(U\mathcal{S}_+^{n-k+t+1}U^T\right) \cap \mathcal{S}_C$. Since $V^T U^T e = 0$, we have that

$$\left(U\mathcal{S}_+^{n-k+t+1}U^T\right) \cap \mathcal{S}_C = UV\mathcal{S}_+^{n-k+t}V^T U^T. \tag{2.6}$$

To show that $\operatorname{face} \mathcal{K}^\dagger\left(\mathcal{E}^n(1{:}k, \bar{D})\right) = \left(U\mathcal{S}_+^{n-k+t+1}U^T\right) \cap \mathcal{S}_C$, we need to find

$$\hat{Y} = UZU^T \in \mathcal{K}^\dagger\left(\mathcal{E}^n(1{:}k, \bar{D})\right), \quad \text{with } \operatorname{rank}(\hat{Y}) = n - k + t,\ \hat{Y}e = 0,\ Z \in \mathcal{S}_+^{n-k+t+1}. \tag{2.7}$$

To accomplish this, we let $T_1 = \begin{bmatrix} S & 0 \\ 0 & 1 \end{bmatrix}$. Then $T_1 \succ 0$ and

$$B + \frac{1}{k}ee^T = U_B T_1 U_B^T = \bar{P}\bar{P}^T, \quad \text{where } \bar{P} := U_B T_1^{1/2} \in \mathcal{M}^{\,k\times(t+1)}.$$

Let

$$P := \left[\begin{array}{c|c} \bar{P} & 0 \\ \hline 0 & I_{n-k-1} \\ -e^T\bar{P} & -e^T \end{array}\right] \in \mathcal{M}^{\,n\times(n-k+t)}.$$

Since $\bar{P}$ has full-column rank, we see that P also has full-column rank. Moreover, $P^T e = 0$. Therefore,

$$\hat{Y} := PP^T = \left[\begin{array}{c|cc} \bar{P}\bar{P}^T & 0 & -e \\ \hline 0 & I_{n-k-1} & -e \\ -e^T & -e^T & n-1 \end{array}\right] \in \mathcal{S}_+^n,$$

satisfies $\hat{Y}e = 0$ and $\operatorname{rank}(\hat{Y}) = n - k + t$. Furthermore, we have that $\hat{Y} = UZU^T$, where

$$Z = \left[\begin{array}{cc|cc} S & 0 & 0 & 0 \\ 0 & 1 & 0 & -\sqrt{k} \\ \hline 0 & 0 & I_{n-k-1} & -e \\ 0 & -\sqrt{k} & -e^T & n-1 \end{array}\right] \in \mathcal{S}^{n-k+t+1}.$$

Note that we can also write Z as

$$Z = \begin{bmatrix} S & 0 \\ 0 & T \end{bmatrix} \in \mathcal{S}^{n-k+t+1},$$

where

$$T := \begin{bmatrix} 1 & 0 & -\sqrt{k} \\ 0 & I_{n-k-1} & -e \\ -\sqrt{k} & -e^T & n-1 \end{bmatrix} \in \mathcal{S}^{n-k+1}.$$

The eigenvalues of T are 0, 1, and n, with multiplicities 1, $n-k-1$, and 1, respectively. Therefore, $\operatorname{rank}(T) = n-k$, which implies that $\operatorname{rank}(Z) = n-k+t$ and $Z \succeq 0$.

Letting $\hat{D} := \mathcal{K}(\hat{Y})$, we have that $\hat{D} \in \mathcal{E}^n(1\!:\!k, \bar{D})$, since

$$\hat{D}[1\!:\!k] = \mathcal{K}(\hat{Y}[1\!:\!k]) = \mathcal{K}(\bar{P}\bar{P}^T) = \mathcal{K}\left(B + \frac{1}{k}ee^T\right) = \mathcal{K}(B) = \bar{D}.$$

Therefore, $\hat{Y}$ satisfies (2.7), completing the proof. □

Remark 2.4. *Theorem 2.3 provides a reduction in the dimension of the* ***EDM*** *completion problem. Initially, our problem consists in finding $Y \in \mathcal{S}^n_+ \cap \mathcal{S}_C$ such that the constraint*

$$\mathcal{K}(Y[\alpha]) = D[\alpha], \quad \alpha = 1\!:\!k,$$

holds. After the reduction, we have the smaller dimensional variable $Z \in \mathcal{S}^{n-k+t}_+$; by construction $Y := (UV)Z(UV)^T$ will automatically satisfy the above constraints. This is a reduction of $k-t-1 = (n-1)-(n-k+t)$ in the dimension of the matrix variable. The addition of the vector e to the range of B, i.e. using $U_B := \begin{bmatrix}\bar{U}_B & \frac{1}{\sqrt{k}}e\end{bmatrix}$, has a geometric interpretation. If $B = PP^T$, $P \in \mathcal{M}^{k\times t}$, then the rows of P provide centered *positions for the k sensors in the clique α. However, these sensors are not necessarily centered once they are combined with the remaining $n-k$ sensors. Therefore, we have to allow for translations, e.g. to $P + ev^T$ for some v. The multiplication $(P+ev^T)(P+ev^T)^T = PP^T + Pve^T + ev^TP^T + ev^Tve^T$ is included in the set of matrices that we get after adding e to the range of B. Note that $Pve^T + ev^TP^T + ev^Tve^T = \mathcal{D}_e(y)$, for $y = Pv + \frac{1}{2}ev^Tv$.*

The special case $k = 1$ is of interest.

Corollary 2.5. *Suppose that the hypotheses of Theorem 2.3 hold but that $k = 1$ and $\bar{D} = 0$. Then $U_B = 1$, $U = I_n$, and*

$$\operatorname{face}\mathcal{K}^\dagger\left(\mathcal{E}^n(1\!:\!k,\bar{D})\right) = \operatorname{face}\mathcal{K}^\dagger\left(\mathcal{E}^n\right) = \mathcal{S}^n_+ \cap \mathcal{S}_C = V\mathcal{S}^{n-1}_+V^T, \tag{2.8}$$

where $\begin{bmatrix}V & \frac{1}{\sqrt{n}}e\end{bmatrix} \in \mathcal{M}^n$ is orthogonal.

Proof. Since $k = 1$, necessarily we get $t = 0$ and we can set $U_B = 1$. □

2.1.1 Disjoint Cliques Reduction

Theorem 2.3 can be easily extended to two or more disjoint cliques; see also [14].

Corollary 2.6. *Let $D \in \mathcal{E}^n$ with embedding dimension r. Let $k_0 := 1 < k_1 < \ldots < k_l \le n$. For $i = 1, \ldots, l$, let $\bar{D}_i := D[k_{i-1}\!:\!k_i] \in \mathcal{E}^{k_i-k_{i-1}+1}$ with embedding dimension t_i, $B_i := \mathcal{K}^\dagger(\bar{D}_i) = \bar{U}_{B_i}S\bar{U}^T_{B_i}$, where $\bar{U}_{B_i} \in \mathcal{M}^{k\times t_i}$, $\bar{U}^T_{B_i}\bar{U}_{B_i} = I_{t_i}$, $S_i \in \mathcal{S}^{t_i}_{++}$, and $U_{B_i} := \begin{bmatrix}\bar{U}_{B_i} & \frac{1}{\sqrt{k_i}}e\end{bmatrix} \in \mathcal{M}^{k\times(t_i+1)}$. Let*

$$U := \begin{bmatrix} U_{B_1} & \ldots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \ldots & U_{B_l} & 0 \\ 0 & \ldots & 0 & I_{n-k_l} \end{bmatrix}$$

and $\begin{bmatrix} V & \frac{U^T e}{\|U^T e\|} \end{bmatrix} \in \mathcal{M}^{\,n-k_l+\sum_{i=1}^{l} t_i+l}$ *be orthogonal. Then*

$$\begin{aligned} \bigcap_{i=1}^{l} \text{face } \mathcal{K}^{\dagger}\left(\mathcal{E}^n(k_{i-1}\!:\!k_i, \bar{D}_i)\right) &= \left(U\mathcal{S}_+^{n-k_l+\sum_{i=1}^{l} t_i+l} U^T\right) \cap \mathcal{S}_C \\ &= (UV)\mathcal{S}_+^{n-k_l+\sum_{i=1}^{l} t_i+l-1}(UV)^T. \end{aligned} \tag{2.9}$$

Proof. The result follows from noting that the range of U is the intersection of the ranges of the matrices U_{B_i} with appropriate identity blocks added. □

2.2 Two (Intersecting) Clique Reduction

The construction (2.6) illustrates how we can find the intersection of two faces. Using this approach, we now extend Theorem 2.3 to two cliques that (possibly) intersect; see the ordered indices in (2.10) and the corresponding Venn diagram in Figure 2.1. We also find expressions for the intersection of the corresponding faces in $\mathcal{S}_+^n$; see equation (2.12). The key is to find the intersection of the subspaces that represent the faces, as in condition (2.11).

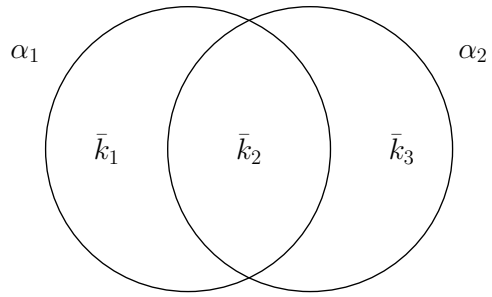

Figure 2.1: Venn diagram of the sets of ordered indices, α_1 and α_2, in Theorem 2.7

Theorem 2.7. *Let $D \in \mathcal{E}^n$ with embedding dimension r and, as in Figure 2.1, define the sets of positive integers*

$$\begin{gathered} \alpha_1 := 1\!:\!(\bar{k}_1+\bar{k}_2), \quad \alpha_2 := (\bar{k}_1+1)\!:\!(\bar{k}_1+\bar{k}_2+\bar{k}_3) \subseteq 1\!:\!n, \\ k_1 := |\alpha_1| = \bar{k}_1+\bar{k}_2, \quad k_2 := |\alpha_2| = \bar{k}_2+\bar{k}_3, \\ k := \bar{k}_1+\bar{k}_2+\bar{k}_3. \end{gathered} \tag{2.10}$$

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

$$\mathcal{R}(\bar{U}) = \mathcal{R}\left(\begin{bmatrix} U_1 & 0 \\ 0 & I_{\bar{k}_3} \end{bmatrix}\right) \cap \mathcal{R}\left(\begin{bmatrix} I_{\bar{k}_1} & 0 \\ 0 & U_2 \end{bmatrix}\right), \text{ with } \bar{U}^T\bar{U} = I_{t+1}. \tag{2.11}$$

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

$$\bigcap_{i=1}^{2} \text{face } \mathcal{K}^{\dagger}\left(\mathcal{E}^n(\alpha_i, \bar{D}_i)\right) = \left(U\mathcal{S}_+^{n-k+t+1}U^T\right) \cap \mathcal{S}_C = (UV)\mathcal{S}_+^{n-k+t}(UV)^T. \tag{2.12}$$

Proof. From Theorem 2.3, we have that

$$\text{face } \mathcal{K}^\dagger\left(\mathcal{E}^n(\alpha_1,\bar{D}_1)\right) = \left(\left[\begin{array}{cc|c} U_1 & 0 & 0 \\ 0 & I_{\bar{k}_3} & 0 \\ \hline 0 & 0 & I_{n-k} \end{array}\right] \mathcal{S}_+^{n-k_1+t_1+1} \left[\begin{array}{cc|c} U_1 & 0 & 0 \\ 0 & I_{\bar{k}_3} & 0 \\ \hline 0 & 0 & I_{n-k} \end{array}\right]^T\right) \cap \mathcal{S}_C$$

and, after a permutation of rows and columns in Theorem 2.3,

$$\text{face } \mathcal{K}^\dagger\left(\mathcal{E}^n(\alpha_2,\bar{D}_2)\right) = \left(\left[\begin{array}{cc|c} I_{\bar{k}_1} & 0 & 0 \\ 0 & U_2 & 0 \\ \hline 0 & 0 & I_{n-k} \end{array}\right] \mathcal{S}_+^{n-k_2+t_2+1} \left[\begin{array}{cc|c} I_{\bar{k}_1} & 0 & 0 \\ 0 & U_2 & 0 \\ \hline 0 & 0 & I_{n-k} \end{array}\right]^T\right) \cap \mathcal{S}_C.$$

The range space condition (2.11) then implies that

$$\mathcal{R}(U) = \mathcal{R}\left(\left[\begin{array}{cc|c} U_1 & 0 & 0 \\ 0 & I_{\bar{k}_3} & 0 \\ \hline 0 & 0 & I_{n-k} \end{array}\right]\right) \cap \mathcal{R}\left(\left[\begin{array}{cc|c} I_{\bar{k}_1} & 0 & 0 \\ 0 & U_2 & 0 \\ \hline 0 & 0 & I_{n-k} \end{array}\right]\right),$$

giving us the result (2.12). □

Remark 2.8. *Theorem 2.7 provides a reduction in the dimension of the* ***EDM*** *completion problem. Initially, our problem consists in finding $Y \in \mathcal{S}_+^n \cap \mathcal{S}_C$ such that the two constraints*

$$\mathcal{K}(Y[\alpha_i]) = D[\alpha_i], \quad i = 1, 2,$$

hold. After the reduction, we want to find the smaller dimensional $Z \in \mathcal{S}_+^{n-k+t}$; by construction $Y := (UV)Z(UV)^T$ will automatically satisfy the above constraints.

The explicit expression for the intersection of the two faces is given in equation (2.12) and uses the matrix $\bar{U}$ obtained from the intersection of the two ranges in condition (2.11). Finding a matrix whose range is the intersection of two subspaces can be done using [18, Algorithm 12.4.3]. However, our subspaces have special structure. We can exploit this structure to find the intersection; see Lemma (2.9) and Lemma (2.13) below.

The dimension of the face in (2.12) is reduced to $n - k + t$. However, we can get a dramatic reduction if we have a common block with embedding dimension r, and a reduction in the case the common block has embedding dimension $r - 1$ as well. This provides an algebraic proof using semidefinite programming of the rigidity of the union of the two cliques under this intersection assumption.

2.2.1 Nonsingular Reduction with Intersection Embedding Dimension r

We now consider the case when the intersection of the two cliques results in $D[\alpha_1 \cap \alpha_2]$ having embedding dimension r; see Figure 2.2. We see that we can explicitly find the completion of the ***EDM*** $D[\alpha_1 \cup \alpha_2]$. We first need the following result on the intersection of two structured subspaces.

Lemma 2.9. *Let*

$$U_1 := \begin{bmatrix} U_1' \\ U_1'' \end{bmatrix}, \quad U_2 := \begin{bmatrix} U_2'' \\ U_2' \end{bmatrix}, \quad \hat{U}_1 := \begin{bmatrix} U_1' & 0 \\ U_1'' & 0 \\ 0 & I \end{bmatrix}, \quad \hat{U}_2 := \begin{bmatrix} I & 0 \\ 0 & U_2'' \\ 0 & U_2' \end{bmatrix}$$

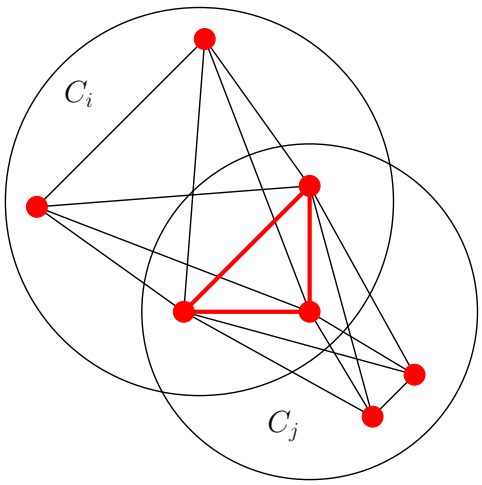

Figure 2.2: Two clique reduction with intersection with embedding dimension r

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

$$\bar{U}_1 := \begin{bmatrix} U_1' \\ U_1'' \\ U_2'(U_2'')^\dagger U_1'' \end{bmatrix}, \quad \bar{U}_2 := \begin{bmatrix} U_1'(U_1'')^\dagger U_2'' \\ U_2'' \\ U_2' \end{bmatrix}. \tag{2.13}$$

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

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

Moreover, if $e_l \in \mathbb{R}^l$ *is the* l^{th} *standard unit vector, and* $U_i e_l = \alpha_i e$, *for some* $\alpha_i \neq 0$, *for* $i = 1, 2$, *then* $\bar{U}_i e_l = \alpha_i e$, *for* $i = 1, 2$.

Proof. From the definitions, $x \in \mathcal{R}(\hat{U}_1) \cap \mathcal{R}(\hat{U}_2)$ if and only if

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} U_1' v_1 \\ U_1'' v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} w_1 \\ U_2'' w_2 \\ U_2' w_2 \end{bmatrix}, \quad \text{for some } v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \; w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}.$$

Note that $U_1'' v_1 = U_2'' w_2$ if and only if $w_2 = (U_2'')^\dagger U_1'' v_1$; this follows from the facts that U_2'' full column rank implies $(U_2'')^\dagger U_2'' = I$, and $\mathcal{R}(U_1'') = \mathcal{R}(U_2'')$ implies $U_2''(U_2'')^\dagger U_1'' = U_1''$. Therefore, $x \in \mathcal{R}(\hat{U}_1) \cap \mathcal{R}(\hat{U}_2)$ if and only if

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} U_1' v_1 \\ U_1'' v_1 \\ U_2'(U_2'')^\dagger U_1'' v_1 \end{bmatrix} = \bar{U}_1 v_1, \quad \text{for some } v_1,$$

with $v_2 := U_2'(U_2'')^\dagger U_1'' v_1$, $w_1 := U_1' v_1$, and $w_2 := (U_2'')^\dagger U_1'' v_1$, implying that $\mathcal{R}(\hat{U}_1) \cap \mathcal{R}(\hat{U}_2) = \mathcal{R}(\bar{U}_1)$; a similar argument shows that $\mathcal{R}(\hat{U}_1) \cap \mathcal{R}(\hat{U}_2) = \mathcal{R}(\bar{U}_2)$.

Now suppose, for $i = 1, 2$, that $U_i e_l = \alpha_i e$, for some $\alpha_i \neq 0$. Then $e \in \mathcal{R}(\hat{U}_1) \cap \mathcal{R}(\hat{U}_2)$, so $e \in \mathcal{R}(\bar{U}_1)$, implying that $\bar{U}_1 v = e$, for some vector v. Since $\bar{U}_1 = \begin{bmatrix} U_1 \\ U_2'(U_2'')^\dagger U_1'' \end{bmatrix}$, we have $U_1 v = e$.

Furthermore, since U_1 has full column rank, we conclude that $v = \frac{1}{\alpha_1} e_l$, implying that $\bar{U}_1 e_l = \alpha_1 e$. Similarly, we can show that $\bar{U}_2 e_l = \alpha_2 e$. □

We now state and prove a key result that shows we can complete the distances in the union of two cliques provided that their intersection has embedding dimension equal to r.

Theorem 2.10. *Let the hypotheses of Theorem 2.7 hold. Let*

$$\beta \subseteq \alpha_1 \cap \alpha_2, \quad \bar{D} := D[\beta], \quad B := \mathcal{K}^\dagger(\bar{D}), \quad \bar{U}_\beta := \bar{U}[\beta, :],$$

where $\bar{U} \in \mathcal{M}^{k\times(t+1)}$ *satisfies equation* (2.11). *Let* $\begin{bmatrix} \bar{V} & \frac{\bar{U}^T e}{\|\bar{U}^T e\|} \end{bmatrix} \in \mathcal{M}^{t+1}$ *be orthogonal. Let*

$$Z := (J\bar{U}_\beta \bar{V})^\dagger B((J\bar{U}_\beta \bar{V})^\dagger)^T. \tag{2.14}$$

If the embedding dimension for $\bar{D}$ *is* r, *then* $t = r$, $Z \in \mathcal{S}^r_{++}$ *is the unique solution of the equation*

$$(J\bar{U}_\beta \bar{V}) Z (J\bar{U}_\beta \bar{V})^T = B, \tag{2.15}$$

and

$$D[\alpha_1 \cup \alpha_2] = \mathcal{K}\left((\bar{U}\bar{V}) Z (\bar{U}\bar{V})^T\right). \tag{2.16}$$

Proof. Since the embedding dimension of $\bar{D}$ is r, we have $\operatorname{rank}(B) = r$. Furthermore, we have $Be = 0$ and $B \in \mathcal{S}^{|\beta|}_+$, implying that $|\beta| \geq r+1$. In addition, since the embedding dimension of D is also r, we conclude that the embedding dimension of $\bar{D}_i$ is r, for $i = 1, 2$. Similarly, the embedding dimension of $D[\alpha_1 \cap \alpha_2]$ is also r.

Since $\bar{U} \in \mathcal{M}^{k\times(t+1)}$ satisfies equation (2.11), we have that

$$\mathcal{R}(\bar{U}) = \mathcal{R}\left(\begin{bmatrix} U_1' & 0 \\ U_1'' & 0 \\ 0 & I_{\bar{k}_3} \end{bmatrix}\right) \cap \mathcal{R}\left(\begin{bmatrix} I_{\bar{k}_1} & 0 \\ 0 & U_2'' \\ 0 & U_2' \end{bmatrix}\right).$$

Note that we have partitioned $U_i = \begin{bmatrix} \bar{U}_i & \frac{1}{\sqrt{k_i}} e \end{bmatrix} \in \mathcal{M}^{k_i \times (r+1)}$ so that $U_i'' = \begin{bmatrix} \bar{U}_i'' & \frac{1}{\sqrt{k_i}} e \end{bmatrix} \in \mathcal{M}^{|\alpha_1 \cap \alpha_2| \times (r+1)}$, for $i = 1, 2$. Moreover, we have used the fact that the embedding dimension of $\bar{D}_i$ is r, so that $t_i = r$, for $i = 1, 2$.

We claim that U_1'' and U_2'' have full column rank and that $\mathcal{R}(U_1'') = \mathcal{R}(U_2'')$. First we let $Y := \mathcal{K}^\dagger(D[\alpha_1 \cup \alpha_2])$. Then $Y \in \mathcal{K}^\dagger\left(\mathcal{E}^k(\alpha_1, \bar{D}_1)\right)$. By Theorem 2.3, there exists $Z_1 \in \mathcal{S}^{\bar{k}_3 + r + 1}_+$ such that

$$Y = \begin{bmatrix} U_1' & 0 \\ U_1'' & 0 \\ 0 & I_{\bar{k}_3} \end{bmatrix} Z_1 \begin{bmatrix} U_1' & 0 \\ U_1'' & 0 \\ 0 & I_{\bar{k}_3} \end{bmatrix}^T.$$

Therefore, $Y[\alpha_1 \cap \alpha_2] = \begin{bmatrix} U_1'' & 0 \end{bmatrix} Z_1 \begin{bmatrix} U_1'' & 0 \end{bmatrix}^T \in U_1'' \mathcal{S}^{r+1}_+ (U_1'')^T$, so

$$\mathcal{R}(Y[\alpha_1 \cap \alpha_2]) \subseteq \mathcal{R}(U_1'').$$

Furthermore, since $\mathcal{K}(Y) = D[\alpha_1 \cup \alpha_2]$, we have that $\mathcal{K}(Y[\alpha_1 \cap \alpha_2]) = D[\alpha_1 \cap \alpha_2] = \mathcal{K}\left(\mathcal{K}^\dagger(D[\alpha_1 \cap \alpha_2])\right)$, so $Y[\alpha_1 \cap \alpha_2] \in \mathcal{K}^\dagger(D[\alpha_1 \cap \alpha_2]) + \mathcal{N}(\mathcal{K})$. Since $\mathcal{N}(\mathcal{K}) = \mathcal{R}(\mathcal{D}_e)$, there exists a vector y such that

$$Y[\alpha_1 \cap \alpha_2] = \mathcal{K}^\dagger(D[\alpha_1 \cap \alpha_2]) + \mathcal{D}_e(y) = \mathcal{K}^\dagger(D[\alpha_1 \cap \alpha_2]) + ye^T + ey^T.$$

By Lemma 2.1, $y \in \mathcal{R}\left(\begin{bmatrix}\mathcal{K}^\dagger(D[\alpha_1 \cap \alpha_2]) & e\end{bmatrix}\right)$. Therefore,

$$\mathcal{R}(Y[\alpha_1 \cap \alpha_2]) = \mathcal{R}\left(\begin{bmatrix}\mathcal{K}^\dagger(D[\alpha_1 \cap \alpha_2]) & e\end{bmatrix}\right).$$

Moreover, rank $\mathcal{K}^\dagger(D[\alpha_1 \cap \alpha_2]) = r$ and $\mathcal{K}^\dagger(D[\alpha_1 \cap \alpha_2])e = 0$, so

$$r + 1 = \dim \mathcal{R}(Y[\alpha_1 \cap \alpha_2]) \leq \dim \mathcal{R}(U_1'') \leq r + 1.$$

Therefore, U_1'' has full column rank and $\mathcal{R}(U_1'') = \mathcal{R}(Y[\alpha_1 \cap \alpha_2])$. Similarly, we can show that U_2'' has full column rank and $\mathcal{R}(U_2'') = \mathcal{R}(Y[\alpha_1 \cap \alpha_2])$, so we conclude that $\mathcal{R}(U_1'') = \mathcal{R}(U_2'')$.

We now claim that $t = r$, where $\bar{U} \in \mathcal{M}^{k\times(t+1)}$ satisfies equation (2.11). Since $U_1'', U_2'' \in \mathcal{M}^{|\alpha_1\cap\alpha_2|\times(r+1)}$ have full column rank and $\mathcal{R}(U_1'') = \mathcal{R}(U_2'')$, we have by Lemma 2.9 that $\mathcal{R}(\bar{U}) = \mathcal{R}(\bar{U}_1) = \mathcal{R}(\bar{U}_2)$, where

$$\bar{U}_1 := \begin{bmatrix} U_1' \\ U_1'' \\ U_2'(U_2'')^\dagger U_1'' \end{bmatrix} \quad \text{and} \quad \bar{U}_2 := \begin{bmatrix} U_1'(U_1'')^\dagger U_2'' \\ U_2'' \\ U_2' \end{bmatrix}.$$

Therefore,

$$t + 1 = \dim \mathcal{R}(\bar{U}) = \dim \mathcal{R}(\bar{U}_1) = \dim \mathcal{R}(\bar{U}_2) = r + 1,$$

so we have $t = r$, as claimed.

Recall, $Y = \mathcal{K}^\dagger(D[\alpha_1 \cup \alpha_2])$, so $Y \in \cap_{i=1,2}\mathcal{K}^\dagger\left(\mathcal{E}^k(\alpha_i, \bar{D}_i)\right)$. Thus, Theorem 2.7 implies that there exists $\bar{Z} \in \mathcal{S}_+^r$ such that $Y = (\bar{U}\bar{V})\bar{Z}(\bar{U}\bar{V})^T$. Observe that $\mathcal{K}(Y[\beta]) = D[\beta] = \bar{D}$. Thus,

$$\mathcal{K}\left((\bar{U}_\beta\bar{V})\bar{Z}(\bar{U}_\beta\bar{V})^T\right) = \bar{D},$$

implying that

$$\mathcal{K}^\dagger\,\mathcal{K}\left((\bar{U}_\beta\bar{V})\bar{Z}(\bar{U}_\beta\bar{V})^T\right) = B.$$

Since $\mathcal{K}^\dagger\,\mathcal{K}$ is the projection onto $\mathcal{R}(\mathcal{K}^*) = \mathcal{S}_C$, we have that $\mathcal{K}^\dagger\,\mathcal{K}(\cdot) = J(\cdot)J$. Therefore, we have that $\bar{Z}$ satisfies equation (2.15). It remains to show that equation (2.15) has a unique solution. Let $A := J\bar{U}_\beta\bar{V} \in \mathcal{M}^{|\beta|\times r}$. Then $A\bar{Z}A^T = B$ and rank $(B) = r$ implies that rank $(A) \geq r$, so A has full column rank. This implies that equation (2.15) has a unique solution, and that $\bar{Z} = A^\dagger B(A^\dagger)^T = Z$. Finally, since $Y = (\bar{U}\bar{V})Z(\bar{U}\bar{V})^T$ and $D[\alpha_1 \cup \alpha_2] = \mathcal{K}(Y)$, we get equation (2.16). □

The following result shows that if we know the minimal face of $\mathcal{S}_+^n$ containing $\mathcal{K}^\dagger(D)$, and we know a small submatrix of D, then we can compute a set of points in $\mathbb{R}^r$ that generate D by solving a small equation.

Corollary 2.11. *Let $D \in \mathcal{E}^n$ with embedding dimension r, and let $\beta \subseteq 1\!:\!n$. Let $U \in \mathcal{M}^{n\times(r+1)}$ satisfy*

$$\text{face}\,\mathcal{K}^\dagger(D) = \left(U\mathcal{S}_+^{r+1}U^T\right) \cap \mathcal{S}_C,$$

let $U_\beta := U[\beta, :]$, and let $\begin{bmatrix} V & \frac{U^Te}{\|U^Te\|}\end{bmatrix} \in \mathcal{M}^{r+1}$ be orthogonal. If $D[\beta]$ has embedding dimension r, then

$$(JU_\beta V)Z(JU_\beta V)^T = \mathcal{K}^\dagger(D[\beta])$$

has a unique solution $Z \in \mathcal{S}_{++}^r$, and $D = \mathcal{K}(PP^T)$, where $P := UVZ^{1/2} \in \mathbb{R}^{n\times r}$.

Proof. Apply Theorem 2.10 with $\alpha_1 = \alpha_2 = 1\!:\!n$. □

Remark 2.12. *A more efficient way to calculate Z uses the full rank factorization*

$$B = QD^{1/2}\left(QD^{1/2}\right)^T, \quad Q^TQ = I_r, \quad D \in \mathcal{S}^r_{++}.$$

Let $C = (J\bar{U}_\beta\bar{V})^\dagger\left(QD^{1/2}\right)$. Then Z in (2.14) can be found from $Z = CC^T$. Note that our algorithm postpones finding Z until the end, i.e. until we can no longer perform any clique reductions. At each iteration, we compute the matrix $\bar{U}$ that represents the face corresponding to the union of two cliques; $\bar{U}$ is chosen from one of $\bar{U}_i$, for $i = 1, 2$ in (2.13). Moreover, for stability, we maintain $\bar{U}^T\bar{U} = I$, $\bar{U}e_{r+1} = \alpha e$.

For many of our test problems, we can repeatedly apply Theorem 2.10 until there is only one clique left. Since each repetition reduces the number of cliques by one, this means that there are at most n such steps.

2.2.2 Singular Reduction with Intersection Dimension $r-1$

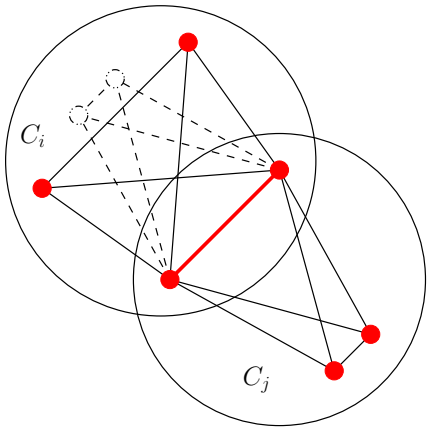

Figure 2.3: Two clique reduction with intersection having embedding dimension $< r$

We now show that if the embedding dimension of the intersection is $r-1$ (is deficient), then we can find at most two completions. And then, the correct one can generally be chosen by using the radio range R as a lower bound; see Figure 2.3. We first need the following extension of Lemma 2.9 on the intersection of two structured subspaces for the case where the common middle blocks are not full rank.

Lemma 2.13. *Let $U_i, \hat{U}_i, \bar{U}_i$, for $i = 1, 2$, be defined and appropriately blocked as in Lemma 2.9, with $U_i'' \in \mathcal{M}^{k\times(r+1)}$ having rank r, for $i = 1, 2$, and $\mathcal{R}(U_1'') = \mathcal{R}(U_2'')$. Let $0 \neq u_i \in \mathcal{N}(U_i'')$, for $i = 1, 2$. If $\bar{U} \in \mathcal{M}^{k\times(t+1)}$ satisfies $\mathcal{R}(\bar{U}) = \mathcal{R}(\hat{U}_1) \cap \mathcal{R}(\hat{U}_2)$, then $t = r + 1$ and*

$$\begin{aligned}\mathcal{R}(\bar{U}) &= \mathcal{R}\left(\begin{bmatrix} U_1' & 0 \\ U_1'' & 0 \\ U_2'(U_2'')^\dagger U_1'' & U_2'u_2 \end{bmatrix}\right) = \mathcal{R}\left(\begin{bmatrix} \bar{U}_1 & \begin{bmatrix} 0 \\ 0 \\ U_2'u_2 \end{bmatrix} \end{bmatrix}\right) \\ &= \mathcal{R}\left(\begin{bmatrix} U_1'(U_1'')^\dagger U_2'' & U_1'u_1 \\ U_2'' & 0 \\ U_2' & 0 \end{bmatrix}\right) = \mathcal{R}\left(\begin{bmatrix} \bar{U}_2 & \begin{bmatrix} U_1'u_1 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix}\right).\end{aligned} \tag{2.17}$$

Moreover, if $e_l \in \mathbb{R}^l$ is the l^{th} standard unit vector, and $U_i e_l = \alpha_i e$, for some $\alpha_i \neq 0$, for $i = 1, 2$, then $\bar{U}_i e_l = \alpha_i e$, for $i = 1, 2$.

Proof. From the definitions, $x \in \mathcal{R}(\bar{U})$ if and only if

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} U_1' v_1 \\ U_1'' v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} w_1 \\ U_2'' w_2 \\ U_2' w_2 \end{bmatrix}, \text{ for some } v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}. \tag{2.18}$$

Since $\mathcal{R}(U_1'') = \mathcal{R}(U_2'')$, and $U_i'', i = 1, 2$, are both rank r, we conclude that $x_2 = U_1'' v_1 = U_2'' w_2$, for some v_1, w_2 if and only if $x_2 \in \mathcal{R}(U_1'')$, with v_1, w_2 determined by

$$v_1 = (U_1'')^\dagger x_2 + \alpha_1 u_1, \text{ for some } \alpha_1 \in \mathbb{R}, \qquad w_2 = (U_2'')^\dagger U_1'' v_1 + \alpha_2 u_2, \text{ for some } \alpha_2 \in \mathbb{R}.$$

i.e. we get

$$\begin{gathered} x_2 = U_1'' v_1 = U_2'' w_2, \text{ for some } v_1, w_2, \\ \text{if and only if} \\ x_2 = U_1'' v_1, \text{ for some } v_1, \text{ with } w_2 = (U_2'')^\dagger U_1'' v_1 + \alpha_2 u_2, \text{ for some } \alpha_2 \in \mathbb{R}. \end{gathered} \tag{2.19}$$

After substituting for v_2 with $v_2 = U_2' w_2 = U_2' \left((U_2'')^\dagger U_1'' v_1 + \alpha_2 u_2 \right)$, we conclude that (2.18) holds if and only if the first equality in (2.17) holds, i.e. if and only if

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} U_1' v_1 \\ U_1'' v_1 \\ U_2' (U_2'')^\dagger U_1'' v_1 + \alpha_2 U_2' u_2 \end{bmatrix}, \text{ for some } v_1, \alpha_2,$$

where

$$v_2 = U_2' (U_2'')^\dagger U_1'' v_1 + \alpha_2 U_2' u_2, \quad w_1 = U_1' v_1, \quad w_2 = (U_2'')^\dagger U_1'' v_1 + \alpha_2 u_2.$$

The second equality in (2.17) follows similarly. The last statements about $\bar{U}_i e_\ell$ follow as in the proof of Lemma 2.9. □

In the rigid case in Theorem 2.10, we use the expression for $\bar{U}$ from Lemma 2.9 to obtain a unique Z in order to get the completion of $D[\alpha_1 \cup \alpha_2]$. The Z is unique because the $r+1$ columns of $\bar{U}$ that represent the new clique $\alpha_1 \cup \alpha_2$ are linearly independent, $e \in \mathcal{R}(\bar{U})$, $\text{rank}\,(B) = r$, and $Be = 0$. This means that the solution C of $(J\bar{U}_\beta \bar{V})C = QD^{1/2}$ in Remark 2.12 exists and is unique. (Recall that $J\bar{U}_\beta \bar{V}$ is full column rank.) This also means that the two matrices, U_1 and U_2, that represent the cliques, α_1 and α_2, respectively, can be replaced by the single matrix $\bar{U}$ without actually calculating C; we can use $\bar{U}$ to represent the clique $\alpha_1 \cup \alpha_2$ and complete all or part of the partial ***EDM*** $D[\alpha_1 \cup \alpha_2]$ only when needed.

We have a similar situation for the singular intersection case following Lemma 2.13. We have the matrix $\bar{U}$ to represent the intersection of the two subspaces, where each subspace represents one of the cliques, α_1 and α_2. However, this is not equivalent to *uniquely* representing the union of the two cliques, α_1 and α_2, since there is an extra column in $\bar{U}$ compared to the nonsingular case. In addition, since $\text{rank}\,(B) = r-1$, then $J\bar{U}_\beta \bar{V}$ is not necessarily full column rank. Therefore, there may be infinite solutions for C in Remark 2.12; any $C \in (J\bar{U}_\beta \bar{V})^\dagger \left(QD^{1/2} \right) + \mathcal{N}(J\bar{U}_\beta \bar{V})$ will give us a solution. Moreover, these solutions will not necessarily satisfy $\mathcal{K}\left((\bar{U}C)(\bar{U}C)^T \right) = D[\alpha_1 \cup \alpha_2]$. We now see that we can continue and use the $\bar{U}$ to represent a set of cliques rather than just $\alpha_1 \cup \alpha_2$. Alternatively, in many cases, we can use the radio range R as a lower bound and then evaluate the correct C in order to get the correct number of columns for $\bar{U}$; we can then get the correct completion of $D[\alpha_1 \cup \alpha_2]$.

Theorem 2.14. *Let the hypotheses of Theorem 2.10 hold with the special case that* $U_i^T U_i = I$, $U_i e_{r+1} = \alpha_i e$, *for* $i = 1, 2$. *In addition, let* $\bar{U}$ *be defined by one of the expressions in* (2.17) *in Lemma 2.13. For* $i = 1, 2$, *let* $\beta \subset \delta_i \subseteq \alpha_i$ *and* $A_i := J\bar{U}_{\delta_i}\bar{V}$, *where* $\bar{U}_{\delta_i} := \bar{U}(\delta_i, :)$. *Furthermore, let* $B_i := \mathcal{K}^\dagger(D[\delta_i])$, *define the linear system*

$$\begin{array}{rcl} A_1 Z A_1^T & = & B_1 \\ A_2 Z A_2^T & = & B_2, \end{array} \tag{2.20}$$

and let $\bar{Z} \in \mathcal{S}^t$ *be a particular solution of this system* (2.20). *If the embedding dimensions of* $D[\delta_1]$ *and* $D[\delta_2]$ *are both* r, *but the embedding dimension of* $\bar{D} := D[\beta]$ *is* $r - 1$, *then the following holds.*

1. $\dim \mathcal{N}(A_i) = 1$, *for* $i = 1, 2$.
2. *For* $i = 1, 2$, *let* $n_i \in \mathcal{N}(A_i)$, $\|n_i\|_2 = 1$, *and* $\Delta Z := n_1 n_2^T + n_2 n_1^T$. *Then,* Z *is a solution of the linear system* (2.20) *if and only if*

$$Z = \bar{Z} + \tau \Delta Z, \quad \text{for some } \tau \in \mathbb{R}. \tag{2.21}$$

3. *There are at most two nonzero solutions,* τ_1 *and* τ_2, *for the generalized eigenvalue problem* $-\Delta Z v = \tau \bar{Z} v$, $v \neq 0$. *Set* $Z_i := \bar{Z} + \frac{1}{\tau_i}\Delta Z$, *for* $i = 1, 2$. *Then*

$$D[\alpha_1 \cup \alpha_2] \in \left\{ \mathcal{K}(\bar{U}\bar{V} Z_i \bar{V}^T \bar{U}^T) : i = 1, 2 \right\}.$$

Proof. We follow a similar proof as in the nonsingular case. For simplicity, we assume that $\delta_i = \alpha_i$, for $i = 1, 2$ (choosing smaller δ_i can reduce the cost of solving the linear systems).

That a particular solution $\bar{Z}$ exists for the system (2.20), follows from the fact that $\bar{U}$ provides a representation for the intersection of the two faces (or the union of the two cliques).

Since the embedding dimension of $\bar{D}$ is $r - 1$, we have rank $(B) = r - 1$. Furthermore, we have $Be = 0$ and $B \in \mathcal{S}_+^{|\beta|}$, implying that $|\beta| \geq r$. Without loss of generality, and for simplicity, we assume that $|\beta| = r$. Therefore, there exists $0 \neq u_i \in \mathcal{N}(U_i'')$, for $i = 1, 2$. From Lemma 2.13, we can assume that we maintain $\bar{U}_i^T \bar{U}_i = I$, $\bar{U}_i e_{r+1} = \alpha_i e$, for some $\alpha_i \neq 0$, for $i = 1, 2$. Therefore, the action of $\bar{V}$ is equivalent to removing the $r + 1$ column of $\bar{U}_i$. We can then explicitly use u_i to write down $n_i \in \mathcal{N}(A_i)$. By construction, we now have $A_i(n_1 n_2^T + n_2 n_1^T)A_i^T = 0$, for $i = 1, 2$.

From the first expression for $\bar{U}$ in (2.17), we see that the choices for n_1 and n_2 in Part 1 are in the appropriate nullspaces. The dimensions follow from the assumptions on the embedding dimensions.

Part 2 now follows from the definition of the general solution of a linear system of equations, i.e. the sum of a particular solution with any solution of the homogeneous equation.

Part 3 now follows from the role that $\bar{U}$ plays as a representation for the union of the two cliques. □

Remark 2.15. *As above in the nonsingular case, a more efficient way to calculate* $\bar{Z}$ *uses the full rank factorization*

$$B_i = Q D^{1/2} \left(Q_i D_i^{1/2} \right)^T, \quad Q_i^T Q_i = I_r, \quad D_i \in \mathcal{S}_{++}^r, \quad i = 1, 2.$$

(We have assumed that both have embedding dimension r, *though we only need that one does.) We solve the equations* $A_i C = \left(Q_i D_i^{1/2} \right) \bar{Q}_i$, $\bar{Q}_i \bar{Q}_i^T = I$, *for* $i = 1, 2$, *for the unknowns* C, *and* $\bar{Q}_i$,

for $i = 1, 2$. Then a particular solution $\bar{Z}$ in (2.20) can be found from $\bar{Z} = CC^T$. Note that the additional orthogonal matrices $\bar{Q}_i$, for $i = 1, 2$ are needed since, they still allow $A_iC(A_iC)^T = B_i$, for $i = 1, 2$. Also, without loss of generality, we can assume $\bar{Q}_1 = I$.

2.3 Clique Initialization and Absorption

Using the above clique reductions, we now consider techniques that allow one clique to grow/absorb other cliques. This applies Theorem 2.10. We first consider an elementary and fast technique to find some of the existing cliques.

Lemma 2.16. *For each $i \in \{1, \ldots, n\}$, use half the radio range and define the set*

$$C_i := \left\{ j \in \{1, \ldots, n\} : D_{ij} \leq (R/2)^2 \right\}.$$

Then each C_i corresponds to a clique of sensors that are within radio range of each other.

Proof. Let $j, k \in C_i$ for a given $i \in \{1, \ldots, n\}$. An elementary application of the triangle inequality shows that $\sqrt{(D_{jk})} \leq \sqrt{(D_{ji})} + \sqrt{(D_{ki})} \leq R$. □

We can now assume that we have a set of indices $\mathcal{C} \subseteq 1\!:\!n$ corresponding to at most n cliques, $\{C_i\}_{i \in \mathcal{C}}$. We can combine cliques using the reductions given in Theorems 2.10 and 2.14. We now see how a clique can grow further by absorbing individual sensors; see Figure 2.4.

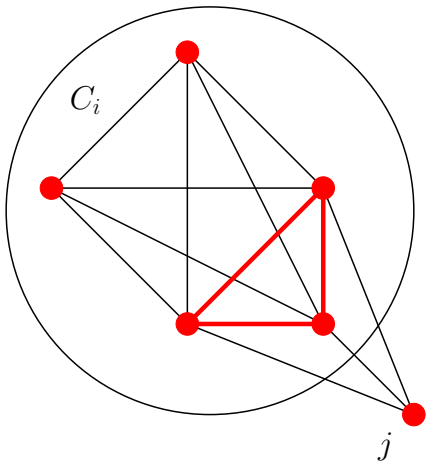

Figure 2.4: Absorption with intersection having embedding dimension r

Corollary 2.17. *Let C_k, for $k \in \mathcal{C}$, be a given clique with node $l \notin C_k$, $\beta := \{j_1, \ldots, j_{r+1}\} \subseteq C_k$, and $D_{lj_i} \leq R^2$, for $i = 1, \ldots, r+1$. (Alternatively, the distances D_{lj_i}, for $i = 1, \ldots, r+1$ are known.) If*

$$\operatorname{rank} \mathcal{K}^{\dagger}(D[\beta]) = r, \tag{2.22}$$

then l can be absorbed by the clique C_k and we can complete the missing elements in column (row) l of $D[C_k \cup \{l\}]$.

Proof. Let $\alpha_1 := C_k$, $\alpha_2 := \{j_1, \ldots, j_{r+1}, l\}$, and $\beta := \alpha_1 \cap \alpha_2 = \{j_1, \ldots, j_{r+1}\}$. Then the conditions in Theorem 2.10 are satisfied and we can recover all the missing elements in $D[C_k \cup \{l\}]$. □

2.3.1 Clique Absorption with Degenerate Intersection

We can apply the same reasoning as for the clique absorption in the nonsingular case, except now we apply Theorem 2.14. We still use the radio range as a lower bound to obtain a unique completion. See Figure 2.5.

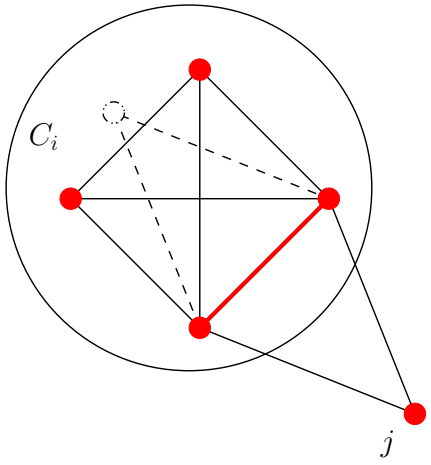

Figure 2.5: Degenerate absorption with intersection with embedding dimension $< r$

Corollary 2.18. *Let C_k, for $k \in \mathcal{C}$, be a given clique with node $l \notin C_k$, $\beta := \{j_1, \dots j_r\} \subseteq C_k$, and $D_{lj_i} \leq R^2$, for $i = 1, \dots, r$. (Alternatively, the distances D_{lj_i}, for $i = 1, \dots, r$ are known.) If*

$$\operatorname{rank} \mathcal{K}^\dagger(D[\beta]) = r - 1, \tag{2.23}$$

then we can use the lower bound given by the radio range, and l can be absorbed by the clique C_k. We can also complete the missing elements in column (row) l of $D[C_k \cup \{l\}]$.

Proof. Let $\alpha_1 := C_k$, $\alpha_2 := \{j_1, \dots, j_r, l\}$, and $\beta := \alpha_1 \cap \alpha_2 = \{j_1, \dots, j_r\}$. Then the conditions in Theorem 2.14 are satisfied and we can recover all the missing elements in $D[C_k \cup \{l\}]$. □

3 Clique Reduction Algorithm and Numerical Results

Our algorithm starts by forming a clique C_i around each sensor i using Lemma 2.16. If and when we use this clique, we find a subspace representation from the r eigenvectors corresponding to the r nonzero eigenvalues of $B = \mathcal{K}^\dagger(D[C_i])$.

The algorithm then grows and combines cliques using Theorem 2.10, Theorem 2.14, Corollary 2.17, and Corollary 2.18. In particular, we do not complete the $\boldsymbol{EDM}$ each time we combine or grow cliques, i.e. we do not evaluate the missing distances. Instead, we use the subspace representations of the corresponding faces of the $\boldsymbol{SDP}$ cone and then find the intersection of the subspaces that represent the faces. This yields a subspace representation of the new smaller face representing the larger clique. This is based on Lemma 2.9 and Lemma 2.13 and is therefore inexpensive.

Once we cannot, or need not, grow cliques, we complete the distances using Corollary 2.11. This is also inexpensive. Finally, we rotate and translate the anchors to their original positions using the approach outlined in [14].

n / R	0.9	0.8	0.7	0.6	0.5	0.4
1000	23.5	18.8	14.6	10.8	7.6	5.0
2000	47.0	37.5	29.0	21.5	15.1	9.7
3000	70.6	56.3	43.5	32.3	22.6	14.6
4000	94.1	75.0	58.0	42.9	30.1	19.4
5000	117.6	93.7	72.4	53.7	37.6	24.3
6000	141.3	112.6	87.1	64.5	45.3	29.2
7000	164.7	131.4	101.5	75.2	52.7	34.0
8000	188.3	150.1	116.0	86.0	60.2	38.9
9000	211.9	169.0	130.6	96.8	67.8	43.8
10000	235.6	187.8	145.1	107.5	75.3	48.7

Table 3.1: Average Degree of Graph

3.1 Numerical Tests

Our tests use 9 anchors positioned in a 10×10 square, as in Figure 1.1. The sensors are placed randomly, by means of a uniform random distribution, in the same square region. We vary the number of sensors n from 1000 to 9000 in steps of 1000, and the radio range R from .9 to .4 in steps of $-.1$. See Table 3.1 for the average degree of a vertex (sensor) of the graph. Each number of our output is the average taken over 10 instances. Our tests were done using MATLAB version 7.8.0.347 (R2009a) on a DELL T7400 running Windows Vista, with a 2.50 GHz Core 4 processor and with 4 GB of RAM.

We in particular emphasize the low CPU times and the high accuracy of the solutions. Our algorithm compares well with the recent work in [22, 26], where they use, for example, $R = .6$ for $n = 1000, 2000$, $R = .35$ for $n = 4000$, $R = .2$ for $n = 10000$, and also use 10% of the sensors as anchors and limit the degree for each node in order to maintain a low sparsity for the graph.

3.1.1 Nonsingular Subspace Intersection

In our first set of tests we used only the nonsingular subspace intersection step (Theorem 2.10); see Table 3.2.

1. The remaining cliques at the end of the algorithm. This number is at least 1; and it is exactly 1 if all the sensors were found, i.e. if the $\boldsymbol{EDM}$ was completed.

2. CPU seconds.

3. The maximum distance between the positions of the sensors found and the original positions of those sensors, defined as

$$\text{Max Error} := \max_{i \text{ positioned}} \|p_i - p_i^{true}\|_2.$$

4. The root-mean-square deviation of the positions of the sensors found and the original positions of those sensors, defined as

$$\text{RMSD} := \left(\frac{1}{\#\text{ positioned}} \sum_{i \text{ positioned}} \|p_i - p_i^{true}\|_2^2 \right)^{\frac{1}{2}}.$$

Remaining Cliques

n / R	0.9	0.8	0.7	0.6	0.5	0.4
1000	2.6	8.3	44.6	128.1	242.6	387.7
2000	1.0	1.0	1.2	4.3	79.9	333.8
3000	1.0	1.0	1.0	1.4	5.1	142.5
4000	1.0	1.0	1.0	1.0	1.4	19.2
5000	1.0	1.0	1.0	1.0	1.0	4.2
6000	1.0	1.0	1.0	1.0	1.0	1.6
7000	1.0	1.0	1.0	1.0	1.0	1.1
8000	1.0	1.0	1.0	1.0	1.0	1.0
9000	1.0	1.0	1.0	1.0	1.0	1.0
10000	1.0	1.0	1.0	1.0	1.0	1.0

CPU Seconds

n / R	0.9	0.8	0.7	0.6	0.5	0.4
1000	1.7	1.6	1.5	1.4	1.3	0.9
2000	3.3	3.2	3.2	3.2	3.2	3.0
3000	5.1	5.0	4.8	4.7	4.7	4.7
4000	6.5	6.6	6.7	6.6	6.5	6.4
5000	8.4	8.4	8.5	8.6	8.5	8.3
6000	10.9	10.6	10.6	10.8	10.7	10.5
7000	14.1	13.2	12.7	12.9	13.0	12.8
8000	18.0	16.3	15.7	15.3	15.2	14.7
9000	21.8	19.7	18.3	17.8	17.9	17.4
10000	27.7	24.0	21.5	21.0	20.3	20.2

log(Max Error)

n / R	0.9	0.8	0.7	0.6	0.5	0.4
1000	−11.2	−10.7	–	–	–	–
2000	−11.6	−11.2	−10.7	−10.4	–	–
3000	−11.6	−11.3	−11.1	−10.7	−9.9	–
4000	−11.7	−11.5	−11.0	−10.9	−10.4	−9.6
5000	−11.7	−11.5	−11.1	−11.0	−10.6	−9.8
6000	−11.9	−11.5	−11.3	−11.0	−10.5	−10.1
7000	−12.0	−11.7	−11.2	−11.1	−10.5	−10.1
8000	−11.8	−11.6	−11.5	−11.0	−10.8	−10.2
9000	−11.9	−11.7	−11.5	−11.2	−10.8	−10.2
10000	−12.0	−11.7	−11.2	−11.2	−10.8	−10.4

log(RMSD)

n / R	0.9	0.8	0.7	0.6	0.5	0.4
1000	−11.7	−11.5	–	–	–	–
2000	−12.1	−11.6	−11.2	−10.9	–	–
3000	−12.2	−11.9	−11.5	−11.3	−10.5	–
4000	−12.2	−12.0	−11.4	−11.4	−10.9	−10.3
5000	−12.2	−12.0	−11.5	−11.4	−11.0	−10.3
6000	−12.4	−12.0	−11.8	−11.5	−11.0	−10.6
7000	−12.4	−12.2	−11.7	−11.7	−11.0	−10.5
8000	−12.2	−12.2	−11.9	−11.5	−11.2	−10.7
9000	−12.4	−12.2	−11.9	−11.7	−11.1	−10.7
10000	−12.5	−12.2	−11.8	−11.6	−11.4	−10.8

Table 3.2: Results using nonsingular subspace intersection step (Theorem 2.10)

3.1.2 Nonsingular Subspace Intersection with Node Absorption

Our next tests raise the level of the algorithm and use Theorem 2.10 and Corollary 2.17. The results are in Table 3.3. We see that the number of successul completions has increased and that there has been a small increase in CPU time.

3.1.3 Singular Subspace Intersection and Node Absorption

Using the singular intersection and node absorption further increases the class of problems that we can complete, i.e. we can use Theorem 2.14 and Corollary 2.18. These tests are still ongoing. Preliminary results using Theorem 2.10, Corollary 2.17, and Theorem 2.14, are given in Table 3.4. Again, the set of problems that are completed (1 clique remaining) has increased.

4 Conclusion

The ***SDP*** relaxation of ***SNL*** is highly (implicitly) degenerate, since the feasible set of this ***SDP*** is restricted to a low dimensional face of the ***SDP*** cone, resulting in the failure of the Slater constraint qualification (strict feasibility). We take advantage of this degeneracy by finding explicit representations of intersections of faces of the ***SDP*** cone corresponding to unions of intersecting cliques. In addition, from these representations we force further degeneracy in order to find the minimal face that contains the optimal solution. In many cases, we can efficiently compute the exact solution to the ***SDP*** relaxation without using any ***SDP*** solver.

Our numerical tests show that the CPU times are extremely low, while the accuracy of the solutions is very high. These tests are ongoing.

Remaining Cliques

n / R	0.9	0.8	0.7	0.6	0.5	0.4
1000	1.0	1.2	2.5	14.9	156.4	351.9
2000	1.0	1.0	1.0	1.1	2.6	73.8
3000	1.0	1.0	1.0	1.0	1.1	3.8
4000	1.0	1.0	1.0	1.0	1.0	1.5
5000	1.0	1.0	1.0	1.0	1.0	1.0
6000	1.0	1.0	1.0	1.0	1.0	1.0
7000	1.0	1.0	1.0	1.0	1.0	1.0
8000	1.0	1.0	1.0	1.0	1.0	1.0
9000	1.0	1.0	1.0	1.0	1.0	1.0
10000	1.0	1.0	1.0	1.0	1.0	1.0

CPU Seconds

n / R	0.9	0.8	0.7	0.6	0.5	0.4
1000	1.5	1.6	2.0	3.0	3.1	2.2
2000	3.0	3.0	3.0	3.1	4.1	7.5
3000	4.7	4.8	4.7	4.7	4.8	6.6
4000	6.5	6.6	6.6	6.5	6.4	6.7
5000	8.3	8.3	8.4	8.6	8.4	8.3
6000	10.8	10.5	10.8	10.9	10.8	10.6
7000	14.3	13.7	13.3	13.3	13.3	12.9
8000	17.8	16.3	15.9	15.7	15.7	15.1
9000	22.4	20.3	18.7	17.6	17.7	17.2
10000	27.2	23.8	21.4	21.0	20.4	20.1

log(Max Error)

n / R	0.9	0.8	0.7	0.6	0.5	0.4
1000	−11.2	−10.8	−10.4	−9.4	–	–
2000	−11.6	−11.2	−10.7	−10.4	−9.5	–
3000	−11.6	−11.3	−11.1	−10.7	−9.9	−9.2
4000	−11.7	−11.5	−11.0	−10.9	−10.4	−9.4
5000	−11.7	−11.5	−11.1	−11.0	−10.6	−9.8
6000	−11.9	−11.5	−11.3	−11.0	−10.5	−10.1
7000	−12.0	−11.7	−11.2	−11.1	−10.5	−10.1
8000	−11.8	−11.6	−11.5	−11.0	−10.8	−10.2
9000	−11.9	−11.7	−11.5	−11.2	−10.8	−10.2
10000	−12.0	−11.7	−11.2	−11.2	−10.8	−10.4

log(RMSD)

n / R	0.9	0.8	0.7	0.6	0.5	0.4
1000	−11.7	−11.5	−11.0	−10.2	–	–
2000	−12.1	−11.6	−11.2	−10.9	−10.2	–
3000	−12.2	−11.9	−11.5	−11.3	−10.5	−9.8
4000	−12.2	−12.0	−11.4	−11.4	−10.9	−10.3
5000	−12.2	−12.0	−11.5	−11.4	−11.0	−10.3
6000	−12.4	−12.0	−11.8	−11.5	−11.0	−10.6
7000	−12.4	−12.2	−11.7	−11.7	−11.0	−10.5
8000	−12.2	−12.2	−11.9	−11.5	−11.2	−10.7
9000	−12.4	−12.2	−11.9	−11.7	−11.1	−10.7
10000	−12.5	−12.2	−11.8	−11.6	−11.4	−10.8

Table 3.3: Results using nonsingular subspace intersection with node absorption (Theorem 2.10 and Corollary 2.17)

Remaining Cliques

n / R	0.9	0.8	0.7	0.6	0.5	0.4
1000	1.0	1.2	2.0	9.4	93.0	260.0
2000	1.0	1.0	1.0	1.1	2.2	35.7
3000	1.0	1.0	1.0	1.0	1.1	3.4
4000	1.0	1.0	1.0	1.0	1.0	1.1
5000	1.0	1.0	1.0	1.0	1.0	1.0
6000	1.0	1.0	1.0	1.0	1.0	1.0
7000	1.0	1.0	1.0	1.0	1.0	1.0
8000	1.0	1.0	1.0	1.0	1.0	1.0
9000	1.0	1.0	1.0	1.0	1.0	1.0

CPU Seconds

n / R	0.9	0.8	0.7	0.6	0.5	0.4
1000	1.6	1.8	2.3	6.1	6.3	6.5
2000	3.3	3.3	3.2	3.3	5.3	39.1
3000	5.1	5.1	5.1	5.1	5.4	12.4
4000	6.9	7.0	7.1	7.1	7.0	10.7
5000	8.8	9.0	9.1	9.1	9.1	9.0
6000	11.6	11.2	11.3	11.4	11.3	11.1
7000	14.7	14.0	13.5	13.7	13.7	13.4
8000	19.0	17.1	16.4	16.1	16.1	15.7
9000	24.1	20.8	19.5	18.9	19.1	18.5

log(Max Error)

n / R	0.9	0.8	0.7	0.6	0.5	0.4
1000	−11.2	−10.8	−10.3	−8.7	–	–
2000	−11.6	−11.2	−10.7	−10.4	−9.5	−6.9
3000	−11.6	−11.3	−11.1	−10.7	−10.0	−9.1
4000	−11.7	−11.5	−11.0	−10.9	−10.4	−9.4
5000	−11.7	−11.5	−11.1	−11.0	−10.6	−9.8
6000	−11.9	−11.5	−11.4	−11.0	−10.5	−10.1
7000	−12.0	−11.7	−11.2	−11.2	−10.5	−10.1
8000	−11.8	−11.6	−11.5	−11.0	−10.8	−10.2
9000	−11.9	−11.7	−11.5	−11.2	−10.8	−10.2

log(RMSD)

n / R	0.9	0.8	0.7	0.6	0.5	0.4
1000	−11.7	−11.5	−10.9	−9.8	–	–
2000	−12.1	−11.6	−11.2	−10.9	−10.2	−8.2
3000	−12.2	−11.9	−11.5	−11.3	−10.5	−9.7
4000	−12.2	−12.0	−11.4	−11.4	−10.9	−10.3
5000	−12.3	−12.0	−11.5	−11.4	−11.0	−10.3
6000	−12.4	−12.0	−11.8	−11.5	−11.0	−10.6
7000	−12.4	−12.2	−11.7	−11.7	−11.0	−10.6
8000	−12.2	−12.2	−11.9	−11.5	−11.2	−10.7
9000	−12.4	−12.2	−11.9	−11.7	−11.1	−10.7

Table 3.4: Results using nonsingular subspace intersection, nonsingular node absorption, and singular subspace intersection (Theorem 2.10, Corollary 2.17, and Theorem 2.14)

References

[1] S. AL-HOMIDAN and H. WOLKOWICZ. Approximate and exact completion problems for Euclidean distance matrices using semidefinite programming. *Linear Algebra Appl.*, 406:109–141, 2005. 6, 7

[2] A. ALFAKIH, M.F. ANJOS, N. KRISLOCK, V. PICCIALLI, and H. WOLKOWICZ. Euclidean distance matrices, semidefinite programming, and sensor network localization. Technical Report in progress, University of Waterloo, Waterloo, Ontario, 2009. 2, 3

[3] A. ALFAKIH, A. KHANDANI, and H. WOLKOWICZ. Solving Euclidean distance matrix completion problems via semidefinite programming. *Comput. Optim. Appl.*, 12(1-3):13–30, 1999. Computational optimization—a tribute to Olvi Mangasarian, Part I. 2, 3

[4] B. AMES and S.A. VAVASIS. Nuclear norm minimization for the planted clique and biclique problems. Technical report, University of Waterloo, 2009. 4

[5] P. BISWAS. *Semidefinite programming approaches to distance geometry problems*. PhD thesis, Stanford University, 2007. 2, 3

[6] 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 Transactions on Automation Science and Engineering*, 3:360–371, 2006. 3

[7] P. BISWAS, T.C. LIANG, K.C. TOH, and Y. YE. An SDP based approach for anchor-free 3D graph realization. Technical report, Operation Research, Stanford University, Stanford, CA, 2005. 3

[8] P. BISWAS, K.C. TOH, and Y. YE. A distributed SDP approach for large-scale noisy anchor-free graph reailzation with applications to molecular conformation. *SIAM J. Sci. Comput.*, 30(3):1251–1277, 2008. 3

[9] P. BISWAS and Y. YE. Semidefinite programming for ad hoc wireless sensor network localization. In *Information Processing In Sensor Networks, Proceedings of the third international symposium on Information processing in sensor networks*, pages 46–54, Berkeley, Calif., 2004. 2, 3

[10] P. BISWAS and Y. YE. A distributed method for solving semidefinite programs arising from ad hoc wireless sensor network localization. In *Multiscale optimization methods and applications*, volume 82 of *Nonconvex Optim. Appl.*, pages 69–84. Springer, New York, 2006. 2, 3

[11] M.W. CARTER, H.H. JIN, M.A. SAUNDERS, and Y. YE. SpaseLoc: an adaptive subproblem algorithm for scalable wireless sensor network localization. *SIAM J. Optim.*, 17(4):1102–1128 (electronic), 2006. 3

[12] A. CASSIOLI. *Global optimization of highly multimodal problems*. PhD thesis, Universita di Firenze, Dipartimento di sistemi e informatica, Via di S.Marta 3, 50139 Firenze, Italy, 2008. 3

[13] J. DATTORRO. *Convex Optimization & Euclidean Distance Geometry*. Meboo Publishing, USA, 2005. 2, 3

[14] Y. DING, N. KRISLOCK, J. QIAN, and H. WOLKOWICZ. Sensor network localization, Euclidean distance matrix completions, and graph realization. *Optimization and Engineering*, to appear(CORR 2006-23, to appear), 2006. 2, 3, 7, 9, 11, 21

[15] Y. DING, N. KRISLOCK, J. QIAN, and H. WOLKOWICZ. Sensor network localization, Euclidean distance matrix completions, and graph realization. In *Proceedings of MELT08, San Francisco*, pages 129–134, 2008. 7

[16] T. EREN, D.K. GOLDENBERG, W. WHITELEY, Y.R. YANG, A.S. MORSE, B.D.O. ANDERSON, and P.N. BELHUMEUR. Rigidity, computation, and randomization in network localization, 2004. IEEE INFOCOM. 3

[17] U. FEIGE and R KRAUTHGAMER. Finding and certifying a large hidden clique in a semi-random graph. *Random Structures and Algorithms*, 16(2):195–202, 2000. 4

[18] G.H. GOLUB and C.F. VAN LOAN. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, 3^{nd} edition, 1996. 13

[19] B. HENDRICKSON. *The molecule problem: Determining conformation from pairwise distances*. PhD thesis, Cornell University, Ithaca, New York, 1991. 4

[20] B. HENDRICKSON. Conditions for unique graph realizations. *SIAM J. Comput.*, 21(1):65–84, 1992. 4

[21] H. JIN. *Scalable Sensor Localization Algorithms for Wireless Sensor Networks*. PhD thesis, Toronto University, Toronto, Ontario, Canada, 2005. 3

[22] T.K. PONG and P. TSENG. (Robust) edge-based semidefinite programming relaxation of sensor network localization. Technical Report Jan-09, University of Washington, Seattle, WA, 2009. 2, 22

[23] J. B. SAXE. Embeddability of weighted graphs in k-space is strongly NP-hard. *Proc. 17th Allerton Conf. in Communications, Control, and Computing*, pages 480–489, 1979. 4

[24] A.M-C. SO and Y. YE. Theory of semidefinite programming for sensor network localization. *Math. Program.*, 109(2-3, Ser. B):367–384, 2007. 3, 4

[25] R.J. VANDERBEI and Y. BING. The simplest semidefinite programs are trivial. *Math. Oper. Res.*, 20(3):590–596, 1995. 4

[26] Z. WANG, S. ZHENG, S. BOYD, and Y. YE. Further relaxations of the semidefinite programming approach to sensor network localization. *SIAM J. Optim.*, 19(2):655–673, 2008. 3, 22

[27] H. WOLKOWICZ. Explicit solutions for interval semidefinite linear programs. *Linear Algebra Appl.*, 236:95–104, 1996. 4

Index