Semidefinite Facial Reduction for Low-Rank Euclidean Distance Matrix Completion

by

Nathan Krislock

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Combinatorics & Optimization

Waterloo, Ontario, Canada, 2010

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The main result of this thesis is the development of a theory of semidefinite facial reduction for the Euclidean distance matrix completion problem. Our key result shows a close connection between cliques in the graph of the partial Euclidean distance matrix and faces of the semidefinite cone containing the feasible set of the semidefinite relaxation. We show how using semidefinite facial reduction allows us to dramatically reduce the number of variables and constraints required to represent the semidefinite feasible set. We have used this theory to develop a highly efficient algorithm capable of solving many very large Euclidean distance matrix completion problems exactly, without the need for a semidefinite optimization solver. For problems with a low level of noise, our `SNLSDPclique` algorithm outperforms existing algorithms in terms of both CPU time and accuracy. Using only a laptop, problems of size up to 40,000 nodes can be solved in under a minute and problems with 100,000 nodes require only a few minutes to solve.

Acknowledgements

There are so many for whom I am deeply grateful for their role in helping me during my doctoral studies and in the preparation of this thesis.

First of all, I would like to thank my supervisor, Henry Wolkowicz. The inspiration, guidance, and support that Henry has so generously given me has been tremendous and invaluable. I simply cannot offer enough thanks for everything he has done for me – I am extremely grateful.

A great thanks is given to the members of my examination committee: Forbes Burkowski, Kim-Chuan Toh, Levent Tunçel, and Stephen Vavasis. I am very grateful for their comments and helpful suggestions on my thesis. In particular, I would like to thank Forbes for his help with getting me started with using Chimera together with Python to produce some very nice molecular figures to include in my thesis.

Having the opportunity to visit Franz Rendl at the University of Klagenfurt in Austria for two weeks in 2009 was a truly amazing experience. I would like to sincerely thank Franz for his idea of using point representations in my algorithm, leading to much improved accuracy of the numerical results. I would also like to sincerely thank Franz for his support and advice.

I am deeply grateful to my academic older brother, Miguel Anjos, for his kind support and many discussions we have had during my time at the University of Waterloo. I am especially grateful for Miguel's wisdom and advice on careers in academia, and for his encouragement.

For my coauthors Babak Alipanahi and Ali Ghodsi, I would like to sincerely thank you for your support and enthusiasm for my results. The discussions I had with Babak are among my most rewarding and cherished times at the University of Waterloo.

I owe a heartfelt thanks to the staff members, faculty members, and my fellow graduate students in the Department of Combinatorics & Optimization. There are so many people from C&O who have greatly helped me in my academic journey, and I am forever grateful.

I would especially like to thank the members of the University of Waterloo Continuous Optimization Seminar for the many excellent talks and wonderful discussions we had week after week. This seminar was truly one of the best and most memorable experiences I have from my time at the University of Waterloo.

To my friends who were there for me, especially in times of need, thank you so much for everything. I would especially like to thank my dear friend Jason Hinek for his extreme generosity and support – I really cannot thank you enough.

To my loving and supportive parents, Monique and Don, thank you for believing in me – it has really meant a lot to me. Thanks to my brother Abram for our long discussions giving me the chance to unwind when I really needed it.

Thank you to my family in Japan for welcoming me with such open arms and true kindness. You always made me feel so much at home. Thank you so much for all your support over the years.

Above all, I thank my dear wife Misato for her endless support, guidance, and love. The joy and encouragement she brings me is beyond description. This dream has come true because of you. Thank you.

To our wonderful dog Chana, thank you for bringing so much happiness to our lives everyday these past four years – it has truly been an adventure. Thanks Chana for your loving company next to me everyday while I was writing this thesis.

This work has been partially funded by an NSERC Doctoral Postgraduate Scholarship.

Dedication

This thesis is dedicated to Takefumi Sekita (January 17, 1944 – March 23, 2010).

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Measuring and studying the distances between objects in some collection is a fundamental geometric question of great importance in many scientific areas. The forward computation is straightforward: Given a collection of n points, determine the distances between every pair of those points. The result is $\binom{n}{2} = \frac{1}{2}n(n-1)$ distances. We store the squares of the distances in an n-by-n symmetric matrix, having zeros along its diagonal. Such a matrix is called a *distance matrix*, and when the distance is computed using the *Euclidean norm*,

$$\|x\|_2 = \left(\sum_{i=1}^{r} x_i^2\right)^{1/2}, \qquad \text{where } x = (x_1, \ldots, x_n) \in \mathbb{R}^r,$$

this distance matrix is called a *Euclidean distance matrix*.

The reverse question asks to determine the location of the points in some collection given only the distances between those points, or determine that it is not possible to do so. An early reference studying this problem is Schoenberg (1935). It is in this classic paper where the relationship between *semidefinite matrices* and Euclidean distance matrices is first established: a matrix D is a Euclidean distance matrix if and only if a certain linear map of D is a positive semidefinite matrix. Moreover, factoring this semidefinite matrix then provides the locations of all the points.

Euclidean distance matrices have always been interesting due to their broad applicability in many areas. Diverse areas of scientific study work with large collections of data, and are often interested in measuring this data in terms of the distances between the points in the data. However, Euclidean distance matrices have received increased attention lately for two main reasons.

First of all, there are some very important application areas that are currently in need of ever better methods for using distance information to analyze data and determine the positions of points in some low dimensional space. One important application is sensor network localization. With the advancement of computer and wireless technology, wireless sensors are becoming smaller, more computationally capable, and more inexpensive, causing wireless sensor networks to become more and more ubiquitous. Ad hoc wireless sensor networks can be used to efficiently measure sound, temperature, smoke, vibrations, moisture, pressure, et cetera, in a variety of environments. As these networks are ad hoc, it is important to be able to determine the positions of the sensors after they have been (randomly) placed in the environment. Distances between sensors can be measured, but only between sensors that are within a certain range of communication. The power required to perform computations on these sensors turns out to be much less than the power required to communicate wirelessly with nearby sensors. Therefore, in order to keep these sensors small and inexpensive, their communication range (or radio range) is limited. This means that we will not be able to measure all pairwise distances, allowing only partial inter-sensor distance information to be measured. This partial distance information together with the positions of the anchor nodes (sensors with known position) is then used to try to determine the positions of the sensors. Although the Global Positioning System (GPS) can be used to accurately determine positions, its limitations are that it is often too bulky, requires too much power, and can only be used in an outside environment.

Another application of Euclidean distance matrices is in the study of proteins, specifically protein folding and determining the shape, or conformation, of protein molecules. By understanding the shape of proteins, we are better able to understand how different drugs will interact with these molecules; this is especially important in the area of computer-aided drug design. Interatomic distances can be measured using nuclear magnetic resonance (NMR) techniques, but only between atoms in the molecule that are within a limited range. The relative positions of the atoms can then be determined from this partial distance information using Euclidean distance matrix techniques. A distinguishing feature of the molecular conformation problem over the sensor network localization problem is that are no anchored points – we are only interested in the relative positions of the points.

Finally, we mention the application of multi-dimensional scaling in machine learning and statistics. In machine learning, it is of great interest to determine if a collection of data points can be placed in a lower dimensional space while retaining distances between nearby points. The data can be viewed as residing on a low-dimensional manifold in a high-dimensional space. The goal is to somehow unfold this manifold to obtain the desired low dimensional realization, allowing for hidden structure in the data to be revealed. Applications in this area are, for example, hand-written character recognition and detecting facial expressions.

The second reason for the increased interest in the study of Euclidean distance matrices is due to recent theoretical and algorithmic advances regarding semidefinite matrix problems. We are just recently able to solve semidefinite optimization problems efficiently following the recent discovery of interior-point methods for semidefinite optimization; see, for example, Nesterov and Nemirovskii (1994), Alizadeh (1995), Vandenberghe and Boyd (1996), and Todd (2001). This theoretical and computational advancement now allows us to easily compute solutions to Euclidean distance matrix problems that were until recently, very difficult problems to solve numerically. In addition to these algorithmic advances, there has also been some very nice theoretical and algorithmic results discovered in the closely related area of semidefinite matrix completion, most notably Grone et al. (1984) and Johnson et al. (1998). These theoretical and computational advancements in the area of semidefinite matrix completion has therefore also led to a renewed interest in Euclidean distance matrix problems, due to the close connection between these two problems. However, it should be noted that, in general, the semidefinite matrix completion is an easier problem than the low-dimensional Euclidean distance matrix problems we consider here.

Before discussing the contributions of this thesis, we provide a literature review of the main applied and theoretical areas related to our work.

1.1 Related work

The previous work within the scope of this thesis falls into five main categories:

1. distance geometry and Euclidean distance matrices;
2. graph realization and graph rigidity;
3. facial reduction;
4. sensor network localization;
5. molecular conformation.

Note that these categories are by no means disjoint, so we will focus on the main papers from each topic. As the number of references in these areas is numerous, the following does not comprise the complete list of references on these topics, but instead aims to highlight the key references from each area.

1.1.1 Distance geometry and Euclidean distance matrices

The foundational papers in the area of Euclidean distance matrices are Schoenberg (1935) and Young and Householder (1938). The topic was further developed with the series of papers Gower (1982, 1984, 1985), Farebrother (1987), and Critchley (1988). For papers on the Euclidean distance matrix completion problem and the related semidefinite completion problem, see the classic paper on semidefinite completion Grone et al. (1984), and follow-up papers Bakonyi and Johnson (1995) and Johnson and Tarazaga (1995); also see Laurent (2001) on the topic of the complexity of these completion problems. More on the topic of uniqueness of Euclidean distance matrix completions can be found in the papers Alfakih (2003, 2005). The cone of Euclidean distance matrices and its geometry is described in, for example, Glunt et al. (1990), Hayden et al. (1991), Tarazaga et al. (1996), Tarazaga (2005), and Alfakih (2006b). Using semidefinite optimization to solve Euclidean distance matrix problems is studied in the papers Alfakih et al. (1999) and Al-Homidan and Wolkowicz (2005). Further theoretical results are given in Alfakih and Wolkowicz (2002) and Alfakih (2006a). Books and survey papers containing a treatment of Euclidean distance matrices include, for example, Blumenthal (1970), Laurent (1998), Dattorro (2008), and most recently Alfakih et al. (2009). The topic of rank minimization for Euclidean distance matrix problems is discussed in, for example, Fazel (2002), Fazel et al. (2003), Recht et al. (2008a), Candès and Recht (2008), Recht et al. (2008b), Candès and Plan (2009).

1.1.2 Graph realization and graph rigidity

The complexity of graph realization in a fixed dimension was determined to be NP-hard by Saxe (1979) and Yemini (1979). For studies on graph rigidity, see, for example, Hendrickson (1992), Hendrickson (1995), Jackson and Jordán (2005), Connelly (2005), Belk (2007), Belk and Connelly (2007), Alfakih (2000, 2001, 2007), and the references therein. Graph rigidity for sensor network localization is studied as graph rigidity with some nodes being grounded or anchored; see, for example, Eren et al. (2004) and So and Ye (2007). Semidefinite optimization techniques have also been applied to graph realization and graph rigidity problems; see, for example, Barvinok (1995), So and Ye (2006), and the PhD thesis So (2007).

1.1.3 Facial reduction

Early papers studying faces of convex cones are, for example, Barker (1973), Barker and Carlson (1975), and Barker (1978). An excellent general reference for faces of convex cones is Schneider (1993). The use of facial reduction as a regularization technique for convex optimization problems was pioneered in the papers Borwein and Wolkowicz (1981a,b,c). Moreover, (Borwein and Wolkowicz, 1981a, Section 7) includes an early treatment of facial reduction for semidefinite optimization problems. Facial reduction was successfully used in Zhao et al. (1998) for solving semidefinite relaxations of the quadratic assignment and graph partitioning problems. Facial reduction was later studied by Pataki (2000), and applied to general semidefinite optimization problems by Ramana et al. (1997). Facial reduction techniques have recently resurfaced in a variety of applications; in addition to our own work, see, for example, Waki and Muramatsu (2009a,b).

1.1.4 Sensor network localization

While semidefinite relaxations were discussed earlier in Alfakih et al. (1999) for Euclidean distance matrix problems, the first semidefinite relaxations specialized for the sensor network localization problem were proposed by Doherty et al. (2001).

The paper by Biswas and Ye (2004) followed with what is now called the Biswas-Ye semidefinite relaxation of sensor network localization problem. As problems with only a few hundred sensors could be solved directly using the Biswas-Ye relaxation, Jin (2005) and Carter et al. (2006) proposed the scalable SpaseLoc semidefinite-based build-up algorithm which solves small subproblems using the Biswas-Ye semidefinite relaxation, locating a few sensors at a time. In the follow-up paper Biswas et al. (2006) propose regularization and refinement techniques for handling noisy problems using the Biswas-Ye semidefinite relaxation. In order to handle larger problems, a distributed method was proposed in Biswas and Ye (2006) which clusters points together in small groups, solves the smaller subproblems, then stitches the clusters together; see also the PhD thesis Biswas (2007). To allow the solution of larger problems, Wang et al. (2008) proposed further relaxations of the sensor network localization problem in which they do not insist that the full n-by-n matrix of the Biswas-Ye relaxation be positive semidefinite, but rather only that certain submatrices of this matrix be positive semidefinite; the most successful of these further relaxations is the so-called edge-based SDP relaxation, or ESDP. A noise-aware robust version of the ESDP relaxation called ρ-ESDP is proposed by Pong and Tseng (2010) and which is solved with their Log-barrier Penalty Coordinate Gradient Descent (LPCGD) method. Using techniques from Fukuda et al. (2001), it is shown in Kim et al. (2009a) how to form an equivalent sparse version of the full Biswas-Ye relaxation, called SFSDP. The sparsity in the SFSDP formulation can then be exploited by a semidefinite optimization solver, allowing the solution of noisy instances of the sensor network localization problem with up to 18000 sensors and 2000 anchors to high accuracy in under ten minutes; see Kim et al. (2009b). Most recently, we have shown in Krislock and Wolkowicz (2010) how to use facial reduction to solve a semidefinite relaxation of the sensor network localization problem. Our algorithm is able to solve noiseless problems with up to 100,000 sensors and 4 anchors to high accuracy in under six minutes on a laptop computer; see Table 5.3.

Other relaxations have also been studied; Tseng (2007) considers a second-order cone (SOC) relaxation of the sensor network localization problem, while Nie (2009) studies the sum-of-squares (SOS) relaxation of this problem.

For more applied approaches and general heuristics, see, for example, Bulusu et al. (2000), Savvides et al. (2001), Moore et al. (2004), Megerian et al. (2005), Costa et al. (2006), Nawaz (2008), Yang et al. (2009), Bruck et al. (2009), Stoyanova et al. (2009), and Cassioli (2009). A older survey paper on wireless sensor networks is Akyildiz et al. (2002); for a recent book on wireless ad hoc and sensor network, see Li (2008).

The complexity of the sensor network localization problem is discussed in Aspnes et al. (2004) and Aspnes et al. (2006). References for the single sensor localization problem are, for example, Beck et al. (2008a,b).

1.1.5 Molecular conformation

An early algorithmic treatment of molecular conformation is Havel et al. (1983) in which they give their bound embedding algorithm EMBED. This paper was then followed by the book Crippen and Havel (1988); a review paper Crippen (1991) provides an update three years after the publication of this book. A personal historical perspective is given in Havel (2003).

Other algorithmic developments followed, including: a divide-and-conquer algorithm called ABBIE based on identifying rigid substructures Hendrickson (1990); an alternating projection approach Glunt et al. (1993); a global smoothing continuation code called DGSOL Moré and Wu (1997, 1999); a geometric build-up algorithm Dong and Wu (2002, 2003), Wu and Wu (2007), Wu et al. (2008); an extended recursive geometric build-up algorithm dos Santos Carvalho et al. (2008); a difference of convex functions (d.c.) optimization algorithm An and Tao (2003); a method based on rank-reducing perturbations of the distance matrix that maintain desired structures Emiris and Nikitopoulos (2005); an algorithm for solving a distance matrix based, large-scale, bound constrained, non-convex optimization problem called StrainMin Grooms et al. (2009).

Recently, semidefinite optimization approaches to the molecular conformation problem have been studied in Leung and Toh (2008) and Biswas et al. (2008), for example; see also the PhD thesis Biswas (2007).

1.2 Contributions of this thesis

At this time, it is still very difficult to solve large-scale semidefinite optimization problems. For this reason, we can also only directly solve Euclidean distance matrix problems that are of limited size. However, the applications pull us in the other direction, asking us to be able to solve Euclidean distance matrix problems of larger and larger size. To be able to solve problems of such large size requires special techniques. We show in this thesis that the technique of semidefinite facial reduction allows us to do just that. We will see that with facial reduction, we are able to transform the semidefinite optimization problems corresponding to Euclidean distance matrix problems into equivalent problems requiring much fewer variables and constraints. With the facial reduction algorithm we have developed in this work, we are able to exactly solve many Euclidean distance matrix completion problems that are of very large size, up to a hundred-thousand points, in just a few minutes on a laptop computer, without the need of a semidefinite optimization solver. For problems with a low level of noise, our algorithm outperforms existing algorithms in terms of both CPU time and accuracy. Moreover, not only is the Euclidean distance matrix facial reduction theory we have developed in this work a very nice set of theoretical results, but it also has many far reaching consequences for the various applications we have mentioned above.

Before continuing, we mention that the Euclidean distance matrix problems we are solving are in the class of NP-hard problems. Therefore, we do not expect to be able to efficiently solve all instances of this problem. However, we found that our algorithm was able to solve most of the large problems we tested it on, very efficiently. As the Euclidean distance matrix facial reduction theory we have developed here has opened up many questions, this is just the beginning of a fruitful research project. Although we do not give an answer to all questions here, this work is ongoing and we are very excited by all the possibilities it brings.

Chapter 2

Faces and Euclidean Distance Matrices

We begin by defining the important concepts and terminology we will use. We also include many basic results which we will frequently use in later sections. Proofs of these results have been included for completeness.

2.1 Euclidean spaces

We work with real finite dimensional *Euclidean spaces* $\mathbb{E}$ equipped with an inner-product $\langle \cdot , \cdot \rangle : \mathbb{E} \times \mathbb{E} \to \mathbb{R}$, such as:

- the space of real n-vectors, $\mathbb{R}^n$, with

$$\langle x, y \rangle := x^T y = \sum_{i=1}^{n} x_i y_i;$$

- the space of real m-by-n matrices, $\mathbb{R}^{m \times n}$, with

$$\langle X, Y \rangle := \operatorname{trace}(X^T Y) = \sum_{i=1}^{m} \sum_{j=1}^{n} X_{ij} Y_{ij};$$

- the space of n-by-n real *symmetric matrices*, $\mathcal{S}^n := \left\{ X \in \mathbb{R}^{n \times n} : X = X^T \right\}$, with

$$\langle X, Y \rangle := \operatorname{trace}(XY) = \sum_{i,j=1}^{n} X_{ij} Y_{ij}.$$

The *norm* of $x \in \mathbb{E}$ is defined as $\|x\| := \sqrt{\langle x, x \rangle}$. In $\mathbb{R}^n$, we have the *Euclidean norm*, $\|x\|_2 := \sqrt{x^T x}$; in $\mathbb{R}^{m \times n}$ (or $\mathcal{S}^n$), we have the *Frobenius norm*, $\|X\|_F := \sqrt{\operatorname{trace}(X^T X)}$.

2.2 Linear maps

We use the MATLAB notation $1{:}n := \{1, \ldots, n\}$ and

$$[A; B] := \begin{bmatrix} A \\ B \end{bmatrix}.$$

For a matrix $A \in \mathbb{R}^{m \times n}$ and sets $S \subseteq 1{:}m$ and $T \subseteq 1{:}n$, we use the notation $A[S, T]$ to represent the *submatrix* of A formed by the rows indexed by S and columns indexed by T. We define:

$$\begin{aligned} A[:, T] &:= A[1{:}m, T]; \\ A[S, :] &:= A[S, 1{:}n]; \\ A[S] &:= A[S, S]. \end{aligned}$$

2.2.1 The adjoint and the fundamental subspaces

Let $\mathbb{E}$ and $\mathbb{Y}$ be Euclidean spaces equipped with the inner-products $\langle \cdot\,, \cdot \rangle_{\mathbb{E}}$ and $\langle \cdot\,, \cdot \rangle_{\mathbb{Y}}$, respectively. The *adjoint* of a linear map $A \colon \mathbb{E} \to \mathbb{Y}$ is the unique linear map $A^* \colon \mathbb{Y} \to \mathbb{E}$ satisfying

$$\langle x, A^* y \rangle_{\mathbb{E}} = \langle Ax, y \rangle_{\mathbb{Y}}\,, \quad \text{for all } x \in \mathbb{E} \text{ and } y \in \mathbb{Y}.$$

For ease of notation, we will often omit the subscripts on the inner-products as they are typically clear from the context. A linear map $A \colon \mathbb{E} \to \mathbb{E}$ is called *self-adjoint* if $A^* = A$.

We use $\operatorname{range}(A)$ and $\operatorname{null}(A)$ to denote the *range space* and *null space* of a linear map $A \colon \mathbb{E} \to \mathbb{Y}$, respectively. These fundamental subspaces satisfy the following well-known duality relationships.

Theorem 2.1. *Let $\mathbb{E}$ and $\mathbb{Y}$ be Euclidean spaces, and $A \colon \mathbb{E} \to \mathbb{Y}$ be linear. Then*

$$\operatorname{range}(A)^\perp = \operatorname{null}(A^*) \quad \textit{and} \quad \operatorname{null}(A)^\perp = \operatorname{range}(A^*).$$

Proof. First we note that $A^* y = 0$ if and only if

$$\langle Ax, y \rangle = \langle x, A^* y \rangle = 0, \quad \text{for all } x \in \mathbb{E}.$$

Therefore, $\operatorname{range}(A)^\perp = \operatorname{null}(A^*)$. To show that $\operatorname{null}(A)^\perp = \operatorname{range}(A^*)$, we will show that the equivalent statement $\operatorname{null}(A) = \operatorname{range}(A^*)^\perp$ holds. This is accomplished by noting that $Ax = 0$ if and only if

$$\langle x, A^* y \rangle = \langle Ax, y \rangle = 0, \quad \text{for all } y \in \mathbb{Y}. \qquad \square$$

The *rank* of a linear map $A \colon \mathbb{E} \to \mathbb{Y}$ is defined as $\operatorname{rank}(A) := \dim(\operatorname{range}(A))$. By Theorem 2.1, $\operatorname{rank}(A) = \operatorname{rank}(A^*)$ implies, and follows from, the Rank-Nullity Theorem: $\dim(\operatorname{null}(A)) + \operatorname{rank}(A) = \dim(\mathbb{E})$.

2.2.2 Orthogonal projections

A linear map $P\colon \mathbb{E} \to \mathbb{E}$ is an *orthogonal projection* if $P^* = P$ and $P^2 = P$. If $P\colon \mathbb{E} \to \mathbb{E}$ is an orthogonal projection, then $I - P$ is also an orthogonal projection; P projects $x \in \mathbb{E}$ orthogonally onto the subspace $\operatorname{range}(P)$ and $I - P$ projects $x \in \mathbb{E}$ orthogonally onto the subspace $\operatorname{range}(I - P) = \operatorname{range}(P)^\perp = \operatorname{null}(P)$. The proof of these facts follow easily from the observation that if $P\colon \mathbb{E} \to \mathbb{E}$ is an orthogonal projection, then

$$\langle (I-P)x, Py\rangle = \langle P(I-P)x, y\rangle = \left\langle (P-P^2)x, y\right\rangle = 0, \quad \text{for all } x, y \in \mathbb{E}.$$

Moreover, $P^2 = P$ implies that

$$\operatorname{range}(P) = \{x \in \mathbb{E} : Px = x\}.$$

We now show that an orthogonal projection onto a given subspace is unique.

Proposition 2.2. *Let V be a subspace of a Euclidean space $\mathbb{E}$. Then there is a unique orthogonal projection $P\colon \mathbb{E} \to \mathbb{E}$ such that* $\operatorname{range}(P) = V$*.*

Proof. Suppose P_1 and P_2 are orthogonal projections such that

$$\operatorname{range}(P_1) = \operatorname{range}(P_2) = V.$$

Then, for $i = 1, 2$,

$$V = \{v \in \mathbb{E} : P_i v = v\} \quad \text{and} \quad V^\perp = \operatorname{null}(P_i).$$

Thus, for all $x = v + w \in V \oplus V^\perp = \mathbb{E}$, we have

$$P_1 x = P_1(v + w) = v = P_2(v + w) = P_2 x,$$

so $P_1 = P_2$. □

2.2.3 The Moore-Penrose pseudoinverse

The *Moore-Penrose pseudoinverse* of a linear map $A\colon \mathbb{E} \to \mathbb{Y}$ is defined as the linear map $A^\dagger\colon \mathbb{Y} \to \mathbb{E}$ that satisfies the *Penrose conditions*:

$$AA^\dagger = (AA^\dagger)^* \tag{2.1a}$$
$$A^\dagger A = (A^\dagger A)^* \tag{2.1b}$$
$$AA^\dagger A = A \tag{2.1c}$$
$$A^\dagger A A^\dagger = A^\dagger. \tag{2.1d}$$

It is well-known that every linear map A has a *singular value decomposition*

$$A(\cdot) = \sum_{i=1}^{r} \sigma_i \langle v_i, \cdot\rangle\, u_i,$$

where $\operatorname{rank}(A) = r$, $\{u_1, \ldots, u_r\} \subseteq \mathbb{Y}$ is an orthonormal basis for the subspace $\operatorname{range}(A)$, and $\{v_1, \ldots, v_r\} \subseteq \mathbb{E}$ is an orthonormal basis for the subspace $\operatorname{range}(A^*)$; see, for example Golub and Van Loan (1996). It is easy to verify that the linear map given by

$$A^\dagger(\cdot) = \sum_{i=1}^{r} \frac{1}{\sigma_i} \langle u_i, \cdot \rangle \, v_i$$

satisfies the Penrose conditions (2.1). For example, we have

$$\begin{aligned}
AA^\dagger Ax &= A\left(\sum_{j=1}^{r} \frac{1}{\sigma_j} \langle u_j, Ax \rangle \, v_j\right) \\
&= A\left(\sum_{j=1}^{r} \frac{1}{\sigma_j} \sigma_j \langle v_j, x \rangle \, v_j\right) \\
&= A\left(\sum_{j=1}^{r} \langle v_j, x \rangle \, v_j\right) \\
&= \sum_{i=1}^{r} \sigma_i \langle v_i, x \rangle \, u_i \\
&= Ax,
\end{aligned}$$

for all $x \in \mathbb{E}$, so $AA^\dagger A = A$.

We will prove below in Theorem 2.3 that the Penrose conditions are uniquely satisfied, but first we give a few implications of these conditions. If $P = AA^\dagger$, then $P = P^*$ and $P^2 = AA^\dagger AA^\dagger = AA^\dagger = P$. Therefore, $AA^\dagger$ is an orthogonal projection:

- $AA^\dagger$ projects orthogonally onto the subspace $\operatorname{range}(AA^\dagger) = \operatorname{range}(A)$;
- $I - AA^\dagger$ projects orthogonally onto the subspace $\operatorname{range}(A)^\perp = \operatorname{null}(A^*)$.

Similarly, $A^\dagger A$ is an orthogonal projection:

- $I - A^\dagger A$ projects orthogonally onto the subspace $\operatorname{range}(I - A^\dagger A) = \operatorname{null}(A)$;
- $A^\dagger A$ projects orthogonally onto the subspace $\operatorname{null}(A)^\perp = \operatorname{range}(A^*)$.

The proofs for all these facts follow easily from the above Penrose conditions.

Theorem 2.3 ((Björck, 1996, Theorem 1.2.11)). *Let $\mathbb{E}$ and $\mathbb{Y}$ be Euclidean spaces, and $A\colon \mathbb{E} \to \mathbb{Y}$ be linear. Then the Penrose conditions* (2.1) *are satisfied by a unique linear map from $\mathbb{Y}$ to $\mathbb{E}$.*

Proof. We have already seen that the Penrose conditions (2.1) are satisfied by at least one linear map. Now suppose that there are two linear maps $X_i\colon \mathbb{Y} \to \mathbb{E}$, $i = 1, 2$, which satisfy the Penrose conditions (2.1). From the above discussion, we have that AX_1, AX_2, X_1A, and X_2A are orthogonal projections. By uniqueness of orthogonal projections (Proposition 2.2), we have that

$$AX_1 = AX_2 \quad \text{and} \quad X_1A = X_2A.$$

This fact together with condition (2.1d) implies that

$$X_1 = X_1AX_1 = X_1AX_2 = X_2AX_2 = X_2. \qquad \square$$

We now relate the four fundamental subspaces of $A^\dagger$ and A. Note that, by Theorem 2.3, it is easy to show that $(A^\dagger)^* = (A^*)^\dagger$, so we can omit the brackets without ambiguity.

Proposition 2.4. *Let $\mathbb{E}$ and $\mathbb{Y}$ be Euclidean spaces, and $A\colon \mathbb{E} \to \mathbb{Y}$ be linear. Then*

$$\operatorname{range}(A^\dagger) = \operatorname{range}(A^*), \qquad \operatorname{null}(A^{\dagger *}) = \operatorname{null}(A), \tag{2.2}$$

$$\operatorname{range}(A^{\dagger *}) = \operatorname{range}(A), \qquad \operatorname{null}(A^\dagger) = \operatorname{null}(A^*). \tag{2.3}$$

Proof. We first prove (2.2). Let $v = A^\dagger y \in \operatorname{range}(A^\dagger)$. Then

$$v = A^\dagger y = A^\dagger A A^\dagger y \in \operatorname{range}(A^*),$$

since $A^\dagger A$ is the orthogonal projection onto $\operatorname{range}(A^*)$. Conversely, if $v \in \operatorname{range}(A^*)$, then $v = A^\dagger A v \in \operatorname{range}(A^\dagger)$. Therefore, $\operatorname{range}(A^\dagger) = \operatorname{range}(A^*)$, from which $\operatorname{null}(A^{\dagger *}) = \operatorname{null}(A)$ immediately follows.

To prove (2.3), we first let $y \in \operatorname{null}(A^\dagger)$. Then, projecting y onto $\operatorname{null}(A^*)$, we obtain

$$(I - AA^\dagger)y = y - AA^\dagger y = y,$$

since $A^\dagger y = 0$. Therefore, $y \in \operatorname{null}(A^*)$. Conversely, let $y \in \operatorname{null}(A^*)$. Then, projecting y onto $\operatorname{null}(A^*)$, we obtain

$$y = (I - AA^\dagger)y = y - AA^\dagger y,$$

so $AA^\dagger y = 0$. Therefore, $A^\dagger y = A^\dagger A A^\dagger y = 0$, so $y \in \operatorname{null}(A^\dagger)$. Therefore, $\operatorname{null}(A^\dagger) = \operatorname{null}(A^*)$, from which $\operatorname{range}(A^{\dagger *}) = \operatorname{range}(A)$ immediately follows. $\square$

The following result describes the bijective property of A and $A^\dagger$ between the subspaces $\operatorname{range}(A^\dagger)$ and $\operatorname{range}(A)$.

Proposition 2.5. *Let $\mathbb{E}$ and $\mathbb{Y}$ be Euclidean spaces, and $A\colon \mathbb{E} \to \mathbb{Y}$ be linear. Then $A\colon \operatorname{range}(A^\dagger) \to \operatorname{range}(A)$ is a bijection and $A^\dagger\colon \operatorname{range}(A) \to \operatorname{range}(A^\dagger)$ is its inverse.*

Proof. To show that $A\colon \text{range}(A^\dagger) \to \text{range}(A)$ is bijective, we first let $y \in \text{range}(A)$. Then $y = Ax$, for some $x = u + v \in \text{null}(A) \oplus \text{null}(A)^\perp = \mathbb{E}$. Therefore, $y = A(u + v) = Av$. Since $v \in \text{null}(A)^\perp = \text{range}(A^*) = \text{range}(A^\dagger)$, we have that $y \in A\text{range}(A^\dagger)$. Therefore, $\text{range}(A) \subseteq A\text{range}(A^\dagger)$, so A maps $\text{range}(A^\dagger)$ onto $\text{range}(A)$. Now suppose that $Ax = Ay$ for some $x, y \in \text{range}(A^\dagger) = \text{null}(A)^\perp$. Then $x - y \in \text{null}(A)$, so

$$\|x - y\|^2 = \langle x - y, x - y\rangle = \langle x, x - y\rangle - \langle y, x - y\rangle = 0,$$

implying that $x = y$. Therefore, $A\colon \text{range}(A^\dagger) \to \text{range}(A)$ is also injective. To complete the proof, we must simply show that $A^\dagger\colon \text{range}(A) \to \text{range}(A^\dagger)$ is the inverse of the bijection $A\colon \text{range}(A^\dagger) \to \text{range}(A)$. To this end, we let $x \in \text{range}(A^\dagger)$ and $y := Ax$. Since $A^\dagger A$ is the orthogonal projection onto $\text{range}(A^*) = \text{range}(A^\dagger)$, we have $A^\dagger y = A^\dagger Ax = x$, completing the proof. □

We will find the following property of the Moore-Penrose pseudoinverse to be of great use.

Proposition 2.6. *Let $\mathbb{E}$ and $\mathbb{Y}$ be Euclidean spaces, $A\colon \mathbb{E} \to \mathbb{Y}$ be linear, $x \in \mathbb{E}$, and $b \in \mathbb{Y}$. Then*

$$Ax = b \quad \Leftrightarrow \quad b \in \text{range}(A) \;\; \textit{and} \;\; x \in A^\dagger b + \text{null}(A).$$

Proof. First we suppose that $Ax = b$. Clearly we have $b \in \text{range}(A)$. Since $AA^\dagger$ is the orthogonal projection onto $\text{range}(A)$, we have $AA^\dagger b = b$. Therefore, $Ax = AA^\dagger b$, so $A(x - A^\dagger b) = 0$, implying that $x - A^\dagger b \in \text{null}(A)$. Thus, $x \in A^\dagger b + \text{null}(A)$. Now suppose that $b \in \text{range}(A)$ and $x = A^\dagger b + u$, for some $u \in \text{null}(A)$. Again we have $AA^\dagger b = b$, so $Ax = AA^\dagger b + Au = b$. □

Another well-known property of the Moore-Penrose pseudoinverse is that it gives us a formula for the minimum norm least-squares solution of a linear system.

Proposition 2.7. *Let $\mathbb{E}$ and $\mathbb{Y}$ be Euclidean spaces, $A\colon \mathbb{E} \to \mathbb{Y}$ be linear, $b \in \mathbb{Y}$, and*

$$S := \text{argmin}\left\{\|Ax - b\| : x \in \mathbb{E}\right\}.$$

Then $\bar{x} := A^\dagger b$ is the unique minimizer of

$$\min\left\{\|x\| : x \in S\right\}.$$

Proof. First we will show that $\bar{x} = A^\dagger b \in S$. As is well-known, x minimizes $\|Ax - b\|$ if and only if x satisfies the *normal equations*

$$A^*(b - Ax) = 0. \tag{2.4}$$

To show that $\bar{x}$ satisfies (2.4), we simply observe that

$$b - AA^\dagger b = (I - AA^\dagger)b \in \text{null}(A^*),$$

since $I - AA^\dagger$ is the orthogonal projection onto $\mathrm{null}(A^*)$. To show that $\bar{x}$ is the minimum norm least-squares solution, we begin by letting $x \in S$ and noting that $x = \bar{x} + v$, for some $v \in \mathrm{null}(A^*A)$. First we observe that $Av = 0$; otherwise, we have the contradiction

$$0 \neq \|Av\|^2 = \langle Av, Av \rangle = \langle v, A^*Av \rangle = 0.$$

Note that $\bar{x} = A^\dagger b \in \mathrm{range}(A^\dagger) = \mathrm{range}(A^*)$. Thus, $\bar{x} = A^*y$, for some $y \in \mathbb{Y}$, so we have

$$\begin{aligned} \|x\|^2 &= \|\bar{x} + v\|^2 \\ &= \|\bar{x}\|^2 + 2\langle \bar{x}, v \rangle + \|v\|^2 \\ &= \|\bar{x}\|^2 + 2\langle A^*y, v \rangle + \|v\|^2 \\ &= \|\bar{x}\|^2 + 2\langle y, Av \rangle + \|v\|^2 \\ &= \|\bar{x}\|^2 + 2\langle y, 0 \rangle + \|v\|^2 \\ &= \|\bar{x}\|^2 + \|v\|^2 \\ &\geq \|\bar{x}\|^2. \end{aligned}$$

Therefore, $\|\bar{x}\| \leq \|x\|$ for all $x \in S$, and this inequality is strict if and only if $x \neq \bar{x}$. □

For further information on this topic of generalized inverses, a classic reference is Ben-Israel and Greville (2003).

2.3 Convexity and topology

Let $\mathbb{E}$ be a Euclidean space. A set $A \subseteq \mathbb{E}$ is *affine* if $\lambda x + (1-\lambda)y \in A$, for all $x, y \in A$ and $\lambda \in \mathbb{R}$. A set $S \subseteq \mathbb{E}$ is *convex* if $\lambda x + (1-\lambda)y \in S$, for all $x, y \in S$ and $0 \leq \lambda \leq 1$. A function $f\colon \mathbb{E} \to \mathbb{R}$ is *convex* if $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$, for all $x, y \in \mathbb{E}$ and $0 \leq \lambda \leq 1$.

The *affine hull* of a set $S \subseteq \mathbb{E}$ is the smallest affine set in $\mathbb{E}$ containing S, and is denoted $\mathrm{aff}(S)$; that is,

$$\mathrm{aff}(S) := \bigcap_{\substack{S \subseteq A \subseteq \mathbb{E} \\ A \text{ affine}}} A.$$

The *unit ball* in $\mathbb{E}$ is defined as

$$\mathbb{B} := \{x \in \mathbb{E} : \|x\| \leq 1\}.$$

The ball around $x \in \mathbb{E}$ with radius $\varepsilon > 0$ is then given by

$$x + \varepsilon\mathbb{B} = \{y \in \mathbb{E} : \|x - y\| \leq \varepsilon\}.$$

The *relative interior* of a convex set $S \subseteq \mathbb{E}$ is given by

$$\operatorname{relint}(S) := \{x \in \operatorname{aff}(S) : (x + \varepsilon\mathbb{B}) \cap \operatorname{aff}(S) \subseteq S, \text{ for some } \varepsilon > 0\}.$$

The *core* of a set S is the set of points $x \in S$ set such that you can move in every direction $d \in \mathbb{E}$ slightly and remain in the set. It is known that for convex $S \subseteq \mathbb{E}$ that $\operatorname{core}(S) = \operatorname{int}(S)$; see, for example, Borwein and Lewis (2000). The following well-known theorem states that for convex $S \subseteq \mathbb{E}$ we have $\operatorname{relint}(S) = \operatorname{relcore}(S)$; that is, the relative interior of a convex set $S \subseteq \mathbb{E}$ is characterized by the set of points $x \in S$ such that, for all $y \in S$, you can move in the direction $d = x - y$ slightly and remain in the set.

Theorem 2.8 ((Rockafellar, 1970, Theorem 6.4))**.** *Let $S \subseteq \mathbb{E}$ be a convex set. Then* $\operatorname{relint}(S) = \{x \in S : \forall y \in S, \exists \mu > 1$ *such that* $\mu x + (1 - \mu)y \in S\}$. □

The following is another theorem from the classic book by Rockafellar that we will find useful. This theorem gives us a necessary and sufficient condition for the existence of a nontrivial supporting hyperplane to a set containing a subset of the set.

Theorem 2.9 ((Rockafellar, 1970, Theorem 11.6))**.** *Let $S \subseteq F \subseteq \mathbb{E}$ such that S and F are convex, and $S \neq \emptyset$. Then there exists a supporting hyperplane H to F such that $S \subseteq H$ and $F \nsubseteq H$ if and only if* $S \cap \operatorname{relint}(F) = \emptyset$. □

2.4 Cones and faces

Let $\mathbb{E}$ be a Euclidean space, and let $K \subseteq \mathbb{E}$. We say that K is a *cone* if $\mathbb{R}_+K = K$, where

$$\mathbb{R}_+K := \{tx : t \in \mathbb{R}_+, x \in K\}.$$

The *dual cone* of an arbitrary set $K \subseteq \mathbb{E}$ is defined as

$$K^* := \{y \in \mathbb{E} : \langle x, y\rangle \geq 0, \forall x \in K\}.$$

The dual cone K^* is always a closed convex cone. A cone $K \subseteq \mathbb{E}$ is called *self-dual* if there exists an inner-product $\langle \cdot , \cdot \rangle$ under which $K^* = K$. A well-known result is that $K = K^{**}$ if and only if K is a closed convex cone.

We denote the sets of *positive semidefinite* and *positive definite* matrices by $\mathcal{S}^n_+$ and $\mathcal{S}^n_{++}$, respectively. Thus,

$$\begin{aligned} \mathcal{S}^n_+ &:= \left\{X \in \mathcal{S}^n : v^TXv \geq 0,\ \forall v \in \mathbb{R}^n\right\}, \text{ and} \\ \mathcal{S}^n_{++} &:= \left\{X \in \mathcal{S}^n : v^TXv > 0,\ \forall v \in \mathbb{R}^n \text{ s.t. } v \neq 0\right\}. \end{aligned}$$

We say that $X \succeq Y$ (resp. $X \succ Y$) if $X - Y \in \mathcal{S}^n_+$ (resp. $X - Y \in \mathcal{S}^n_{++}$); this is known as the Löwner partial order. Similarly, we denote the *nonnegative orthant* and the *positive orthant* by

$$\begin{aligned} \mathbb{R}^n_+ &:= \{x \in \mathbb{R}^n : x \geq 0\}, \text{ and} \\ \mathbb{R}^n_{++} &:= \{x \in \mathbb{R}^n : x > 0\}, \end{aligned}$$

respectively. It is well-known that the cones $\mathcal{S}^n_+$ and $\mathbb{R}^n_+$ are self-dual; see, for example, Krislock (2003) for proofs of these results. Classic references on the topic of cones in mathematical optimization are the book Berman (1973) and the preceding paper Berman and Ben-Israel (1971).

2.4.1 Faces of convex cones

If $K \subseteq \mathbb{E}$ is a convex cone, we say that F is a *face* of K Ramana et al. (1997); Rockafellar (1970) (denoted $F \trianglelefteq K$) if F is a convex cone, $F \subseteq K$, and

$$x, y \in K \text{ and } x + y \in F \ \Rightarrow x, y \in F.$$

In addition, if $F \trianglelefteq K$ and $F \neq K$, then we write $F \lhd K$.

For arbitrary subsets F and K of $\mathbb{E}$ such that $F \subseteq K$, we define the *conjugate face* of F (with respect to K) as

$$F^c := F^\perp \cap K^*.$$

We now show that the conjugate face F^c is always a face of the dual cone K^*.

Proposition 2.10. *Let F and K be arbitrary subsets of $\mathbb{E}$. If $F \subseteq K$, then $F^c \trianglelefteq K^*$.*

Proof. Clearly $F^c \subseteq K^*$, and since F^c is the intersection of convex cones, F^c is a convex cone. To show that $F^c \trianglelefteq K^*$, let $x, y \in K^*$ such that $x + y \in F^c$. Thus, if $z \in F \subseteq K$, then $\langle x, z \rangle \geq 0$, $\langle y, z \rangle \geq 0$, and

$$0 = \langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle .$$

Therefore, $\langle x, z \rangle = \langle y, z \rangle = 0$, for all $z \in F$, so $x, y \in F^\perp$. Finally, since $x, y \in K^*$, we have $x, y \in F^c$. Therefore, $F^c \trianglelefteq K^*$. □

A face $F \trianglelefteq K$ is *exposed* if there exists $\phi \in K^*$ such that

$$F = \{x \in K : \langle \phi, x \rangle = 0\} = K \cap \{\phi\}^\perp .$$

A cone K is *facially exposed* if every face $F \trianglelefteq K$ is exposed. The next result shows that each point in the conjugate face F^c defines an exposed face that contains the face F.

Proposition 2.11. *Let $K \subseteq \mathbb{E}$ be a convex cone. If $F \trianglelefteq K$ and $\phi \in F^c$, then $F \trianglelefteq K \cap \{\phi\}^\perp \trianglelefteq K$.*

Proof. First observe that $F \subseteq K$ and $\phi \in F^\perp$ implies that $F \subseteq K \cap \{\phi\}^\perp$. Clearly, $K \cap \{\phi\}^\perp$ is a convex cone. Suppose $x, y \in K \cap \{\phi\}^\perp$ such that $x + y \in F$. Since $x, y \in K$ and $F \trianglelefteq K$, we have $x, y \in F$. Therefore, $F \trianglelefteq K \cap \{\phi\}^\perp$.

Next, we observe that $K \cap \{\phi\}^\perp \subseteq K$, and suppose $x, y \in K$ such that $x + y \in K \cap \{\phi\}^\perp$. Then $\phi \in K^*$ implies that $\langle \phi, x \rangle \geq 0$ and $\langle \phi, y \rangle \geq 0$. Moreover,

$$0 = \langle \phi, x + y \rangle = \langle \phi, x \rangle + \langle \phi, y \rangle ,$$

so $\langle \phi, x \rangle = \langle \phi, y \rangle = 0$. Therefore, $x, y \in K \cap \{\phi\}^\perp$, implying that $K \cap \{\phi\}^\perp \trianglelefteq K$. □

The *minimal face* of a convex cone K containing a set $S \subseteq K$ is denoted face(S); that is,

$$\text{face}(S) := \bigcap_{S \subseteq F \trianglelefteq K} F.$$

Usually we do not specify explicitly the cone K being considered when discussing the minimal face of a set S since it is often clear from the context. The following proposition shows that face(S) is indeed a face of K.

Proposition 2.12. *Let $K \subseteq \mathbb{E}$ be a convex cone. If $S \subseteq K$, then* face$(S) \trianglelefteq K$*.*

Proof. First, we note that face(S) is the intersection of convex cones in K, so face(S) is a convex cone in K. To show that face$(S) \trianglelefteq K$, let $x, y \in K$ such that $x + y \in$ face(S). Let $F \trianglelefteq K$ such that $S \subseteq F$. Then $x + y \in F$, so $x, y \in F$. Therefore, $x, y \in F$, for all $F \trianglelefteq K$ such that $S \subseteq F$, so $x, y \in$ face(S). Thus, face$(S) \trianglelefteq K$. □

Using Theorem 2.8 and Theorem 2.9, we now give a proof for a useful result that provides a simple condition for when face$(S) = F$.

Proposition 2.13. *Let $K \subseteq \mathbb{E}$ be a convex cone and let $S \subseteq F \trianglelefteq K$. Then:*

1. *if $S \cap \operatorname{relint}(F) \neq \emptyset$, then* face$(S) = F$*;*
2. *if S is nonempty and convex, and* face$(S) = F$*, then $S \cap \operatorname{relint}(F) \neq \emptyset$.*

Proof. Since $S \subseteq F \trianglelefteq K$, we have face$(S) \subseteq F$. Conversely, let $x \in F$ and $y \in S \cap \operatorname{relint}(F)$. Then, by Theorem 2.8, $(1 - \mu)x + \mu y \in F$, for some $\mu > 1$. Let $z := (1 - \mu)x + \mu y$. Since $y \in$ face(S), μ is positive, and face(S) is a cone, we have $\mu y \in$ face(S); that is

$$z + (\mu - 1)x \in \text{face}(S).$$

Since $x \in K$, $\mu - 1$ is positive, and K is a cone, we have $(\mu - 1)x \in K$. Since $z, (\mu - 1)x \in K$ and face$(S) \trianglelefteq K$, we have $z, (\mu - 1)x \in$ face(S). Finally, since face(S) is a cone, we have $x \in$ face(S). Thus, $F \subseteq$ face(S).

Now suppose S is convex and $\operatorname{face}(S) = F$. The following argument is inspired by an argument in the proof of (Schneider, 1993, Theorem 2.1.2). Suppose, for the sake of contradiction, that $S \cap \operatorname{relint}(F) = \emptyset$. Then, by Theorem 2.9, there exists a supporting hyperplane H to F containing S, but not containing F. Since F is a cone, we may assume that

$$H = \{x \in \mathbb{E} : \langle \phi, x \rangle = 0\},$$

for some $\phi \in \mathbb{E}$. That is, $\langle \phi, x \rangle = 0$, for all $x \in S$, $\langle \phi, y \rangle \geq 0$, for all $y \in F$, and there exists $\bar{y} \in F$ such that $\langle \phi, \bar{y} \rangle > 0$. As in the proof of Proposition 2.11, we can show that $F \cap H \trianglelefteq K$. Since $S \subseteq F \cap H$, we have that $\operatorname{face}(S) \subseteq F \cap H$. But $\operatorname{face}(S) = F$ then implies that $F \subseteq F \cap H$, contradicting the fact that $F \nsubseteq H$. Therefore, $S \cap \operatorname{relint}(F) \neq \emptyset$. □

2.4.2 A face of the semidefinite cone

We will now give an example of a face of the semidefinite cone, and give a few of its properties. First we prove a useful lemma.

Lemma 2.14. *Let $U \in \mathbb{R}^{n \times t}$ be an arbitrary matrix. Then:*

1. *$Y \in U\mathcal{S}^t_+U^T$ if and only if $Y \succeq 0$ and* $\operatorname{range}(Y) \subseteq \operatorname{range}(U)$*;*
2. *if U is full column rank, then $Y \in U\mathcal{S}^t_{++}U^T$ if and only if $Y \succeq 0$ and* $\operatorname{range}(Y) = \operatorname{range}(U)$*.*

Proof. We prove each result in order.

1. If $Y \in U\mathcal{S}^t_+U^T$, then $Y \succeq 0$ and $\operatorname{range}(Y) \subseteq \operatorname{range}(U)$ clearly holds. Now suppose that $Y \succeq 0$ and $\operatorname{range}(Y) \subseteq \operatorname{range}(U)$. Since $Y \succeq 0$, it is well-known that there exists a unique positive semidefinite square root $Y^{1/2}$ of Y. Moreover, $\operatorname{range}(Y^{1/2}) = \operatorname{range}(Y)$, so there exists $\Phi \in \mathbb{R}^{t \times n}$ such that $Y^{1/2} = U\Phi$. Thus, $Y = UZU^T$, where $Z := \Phi\Phi^T \in \mathcal{S}^t_+$, from which we conclude that $Y \in U\mathcal{S}^t_+U^T$.

2. Suppose that U is full column rank. Let $Y = UZU^T \in U\mathcal{S}^t_{++}U^T$. Then $Y \succeq 0$ and $\operatorname{range}(Y) \subseteq \operatorname{range}(U)$ clearly hold. On the other hand, suppose $v = Ux \in \operatorname{range}(U)$ and let $z \in \mathbb{R}^t$ such that $U^Tz = Z^{-1}x$; such a z exists since $\operatorname{rank}(U^T) = \operatorname{rank}(U) = t$ implies that $\operatorname{range}(U^T) = \mathbb{R}^t$. Therefore, $v = Ux = UZU^Tz = Yz \in \operatorname{range}(Y)$, from which we conclude that $\operatorname{range}(Y) = \operatorname{range}(U)$.

   Now suppose that $Y \succeq 0$ and $\operatorname{range}(Y) = \operatorname{range}(U)$. Then, $U = YV$ for some $V \in \mathbb{R}^{n \times t}$. Moreover, from Item 1 we have that $Y = UZU^T$, for some $Z \in \mathcal{S}^t_+$. Thus, $U = UZU^TV$. Therefore, $\operatorname{rank}(Z) \geq \operatorname{rank}(U) = t$, implying that $Z \in \mathcal{S}^t_{++}$. Thus, $Y \in U\mathcal{S}^t_{++}U^T$. □

Proposition 2.15. *Let $Q = \begin{bmatrix} U & V \end{bmatrix} \in \mathbb{R}^{n\times n}$ be orthogonal, and let*

$$F := U\mathcal{S}_+^t U^T = \left\{ Q \begin{bmatrix} B & 0 \\ 0 & 0 \end{bmatrix} Q^T : B \in \mathcal{S}_+^t \right\}.$$

Then:

1. $F \trianglelefteq \mathcal{S}_+^n$;
2. $F = \left\{ X \in \mathcal{S}_+^n : \operatorname{range}(X) \subseteq \operatorname{range}(U) \right\} = \left\{ X \in \mathcal{S}_+^n : \operatorname{range}(V) \subseteq \operatorname{null}(X) \right\}$;
3. $\operatorname{relint}(F) = U\mathcal{S}_{++}^t U^T = \left\{ X \in \mathcal{S}_+^n : \operatorname{range}(X) = \operatorname{range}(U) \right\}$;
4. $F^c = V\mathcal{S}_+^{n-t} V^T = \left\{ Q \begin{bmatrix} 0 & 0 \\ 0 & B \end{bmatrix} Q^T : B \in \mathcal{S}_+^{n-t} \right\}$;
5. $F^c = \left\{ Y \in \mathcal{S}_+^n : \operatorname{range}(Y) \subseteq \operatorname{range}(V) \right\} = \left\{ Y \in \mathcal{S}_+^n : \operatorname{range}(U) \subseteq \operatorname{null}(Y) \right\}$;
6. $\operatorname{relint}(F^c) = V\mathcal{S}_{++}^{n-t} V^T = \left\{ Y \in \mathcal{S}_+^n : \operatorname{range}(Y) = \operatorname{range}(V) \right\}$;
7. $F \trianglelefteq \mathcal{S}_+^n \cap \{Y\}^\perp$, *for all* $Y \in F^c$; $F = \mathcal{S}_+^n \cap \{Y\}^\perp$, *for all* $Y \in \operatorname{relint}(F^c)$;
8. $F^c \trianglelefteq \mathcal{S}_+^n \cap \{X\}^\perp$ *for all* $X \in F$; $F^c = \mathcal{S}_+^n \cap \{X\}^\perp$, *for all* $X \in \operatorname{relint}(F)$.

Proof. We prove each result in order.

1. Observe that F is a convex cone contained in $\mathcal{S}_+^n$. Now suppose that $X, Y \in \mathcal{S}_+^n$ such that $X + Y \in F$. Then $X + Y = Q \begin{bmatrix} B & 0 \\ 0 & 0 \end{bmatrix} Q^T$, for some $B \in \mathcal{S}_+^t$. Then
$$Q^T X Q + Q^T Y Q = \begin{bmatrix} B & 0 \\ 0 & 0 \end{bmatrix},$$
and $Q^T X Q, Q^T Y Q \succeq 0$ implies that
$$Q^T X Q = \begin{bmatrix} B_x & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad Q^T Y Q = \begin{bmatrix} B_y & 0 \\ 0 & 0 \end{bmatrix},$$
for some $B_x, B_y \in \mathcal{S}_+^t$. Therefore, $X, Y \in F$, so $F \trianglelefteq \mathcal{S}_+^n$.

2. The first equality follows directly from Lemma 2.14. The second equality follows from the fact that $\operatorname{range}(X) \subseteq \operatorname{range}(U)$ if and only if $\operatorname{range}(U)^\perp \subseteq \operatorname{range}(X)^\perp$, and that $\operatorname{range}(U)^\perp = \operatorname{range}(V)$ and $\operatorname{range}(X)^\perp = \operatorname{null}(X)$.

3. Suppose $X = U\Sigma U^T \in \operatorname{relint}(F)$. Let $Y = UU^T$. Then $Y \in F$, so by Theorem 2.8, there exists some $\mu > 1$ such that $(1-\mu)Y + \mu X \in F$. Therefore, there exists $Z \in \mathcal{S}_+^t$ such that
$$UZU^T = (1-\mu)Y + \mu X = U\left[(1-\mu)I + \mu\Sigma\right]U^T.$$

Since $U^TU = I$, we have $Z = (1-\mu)I + \mu\Sigma$; solving for Σ, we get

$$\Sigma = \frac{1}{\mu}Z + \frac{\mu-1}{\mu}I \in \mathcal{S}^n_{++}.$$

Therefore, $X \in U\mathcal{S}^t_{++}U^T$. Since U has full column rank, by Lemma 2.14, we have that $\operatorname{range}(X) = \operatorname{range}(U)$.

Conversely, suppose $X \in \mathcal{S}^n_+$ such that $\operatorname{range}(X) = \operatorname{range}(U)$. Since U has full column rank, by Lemma 2.14, we have that $X = U\Sigma U^T$ for some $\Sigma \in \mathcal{S}^t_{++}$. To show $X \in \operatorname{relint}(F)$, we can show that, for all $Y \in F$, there exists some $\mu > 1$ such that $(1-\mu)Y + \mu X \in F$. To this end, we let $Y = U\Phi U^T \in F$. Since $\mathcal{S}^t_{++} = \operatorname{int}(\mathcal{S}^t_+)$, $\Sigma \in \mathcal{S}^t_{++}$, and $\Phi \in \mathcal{S}^t_+$, we have that there exists some $\mu > 1$ such that $(1-\mu)\Phi + \mu\Sigma \in \mathcal{S}^t_+$. Therefore,

$$(1-\mu)Y + \mu X = U\left[(1-\mu)\Phi + \mu\Sigma\right]U^T \in U\mathcal{S}^t_+U^T = F.$$

Thus, we have $X \in \operatorname{relint}(F)$.

4. We first let $Y = V\Phi V^T \in V\mathcal{S}^{n-t}_+V^T$. To show that $Y \in F^c = F^\perp \cap \mathcal{S}^n_+$, we let $X = U\Sigma U^T \in F$. Observe that $XY = 0$ since Q orthogonal implies that $U^TV = 0$. Thus, $\langle X, Y\rangle = 0$ for all $X \in F$. Since $Y \succeq 0$ clearly holds, we conclude that $Y \in F^c$. To show the other direction, let $Y \in F^c$. Then

$$\left\langle Y, U\Sigma U^T\right\rangle = \left\langle U^TYU, \Sigma\right\rangle = 0, \quad \text{for all } \Sigma \in \mathcal{S}^t_+.$$

In particular, if $\Sigma = I$, then we conclude $\operatorname{trace}(U^TYU) = 0$. Therefore,

$$\|U^TY^{1/2}\|_F^2 = \left\langle U^TY^{1/2}, U^TY^{1/2}\right\rangle = \operatorname{trace}(U^TYU) = 0,$$

implying that $U^TY^{1/2} = 0$. Since

$$\operatorname{null}(U^T) = \operatorname{range}(U)^\perp = \operatorname{range}(V),$$

we have $Y^{1/2} = V\Psi$, for some $\Psi \in \mathbb{R}^{(n-t)\times n}$. Therefore, $Y = V\Phi V^T$, where $\Phi := \Psi\Psi^T \in \mathcal{S}^{n-t}_+$. Thus, $Y \in V\mathcal{S}^{n-t}_+V^T$, as required.

5. Follows as in the proof of item 2.

6. Follows as in the proof of item 3.

7. If $Y \in F^c$, then $F \trianglelefteq \mathcal{S}^n_+ \cap \{Y\}^\perp$ follows from Proposition 2.11. Suppose $Y = V\Phi V^T \in \operatorname{relint}(F^c)$. If $X = U\Sigma U^T \in F$, then $XY = 0$ since $U^TV = 0$. Thus, $X \in \mathcal{S}^n_+ \cap \{Y\}^\perp$. Conversely, suppose $X \in \mathcal{S}^n_+ \cap \{Y\}^\perp$. Then $V\Phi V^TX = YX = 0$. Therefore, $V^TX = 0$ since $V^TV = I$ and $\Phi \succ 0$. Thus, $\operatorname{range}(X) \subseteq \operatorname{null}(V^T) = \operatorname{range}(U)$. Therefore, by Item 2, we have $X \in F$, as required.

8. Follows as in the proof of item 7. □

2.4.3 Semidefinite facial representation theorem

We now present the following well-known representation theorem for faces of the semidefinite cone. Here we show that the example given in Proposition 2.15 gives us the general form of all faces of the semidefinite cone. First we prove a lemma which allows us to represent faces of the semidefinite cone with matrices not having orthogonal columns.

Lemma 2.16. *Let* $U, \bar{U} \in \mathbb{R}^{n\times t}$ *such that* $\operatorname{range}(\bar{U}) = \operatorname{range}(U)$. *Then*

$$U\mathcal{S}^t_+U^T = \bar{U}\mathcal{S}^t_+\bar{U}^T.$$

Proof. From Lemma 2.14, we have $Y \in U\mathcal{S}^t_+U^T$ if and only if $Y \succeq 0$ and $\operatorname{range}(Y) \subseteq \operatorname{range}(U)$. Since $\operatorname{range}(\bar{U}) = \operatorname{range}(U)$, we immediately have $Y \in U\mathcal{S}^t_+U^T$ if and only if $Y \in \bar{U}\mathcal{S}^t_+\bar{U}^T$. □

Theorem 2.17. *Let* $F \trianglelefteq \mathcal{S}^n_+$ *and* $X \in \operatorname{relint}(F)$. *Then*

$$F = U\mathcal{S}^t_+U^T,$$

where $U \in \mathbb{R}^{n\times t}$ *has full column rank and satisfies* $\operatorname{range}(U) = \operatorname{range}(X)$.

Proof. Since $\{X\} \subseteq F \trianglelefteq \mathcal{S}^n_+$ and $\{X\} \cap \operatorname{relint}(F) \neq \emptyset$, Proposition 2.13 implies that $\operatorname{face}\{X\} = F$. So, if we show that $\operatorname{face}\{X\} = U\mathcal{S}^t_+U^T$, we will be done.

Let $\bar{U} \in \mathbb{R}^{n\times t}$ having orthonormal columns and satisfying $\operatorname{range}(\bar{U}) = \operatorname{range}(U)$. By Lemma 2.16, we have $U\mathcal{S}^t_+U^T = \bar{U}\mathcal{S}^t_+\bar{U}^T$; from Proposition 2.15, we know that $\bar{U}\mathcal{S}^t_+\bar{U}^T \trianglelefteq \mathcal{S}^n_+$. Since $\bar{U}$ has full column rank, $X \succeq 0$, and $\operatorname{range}(\bar{U}) = \operatorname{range}(X)$, we have by Lemma 2.14 that $X \in \bar{U}\mathcal{S}^t_{++}\bar{U}^T$. Since $\operatorname{relint}(\bar{U}\mathcal{S}^t_+\bar{U}^T) = \bar{U}\mathcal{S}^t_{++}\bar{U}^T$, we have $\{X\} \cap \operatorname{relint}(\bar{U}\mathcal{S}^t_+\bar{U}^T) \neq \emptyset$. Therefore, by Proposition 2.13, we have

$$\operatorname{face}\{X\} = \bar{U}\mathcal{S}^t_+\bar{U}^T = U\mathcal{S}^t_+U^T.$$ □

Proposition 2.15 and Theorem 2.17 together imply that the semidefinite cone is facially exposed. Early references for this result are (Borwein and Wolkowicz, 1981c, p. 502), (Borwein and Wolkowicz, 1981b, pp. 374–5), and Borwein and Wolkowicz (1981a). Other sources for the above semidefinite facial results are, for example, Ramana et al. (1997), and the earlier papers Barker (1973), and Barker and Carlson (1975).

2.4.4 A theorem of the alternative for semidefinite optimization and the minimal face

The following result is an extension of a result in Cheung et al. (2010), which itself is specialization of the result (Borwein and Wolkowicz, 1981c, Theorem 7.1).

Theorem 2.18. *Exactly one of the two following systems is consistent:*

1. $0 \neq X \succeq 0$, $\mathcal{A}X = 0$, $\langle C, X\rangle \leq 0$;
2. $\mathcal{A}^* y \prec C$.

Furthermore, if there exists $X \in \mathcal{S}^n_+$ such that $\mathcal{A}X = 0$ and $\langle C, X\rangle < 0$, then the system $\mathcal{A}^* y \preceq C$ is also infeasible.

Proof. Suppose that there exists $0 \neq X \succeq 0$ such that $\mathcal{A}X = 0$ and $\langle C, X\rangle \leq 0$. For the sake of contradiction, suppose that there exists $y \in \mathbb{R}^m$ such that $\mathcal{A}^* y \prec C$. Then

$$\begin{aligned} 0 &< \langle C - \mathcal{A}^* y, X\rangle \\ &= \langle C, X\rangle - \langle y, \mathcal{A}X\rangle \\ &= \langle C, X\rangle \\ &\leq 0, \end{aligned}$$

which is a contradiction, so $\nexists y \in \mathbb{R}^m$ such that $\mathcal{A}^* y \prec C$.

Suppose that $C - \mathcal{A}^* y \notin \mathcal{S}^n_{++}$ for all $y \in \mathbb{R}^m$. The following hyperplane separation argument was inspired by the proof in Cheung et al. (2010), but here we take a different approach. First we observe that

$$C \notin \mathcal{S}^n_{++} + \mathcal{A}^*(\mathbb{R}^m).$$

Since $\mathcal{S}^n_{++} + \mathcal{A}^*(\mathbb{R}^m)$ is a convex cone, by weak hyperplane separation we have that there exists $X \neq 0$ and $\beta \in \mathbb{R}$ such that

$$\langle C, X\rangle \leq \beta \leq \langle S + \mathcal{A}^* y, X\rangle, \quad \text{for all } S \in \mathcal{S}^n_{++},\ y \in \mathbb{R}^m.$$

Taking $y = 0$, we see that $X \succeq 0$; otherwise there exists $v \neq 0$ such that $v^T X v < 0$, giving us the contradiction

$$\beta \leq \left\langle tvv^T + I, X\right\rangle = tv^T Xv + \langle I, X\rangle \to -\infty \quad \text{as } t \to \infty.$$

Taking $S = \frac{1}{t} I$ and $y = -t\mathcal{A}X$ for $t > 0$, we see that $\mathcal{A}X = 0$; otherwise we have the contradiction

$$\beta \leq \left\langle \frac{1}{t} I + \mathcal{A}^*(-t\mathcal{A}X), X\right\rangle = \frac{1}{t}\langle I, X\rangle - t\langle \mathcal{A}X, \mathcal{A}X\rangle \to -\infty \quad \text{as } t \to \infty.$$

Taking $S = \frac{1}{t} I$ for $t > 0$, and $y = 0$, we have that $\langle C, X\rangle \leq 0$, since

$$\langle C, X\rangle \leq \beta \leq \left\langle \frac{1}{t} I, X\right\rangle \searrow 0 \quad \text{as } t \to \infty.$$

Therefore, there exists $0 \neq X \succeq 0$ such that $\mathcal{A}X = 0$ and $\langle C, X\rangle \leq 0$.

Now suppose there exists an X such that $X \succeq 0$, $\mathcal{A}X = 0$, $\langle C, X\rangle < 0$. For the sake of contradiction, suppose there exists $\hat{y}$ such that $\mathcal{A}^*\hat{y} \preceq C$; let $\hat{S} := C - \mathcal{A}^*\hat{y} \succeq 0$. Then

$$0 > \langle C, X\rangle = \left\langle \hat{S} + \mathcal{A}^*\hat{y}, X \right\rangle = \left\langle \hat{S}, X \right\rangle \geq 0,$$

giving us the required contradiction. □

The above Theorem of the Alternative, Theorem 2.18, now inspires us to give the following partial description of the conjugate face of the minimal face of a feasible set of a semidefinite optimization problem.

Proposition 2.19. *Let* $F_D := \mathrm{face}(\mathcal{F}_D)$*, where*

$$\mathcal{F}_D := \left\{ S \in \mathcal{S}^n_+ : S = C - \mathcal{A}^* y, \ \textit{for some } y \in \mathbb{R}^m \right\}.$$

If $\mathcal{F}_D \neq \emptyset$*, then* $\mathrm{face}\left\{X \in \mathcal{S}^n_+ : \mathcal{A}X = 0, \langle C, X\rangle = 0\right\} \trianglelefteq F_D^c$.

Proof. Let $X \in \mathcal{S}^n_+$ such that $\mathcal{A}X = 0$ and $\langle C, X\rangle = 0$. If $S = C - \mathcal{A}^* y \in \mathcal{F}_D$, then

$$\langle S, X\rangle = \langle C - \mathcal{A}^* y, X\rangle = \langle C, X\rangle - \langle y, \mathcal{A}X\rangle = \langle C, X\rangle = 0.$$

Therefore, $\langle S, X\rangle = 0$, for all $S \in \mathcal{F}_D$, so $X \in \mathcal{F}_D^\perp$. Since $\mathcal{F}_D \neq \emptyset$, there exists some $\bar{S} \in \mathrm{relint}(\mathcal{F}_D)$. Let $\bar{S}$ have compact eigenvalue decomposition $\bar{S} = U\Lambda U^T$ with $U \in \mathbb{R}^{n\times t}$ having orthonormal columns and diagonal $\Lambda \in \mathcal{S}^t_{++}$. Then $F_D = U\mathcal{S}^t_+ U^T$. Moreover,

$$\begin{aligned} 0 = \bar{S}X = U\Lambda U^T X \quad &\Rightarrow \quad \Lambda U^T X = 0 \\ &\Rightarrow \quad U^T X = 0. \end{aligned}$$

This implies that $X \in F_D^\perp$, so $X \in F_D^c$. Therefore,

$$\left\{X \in \mathcal{S}^n_+ : \mathcal{A}X = 0, \langle C, X\rangle = 0\right\} \subseteq F_D^c,$$

so we have that

$$\mathrm{face}\left\{X \in \mathcal{S}^n_+ : \mathcal{A}X = 0, \langle C, X\rangle = 0\right\} \trianglelefteq F_D^c.$$ □

Since the conjugate face F_D^c can be viewed as a collection of exposed faces containing the minimal face F_D, Proposition 2.19 gives us the following useful corollary.

Corollary 2.20. *Let* $F_D := \mathrm{face}(\mathcal{F}_D)$*, where*

$$\mathcal{F}_D := \left\{ S \in \mathcal{S}^n_+ : S = C - \mathcal{A}^* y, \ \textit{for some } y \in \mathbb{R}^m \right\}.$$

If $\mathcal{F}_D \neq \emptyset$ *and* $X \in \mathcal{S}^n_+$ *satisfies* $\mathcal{A}X = 0$ *and* $\langle C, X\rangle = 0$*, then*

$$F_D \trianglelefteq \mathcal{S}^n_+ \cap \{X\}^\perp = U\mathcal{S}^t_+ U^T,$$

where $U \in \mathbb{R}^{n\times t}$ *has full column rank and satisfies* $\mathrm{range}(U) = \mathrm{null}(X)$.

Proof. By Proposition 2.19, $X \in F_D^c$. By Proposition 2.11, $F_D \trianglelefteq \mathcal{S}^n_+ \cap \{X\}^\perp$. By Proposition 2.15 and Lemma 2.16, $\mathcal{S}^n_+ \cap \{X\}^\perp \ = \ U\mathcal{S}^{n-t}_+ U^T$. □

Dual results

We now give the dual results of the above the results; the proofs proceed in an identical manner.

Theorem 2.21. *Exactly one of the two following systems is consistent:*

1. $0 \neq \mathcal{A}^* y \succeq 0$, $b^T y \leq 0$;
2. $\mathcal{A}X = b$, $X \succ 0$.

Furthermore, if there exists $y \in \mathbb{R}^m$ such that $\mathcal{A}^* y \succeq 0$ and $b^T y < 0$, then the system $(\mathcal{A}X = b,\ X \succeq 0)$ is also infeasible. □

Proposition 2.22. *Let $F_P := \operatorname{face}(\mathcal{F}_P)$, where*

$$\mathcal{F}_P := \left\{ X \in \mathcal{S}^n_+ : \mathcal{A}X = b \right\}.$$

If $\mathcal{F}_P \neq \emptyset$, then $\operatorname{face}\left\{ S \in \mathcal{S}^n_+ : S = \mathcal{A}^* y, b^T y = 0 \right\} \trianglelefteq F_P^c$. □

Corollary 2.23. *Let $F_P := \operatorname{face}(\mathcal{F}_P)$, where*

$$\mathcal{F}_P := \left\{ X \in \mathcal{S}^n_+ : \mathcal{A}X = b \right\}.$$

If $\mathcal{F}_P \neq \emptyset$ and $S \in \mathcal{S}^n_+$ satisfies $S = \mathcal{A}^* y$ for some $y \in \mathbb{R}^m$ such that $b^T y = 0$, then

$$F_P \trianglelefteq \mathcal{S}^n_+ \cap \{S\}^\perp = U \mathcal{S}^t_+ U^T,$$

where $U \in \mathbb{R}^{n \times t}$ has full column rank and satisfies $\operatorname{range}(U) = \operatorname{null}(S)$. □

2.5 Euclidean distance matrices

A *Euclidean distance matrix* (EDM) is a matrix $D \in \mathcal{S}^n$ for which there exists a set of points $p_1, \ldots, p_n \in \mathbb{R}^r$ such that

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

The smallest integer r for which this is possible is called the *embedding dimension* of D, and is denoted $\operatorname{embdim}(D)$. Thus,

$$\operatorname{embdim}(D) := \min \left\{ r : \exists p_1, \ldots, p_n \in \mathbb{R}^r \text{ s.t. } D_{ij} = \|p_i - p_j\|_2^2, \text{ for all } i, j \right\}.$$

The set of Euclidean distance matrices is denoted $\mathcal{E}^n$.

There is a natural relationship between semidefinite matrices and Euclidean distance matrices. Let D be a Euclidean distance matrix that is realized by the points $p_1, \ldots, p_n \in \mathbb{R}^r$. Let

$$P := \begin{bmatrix} p_1^T \\ \vdots \\ p_n^T \end{bmatrix} \quad \text{and} \quad Y := PP^T = \left(p_i^T p_j\right)_{i,j=1}^n.$$

The matrix $Y = PP^T$ is known as the *Gram matrix* of the points $p_1, \ldots, p_n \in \mathbb{R}^r$. Then

$$\begin{aligned} D_{ij} &= \|p_i - p_j\|_2^2 \\ &= p_i^T p_i + p_j^T p_j - 2p_i^T p_j \\ &= Y_{ii} + Y_{jj} - 2Y_{ij}. \end{aligned}$$

Therefore, $D = \mathcal{K}(Y)$, where $\mathcal{K} \colon \mathcal{S}^n \to \mathcal{S}^n$ is the linear map[1] defined as

$$\mathcal{K}(Y)_{ij} := Y_{ii} + Y_{jj} - 2Y_{ij}, \quad \text{for } i, j = 1, \ldots, n.$$

Equivalently, we can define $\mathcal{K}$ by

$$\mathcal{K}(Y) := \operatorname{diag}(Y)e^T + e\operatorname{diag}(Y)^T - 2Y, \tag{2.5}$$

where $e \in \mathbb{R}^n$ is the vector of all ones. From this simple observation, we can see that $\mathcal{K}$ maps the cone of semidefinite matrices, $\mathcal{S}_+^n$, onto $\mathcal{E}^n$. That is, $\mathcal{K}(\mathcal{S}_+^n) = \mathcal{E}^n$. In addition, since $\mathcal{S}_+^n$ is a convex cone, we immediately get that $\mathcal{E}^n$ is a convex cone.

2.5.1 Properties of $\mathcal{K}$

We will now discuss various useful properties of the linear map $\mathcal{K}$. At the end of this section, on page 33, we provide a useful summary of the properties of $\mathcal{K}$ discussed here.

The hollow matrices and the range space of $\mathcal{K}$

One of the first properties one observes about a Euclidean distance matrix D is that its diagonal must be zero. We call matrices with zero diagonal *hollow*. The subspace of $\mathcal{S}^n$ of hollow matrices is denoted $\mathcal{S}_H^n$. That is,

$$\mathcal{S}_H^n := \left\{ D \in \mathcal{S}^n : \operatorname{diag}(D) = 0 \right\}.$$

Proposition 2.24 ((Al-Homidan and Wolkowicz, 2005, Prop. 2.2, Equation (2.8), Item 1))**.** *The range space of $\mathcal{K} \colon \mathcal{S}^n \to \mathcal{S}^n$ is given by*

$$\operatorname{range}(\mathcal{K}) = \mathcal{S}_H^n.$$

Proof. Let $Y \in \mathcal{S}^n$. Then $\mathcal{K}(Y)_{ii} = Y_{ii} + Y_{ii} - 2Y_{ii} = 0$, for all i, so $\mathcal{K}(Y) \in \mathcal{S}_H^n$. Moreover, if $D \in \mathcal{S}_H^n$ and $Y := -\frac{1}{2}D$, then $\mathcal{K}(Y) = D$. □

[1] Early appearances of this linear map are in Schoenberg (1935) and Young and Householder (1938). However, the use of the notation $\mathcal{K}$ for this linear map dates back to Critchley (1988) wherein κ was used due to the fact that the formula for $\mathcal{K}$ is basically the *cosine law* ($c^2 = a^2 + b^2 - 2ab\cos(\gamma)$). Later on, in Johnson and Tarazaga (1995), K was used to denote this linear map.

Translational invariance and the null space of $\mathcal{K}$

The null space of $\mathcal{K}$ is closely related to the translational invariance of distances between a set of points. Suppose that $P \in \mathbb{R}^{n\times r}$ and that

$$\hat{P} = P + ev^T, \quad \text{for some } v \in \mathbb{R}^r;$$

that is, $\hat{P}$ is the matrix formed by translating every row of P by the vector v. Clearly, P and $\hat{P}$ generate the same Euclidean distance matrix, so we have $\mathcal{K}(PP^T) = \mathcal{K}(\hat{P}\hat{P}^T)$. Note that

$$\begin{array}{rcl} \hat{P}\hat{P}^T & = & PP^T + Pve^T + ev^TP + ev^Tve^T \\ & = & PP^T + \mathcal{D}_e(y), \end{array}$$

where $y := Pv + \frac{v^Tv}{2}e$, and $\mathcal{D}_e\colon \mathbb{R}^n \to \mathcal{S}^n$ is the linear map defined as

$$\mathcal{D}_e(y) := ye^T + ey^T. \tag{2.6}$$

Thus, we have

$$0 = \mathcal{K}(\hat{P}\hat{P}^T - PP^T) = \mathcal{K}(\mathcal{D}_e(y)),$$

so $\mathcal{D}_e(y) \in \operatorname{null}(\mathcal{K})$. Below is the general result that fully describes the null space of $\mathcal{K}$.

Proposition 2.25 ((Al-Homidan and Wolkowicz, 2005, Prop. 2.2, Equation (2.8), Item 2))**.** *The null space of $\mathcal{K}$ is given by*

$$\operatorname{null}(\mathcal{K}) = \operatorname{range}(\mathcal{D}_e).$$

Proof. Note that $\mathcal{K}(Y)_{ij} = Y_{ii} + Y_{jj} - 2Y_{ij} = 0$ if and only if $Y_{ij} = \frac{1}{2}(Y_{ii} + Y_{jj})$. Therefore, $\mathcal{K}(Y) = 0$ if and only if $Y = ye^T + ey^T = \mathcal{D}_e(y)$, where $y := \frac{1}{2}\operatorname{diag}(Y)$. □

Rotational invariance of the Gram matrix

Suppose that $P \in \mathbb{R}^{n\times r}$ and that

$$\hat{P} = PQ, \quad \text{for some } Q \in \mathbb{R}^{r\times r} \text{ orthogonal;}$$

that is, $\hat{P}$ is the matrix formed by rotating/reflecting each row of P by the same orthogonal transformation. Again, we clearly have that P and $\hat{P}$ generate the same Euclidean distance matrix, but we can say more. If Y is the Gram matrix of P and $\hat{Y}$ is the Gram matrix of $\hat{P}$, then

$$\hat{Y} = \hat{P}\hat{P}^T = PQQ^TP^T = PP^T = Y.$$

Therefore, we have that the Gram matrix is invariant under orthogonal transformations of the points. Thus, when using a semidefinite matrix Y to represent a Euclidean distance matrix D with $D = \mathcal{K}(Y)$, we must only concern ourselves with translations of the points, since orthogonal transformations will not affect Y.

The adjoint of $\mathcal{K}$ and the centred matrices

We begin by computing the adjoint of $\mathcal{D}_e$.

Proposition 2.26. *The adjoint of $\mathcal{D}_e \colon \mathbb{R}^n \to \mathcal{S}^n$ is the function $\mathcal{D}_e^* \colon \mathcal{S}^n \to \mathbb{R}^n$ given by*

$$\mathcal{D}_e^*(D) = 2De.$$

Proof. Let $y \in \mathbb{R}^n$ and $D \in \mathcal{S}^n$. Then

$$\begin{aligned} \langle \mathcal{D}_e(y), D \rangle &= \left\langle ye^T + ey^T, D \right\rangle \\ &= \sum_{i,j} (y_i + y_j) D_{ij} \\ &= 2 \sum_i y_i \sum_j D_{ij} \\ &= y^T (2De), \end{aligned}$$

so $\mathcal{D}_e^*(D) = 2De$. □

Using the adjoint of $\mathcal{D}_e$, we now compute the adjoint of

$$\mathcal{K}(Y) = \operatorname{diag}(Y)e^T + e\operatorname{diag}(Y)^T - 2Y = \mathcal{D}_e(\operatorname{diag}(Y)) - 2Y.$$

Proposition 2.27. *The adjoint of $\mathcal{K} \colon \mathcal{S}^n \to \mathcal{S}^n$ is the function $\mathcal{K}^* \colon \mathcal{S}^n \to \mathcal{S}^n$ given by*

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

Proof. Let $Y, D \in \mathcal{S}^n$. Then

$$\begin{aligned} \langle \mathcal{K}(Y), D \rangle &= \langle \mathcal{D}_e(\operatorname{diag}(Y)) - 2Y, D \rangle \\ &= \langle Y, \operatorname{diag}^*(\mathcal{D}_e^*(D)) \rangle - 2 \langle Y, D \rangle \\ &= \langle Y, \operatorname{Diag}(2De) - 2D \rangle \\ &= \langle Y, 2\left(\operatorname{Diag}(De) - D\right) \rangle, \end{aligned}$$

so $\mathcal{K}^*(D) = 2\left(\operatorname{Diag}(De) - D\right)$. □

A matrix $Y \in \mathcal{S}^n$ is called *centred* if it has zero row sums. The subspace of $\mathcal{S}^n$ of centred matrices is denoted $\mathcal{S}_C^n$. That is,

$$\mathcal{S}_C^n := \left\{ Y \in \mathcal{S}^n : Ye = 0 \right\}.$$

Proposition 2.28 ((Al-Homidan and Wolkowicz, 2005, Prop. 2.2, Equation (2.9)))**.** *The range space and null space of $\mathcal{K}^*$ are given by*

$$\operatorname{range}(\mathcal{K}^*) = \operatorname{range}(\mathcal{D}_e)^\perp = \mathcal{S}_C^n \quad \textit{and} \quad \operatorname{null}(\mathcal{K}^*) = {\mathcal{S}_H^n}^\perp = \operatorname{range}(\operatorname{Diag}).$$

Proof. First of all, by Proposition 2.25, we have

$$\operatorname{range}(\mathcal{K}^*) = \operatorname{null}(\mathcal{K})^\perp = \operatorname{range}(\mathcal{D}_e)^\perp.$$

Next we note that for $y \in \mathbb{R}^n$ and $Y \in \mathcal{S}^n$, we have

$$\langle \mathcal{D}_e(y), Y \rangle = \langle y, \mathcal{D}_e^*(Y) \rangle = \langle y, 2Ye \rangle .$$

Therefore, $\langle \mathcal{D}_e(y), Y \rangle = 0$ for all $y \in \mathbb{R}^n$ if and only if $Ye = 0$. By Proposition 2.24, we have

$$\operatorname{null}(\mathcal{K}^*) = \operatorname{range}(\mathcal{K})^\perp = \mathcal{S}_H^{n\,\perp}.$$

Clearly $\mathcal{S}_H^{n\,\perp}$ consists of the diagonal matrices, which can be denoted as range(Diag). □

Orthogonal projectors onto the hollow and centred subspaces

Proposition 2.29. *The orthogonal projector onto the subspace of hollow matrices, $\mathcal{S}_H^n$, is the linear map* $\operatorname{offDiag}\colon \mathcal{S}^n \to \mathcal{S}^n$ *defined by*

$$\operatorname{offDiag}(D)_{ij} := \begin{cases} 0 & \text{if } i = j \\ D_{ij} & \text{if } i \neq j. \end{cases}$$

In other words, $\operatorname{offDiag}(D) = D - \operatorname{Diag}(\operatorname{diag}(D))$.

Proof. It is straightforward to check the three sufficient conditions for offDiag to be the orthogonal projector onto $\mathcal{S}_H^n$: $\operatorname{offDiag}^2 = \operatorname{offDiag}$, $\operatorname{offDiag}^* = \operatorname{offDiag}$, and $\operatorname{range}(\operatorname{offDiag}) = \mathcal{S}_H^n$. □

Proposition 2.30. *The orthogonal projector onto the subspace of hollow matrices, $\mathcal{S}_C^n$, is the linear map $J(\cdot)J\colon \mathcal{S}^n \to \mathcal{S}^n$, where*

$$J := I - \frac{1}{n}ee^T.$$

Proof. Since $\frac{1}{n}ee^T\colon \mathbb{R}^n \to \mathbb{R}^n$ is the orthogonal projector onto the subspace span $\{e\}$, $J\colon \mathbb{R}^n \to \mathbb{R}^n$ is the orthogonal projector onto the subspace $\{e\}^\perp$. Therefore, we have that $J(JYJ)J = JYJ$ and $\langle JYJ, X\rangle = \langle Y, JXJ\rangle$, for all $X, Y \in \mathcal{S}^n$. Moreover, $Y \in \mathcal{S}_C^n$ if and only if $Y = JYJ$, so $\operatorname{range}(J(\cdot)J) = \mathcal{S}_C^n$. □

The Moore-Penrose pseudoinverse of $\mathcal{K}$

Let $\mathcal{T}\colon \mathcal{S}^n \to \mathcal{S}^n$ be defined by

$$\mathcal{T}(D) := -\frac{1}{2}J\operatorname{offDiag}(D)J, \quad \text{where } J = I - \frac{1}{n}ee^T. \tag{2.7}$$

We will show that $\mathcal{T}$ is the Moore-Penrose pseudoinverse of $\mathcal{K}$.

Theorem 2.31 ((Al-Homidan and Wolkowicz, 2005, Prop. 2.2, Equation (2.7)))**.** *The linear map $\mathcal{T} \colon \mathcal{S}^n \to \mathcal{S}^n$ defined in equation* (2.7) *is the Moore-Penrose pseudoinverse of $\mathcal{K} \colon \mathcal{S}^n \to \mathcal{S}^n$. That is, $\mathcal{K}^\dagger = \mathcal{T}$, so $\mathcal{K}^\dagger \colon \mathcal{S}^n \to \mathcal{S}^n$ is given by*

$$\mathcal{K}^\dagger(D) = -\frac{1}{2} J \operatorname{offDiag}(D) J, \quad \textit{where } J = I - \frac{1}{n} ee^T. \tag{2.8}$$

Before giving a proof of Theorem 2.31, we present two lemmas which give us expressions for $\mathcal{K}\mathcal{T}$ and $\mathcal{T}\mathcal{K}$.

Lemma 2.32. *Let $\mathcal{T} \colon \mathcal{S}^n \to \mathcal{S}^n$ be defined by equation* (2.7)*. Then $\mathcal{K}\mathcal{T} \colon \mathcal{S}^n \to \mathcal{S}^n$ is the orthogonal projector onto the hollow matrices $\mathcal{S}^n_H$; that is,*

$$\mathcal{K}\mathcal{T}(D) = \operatorname{offDiag}(D).$$

Proof. First we note that if $M \in \mathcal{S}^n$, then

$$\begin{aligned} JMJ &= M - \frac{1}{n}\left(Mee^T + ee^T M + ee^T\right) \\ &= M + \mathcal{D}_e(y), \end{aligned}$$

where $y := -\frac{1}{n}\left(Me + \frac{1}{2}e\right)$. Since $\mathcal{K}(\mathcal{D}_e(y)) = 0$, we have

$$\mathcal{K}(JMJ) = \mathcal{K}(M), \quad \text{for all } M \in \mathcal{S}^n.$$

Therefore,

$$\begin{aligned} \mathcal{K}\mathcal{T}(D) &= -\frac{1}{2}\mathcal{K}(J\operatorname{offDiag}(D)J) \\ &= -\frac{1}{2}\mathcal{K}(\operatorname{offDiag}(D)) \\ &= \operatorname{offDiag}(D), \end{aligned}$$

since $\mathcal{K}(\operatorname{offDiag}(D)) = -2\operatorname{offDiag}(D)$. □

Lemma 2.33. *Let $\mathcal{T} \colon \mathcal{S}^n \to \mathcal{S}^n$ be defined by equation* (2.7)*. Then $\mathcal{T}\mathcal{K} \colon \mathcal{S}^n \to \mathcal{S}^n$ is the orthogonal projector onto the centred matrices $\mathcal{S}^n_C$; that is,*

$$\mathcal{T}\mathcal{K}(Y) = JYJ, \quad \textit{where } J = I - \frac{1}{n}ee^T.$$

Proof. Since $\operatorname{range}(\mathcal{K}) = \mathcal{S}^n_H$, we have

$$\operatorname{offDiag}(\mathcal{K}(Y)) = \mathcal{K}(Y), \quad \text{for all } Y \in \mathcal{S}^n.$$

Next we note that

$$J\left[\mathcal{K}(Y) + 2Y\right]J = J\left[\mathcal{D}_e(\operatorname{diag}(Y))\right]J = 0, \quad \text{for all } Y \in \mathcal{S}^n,$$

since $J(\cdot)J$ is the orthogonal projection onto the subspace $\mathcal{S}^n_C$, and $\operatorname{range}(D_e) = \mathcal{S}^{n\perp}_C$. Therefore,

$$\begin{aligned}\mathcal{T}\mathcal{K}(Y) &= -\frac{1}{2}J\operatorname{offDiag}(\mathcal{K}(Y))J \\ &= -\frac{1}{2}J\mathcal{K}(Y)J \\ &= -\frac{1}{2}(-2JYJ) \\ &= JYJ. \end{aligned} \qquad \square$$

Proof of Theorem 2.31. By Proposition 2.3, we simply must show that $\mathcal{T}$ satisfies the Penrose conditions (2.1).

1. By Lemma 2.32, $\mathcal{K}\mathcal{T}$ is the orthogonal projector onto the subspace $\mathcal{S}^n_H$. Thus, Penrose condition (2.1a) holds.

2. By Lemma 2.33, $\mathcal{T}\mathcal{K}$ is the orthogonal projector onto the subspace $\mathcal{S}^n_C$. Thus, Penrose condition (2.1b) holds.

3. Let $Y \in \mathcal{S}^n$. Then

$$\mathcal{K}\mathcal{T}\mathcal{K}(Y) = \mathcal{K}(JYJ) = \mathcal{K}(Y + \mathcal{D}_e(y)) = \mathcal{K}(Y),$$

   where $y := -\frac{1}{n}\left(Ye + \frac{1}{2}e\right)$. Therefore, Penrose condition (2.1c) holds.

4. Let $D \in \mathcal{S}^n$. Then

$$\mathcal{T}\mathcal{K}\mathcal{T}(D) = \mathcal{T}(\operatorname{offDiag}(D)) = \mathcal{T}(D),$$

   since $\operatorname{offDiag}^2 = \operatorname{offDiag}$. Therefore, Penrose condition (2.1d) holds.

Therefore, $\mathcal{T} = \mathcal{K}^\dagger$. □

We now give descriptions of the four fundamental subspaces associated with $\mathcal{K}^\dagger$. Note that $\mathcal{K}^{\dagger *}\colon \mathcal{S}^n \to \mathcal{S}^n$ is given by

$$\mathcal{K}^{\dagger *}(Y) = -\frac{1}{2}\operatorname{offDiag}(JYJ).$$

Proposition 2.34. *The range space and null space of the linear maps $\mathcal{K}^\dagger$ and $\mathcal{K}^{\dagger *}$ are given by*

$$\operatorname{range}(\mathcal{K}^\dagger) = \mathcal{S}^n_C, \qquad \operatorname{null}(\mathcal{K}^{\dagger *}) = \operatorname{range}(\mathcal{D}_e), \tag{2.9}$$

$$\operatorname{range}(\mathcal{K}^{\dagger *}) = \mathcal{S}^n_H, \qquad \operatorname{null}(\mathcal{K}^\dagger) = \operatorname{range}(\operatorname{Diag}). \tag{2.10}$$

Proof. Result follows from Proposition 2.4, since $\operatorname{range}(\mathcal{K}^*) = \mathcal{S}^n_C$, $\operatorname{null}(\mathcal{K}^*) = \operatorname{range}(\operatorname{Diag})$, $\operatorname{null}(\mathcal{K}) = \operatorname{range}(\mathcal{D}_e)$, and $\operatorname{range}(\mathcal{K}) = \mathcal{S}^n_H$. □

The maps $\mathcal{K}$ and $\mathcal{K}^\dagger$ as bijections

First we consider $\mathcal{K}$ and $\mathcal{K}^\dagger$ restricted to the subspaces $\mathcal{S}^n_C$ and $\mathcal{S}^n_H$, respectively.

Proposition 2.35 ((Al-Homidan and Wolkowicz, 2005, Theorem 2.1))**.** *The map $\mathcal{K}\colon \mathcal{S}^n_C \to \mathcal{S}^n_H$ is a bijection and $\mathcal{K}^\dagger\colon \mathcal{S}^n_H \to \mathcal{S}^n_C$ is its inverse.*

Proof. Result follows from Proposition 2.5 and the facts that $\text{range}(\mathcal{K}) = \mathcal{S}^n_H$ and $\text{range}(\mathcal{K}^\dagger) = \mathcal{S}^n_C$. □

For the main result, we consider $\mathcal{K}$ and $\mathcal{K}^\dagger$ restricted to the convex cones $\mathcal{S}^n_+ \cap \mathcal{S}^n_C$ and $\mathcal{E}^n$, respectively.

Theorem 2.36 ((Al-Homidan and Wolkowicz, 2005, Theorem 2.3))**.** *The map $\mathcal{K}\colon \mathcal{S}^n_+ \cap \mathcal{S}^n_C \to \mathcal{E}^n$ is a bijection and $\mathcal{K}^\dagger\colon \mathcal{E}^n \to \mathcal{S}^n_+ \cap \mathcal{S}^n_C$ is its inverse.*

Proof. As we have seen $\mathcal{K}(\mathcal{S}^n_+) = \mathcal{E}^n$. Therefore, if $D \in \mathcal{E}^n$, there exists $Y \in \mathcal{S}^n_+$ such that $\mathcal{K}(Y) = D$. Let $Y = X + Z \in \mathcal{S}^n_C \oplus \text{range}(\mathcal{D}_e) = \mathcal{S}^n$. Then applying the orthogonal projection onto the subspace $\mathcal{S}^n_C$, we have $JYJ = JXJ + JZJ = X$, so $X \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C$. Now, since

$$D = \mathcal{K}(Y) = \mathcal{K}(X + Z) = \mathcal{K}(X),$$

we see that $\mathcal{K}\colon \mathcal{S}^n_+ \cap \mathcal{S}^n_C \to \mathcal{E}^n$ is surjective. Moreover, Proposition 2.35 implies that $\mathcal{K}\colon \mathcal{S}^n_+ \cap \mathcal{S}^n_C \to \mathcal{E}^n$ is also injective, and that $\mathcal{K}^\dagger\colon \mathcal{E}^n \to \mathcal{S}^n_+ \cap \mathcal{S}^n_C$ is its inverse. □

Embedding dimension and a theorem of Schoenberg

We now give the following well-known result for determining the embedding dimension of a Euclidean distance matrix; see Alfakih et al. (1999).

Theorem 2.37. *Let $D \in \mathcal{E}^n$. Then*

$$\text{embdim}(D) = \text{rank}\mathcal{K}^\dagger(D) \leq n - 1.$$

Proof. Let $r := \text{rank}\mathcal{K}^\dagger(D)$. Since $D \in \mathcal{E}^n$, by Theorem 2.36, we have $\mathcal{K}^\dagger(D) \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C$. Let $\mathcal{K}^\dagger(D) = U\Lambda U^T$ be the compact eigenvalue decomposition of $\mathcal{K}^\dagger(D)$ with $U \in \mathbb{R}^{n\times r}$ having orthonormal columns and diagonal $\Lambda \in \mathcal{S}^r_{++}$. Let

$$P := U\Lambda^{1/2}.$$

Then $\mathcal{K}^\dagger(D) = PP^T$, and $D = \mathcal{K}(PP^T)$. Therefore, $\text{embdim}(D) \leq r$. Now suppose, for the sake of contradiction, that $\bar{r} := \text{embdim}(D) < r$. Then there exists $\bar{P} \in \mathbb{R}^{n\times\bar{r}}$ such that $\mathcal{K}(\bar{P}\bar{P}^T) = D$. Applying $\mathcal{K}^\dagger$ to both sides gives us $\mathcal{K}^\dagger\mathcal{K}(\bar{P}\bar{P}^T) = \mathcal{K}^\dagger(D)$. Since $\mathcal{K}^\dagger\mathcal{K}$ is the orthogonal projection onto the subspace

$\mathcal{S}^n_C$, we have $J\bar{P}\bar{P}^TJ = \mathcal{K}^\dagger(D)$. But $J\bar{P} \in \mathbb{R}^{n\times\bar{r}}$ implies that $\text{rank}\mathcal{K}^\dagger(D) \leq \bar{r}$, giving us the contradiction

$$r = \text{rank}\mathcal{K}^\dagger(D) \leq \bar{r} = \text{embdim}(D) < r.$$

Therefore, $\text{embdim}(D) = \text{rank}\mathcal{K}^\dagger(D)$. Furthermore, since $\text{range}(\mathcal{K}^\dagger) = \mathcal{S}^n_C$, we have $\mathcal{K}^\dagger(D)e = 0$, implying that $\text{rank}\mathcal{K}^\dagger(D) \leq n-1$. □

Finally we give the following much celebrated theorem of Schoenberg.

Theorem 2.38 (Schoenberg (1935))**.** *A matrix $D \in \mathcal{S}^n_H$ is a Euclidean distance matrix if and only if $\mathcal{K}^\dagger(D)$ is positive semidefinite.*

Proof. If $D \in \mathcal{E}^n$, then Theorem 2.36 implies that $\mathcal{K}^\dagger(D) \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C$. On the other hand, suppose $D \in \mathcal{S}^n_H$ and $Y := \mathcal{K}^\dagger(D) \in \mathcal{S}^n_+$. Then $D = \mathcal{K}\mathcal{K}^\dagger(D)$, since $\mathcal{K}\mathcal{K}^\dagger$ is the orthogonal projection onto the subspace $\mathcal{S}^n_H$. Therefore, $D = \mathcal{K}(Y) \in \mathcal{E}^n$, since $\mathcal{K}(\mathcal{S}^n_+) = \mathcal{E}^n$. □

Norm of $\mathcal{K}$

We now provide a proof that the operator norm of $\mathcal{K}$ is $2\sqrt{n}$.

Lemma 2.39 (e.g., (Critchley, 1988, Cor. 2.10))**.**

$$\|\mathcal{K}\|_F := \max_{0\neq S\in\mathcal{S}^n} \frac{\|\mathcal{K}(S)\|_F}{\|S\|_F} = 2\sqrt{n}.$$

Proof. First we note that $\|\mathcal{K}\|_F = \sqrt{\rho(\mathcal{K}^*\mathcal{K})}$, where $\rho(\mathcal{K}^*\mathcal{K})$ is the *spectral radius* of $\mathcal{K}^*\mathcal{K}$, which is defined as

$$\rho(\mathcal{K}^*\mathcal{K}) := \max\left\{|\lambda| : \lambda \text{ is an eigenvalue of } \mathcal{K}^*\mathcal{K}\right\}.$$

Since $\mathcal{K}^*\mathcal{K}$ keeps the cone $\mathcal{S}^n_+ \cap \mathcal{S}_C$ invariant, (that is, $\mathcal{K}^*\mathcal{K}(\mathcal{S}^n_+ \cap \mathcal{S}_C) = \mathcal{S}^n_+ \cap \mathcal{S}_C$), we have by Perron-Frobenius theory (see, for example, the classic book Berman and Plemmons (1979)), that $\rho(\mathcal{K}^*\mathcal{K})$ is an eigenvalue, called the *Perron root*, and its eigenvector, called the *Perron vector*, is the only eigenvector in the relative interior of the cone kept invariant, $\mathcal{S}^n_+ \cap \mathcal{S}_C$.

This implies that we need only to show that there is an eigenvector Y of $\mathcal{K}^*\mathcal{K}$ in $\text{relint}(\mathcal{S}^n_+ \cap \mathcal{S}_C)$, and it satisfies $\mathcal{K}^*\mathcal{K}(Y) = 4nY$. To that end, we let Y be defined by

$$Y_{ij} := \begin{cases} n-1 & \text{if } i = j \\ -1 & \text{if } i \neq j. \end{cases}$$

Then,

$$\begin{aligned} \mathcal{K}(Y)_{ij} &= Y_{ii} + Y_{jj} - 2Y_{ij} \\ &= \begin{cases} 0 & \text{if } i = j \\ 2n & \text{if } i \neq j, \end{cases} \end{aligned}$$

which implies that

$$\begin{aligned}
\mathcal{K}^*(\mathcal{K}(Y))_{ij} &= \begin{cases} 2(\sum_{k=1}^n \mathcal{K}(Y)_{ik} - \mathcal{K}(Y)_{ii}) & \text{if } i = j \\ -2\mathcal{K}(Y)_{ij} & \text{if } i \neq j \end{cases} \\
&= \begin{cases} 4n(n-1) & \text{if } i = j \\ -4n & \text{if } i \neq j \end{cases} \\
&= 4nY_{ij}.
\end{aligned}$$

Therefore, $\mathcal{K}^*\mathcal{K}(Y) = 4nY$. To see that $Y \in \operatorname{relint}(\mathcal{S}_+^n \cap \mathcal{S}_C^n)$, we note that $Ye = 0$ and the eigenvalues of Y are 0 and $n-1$, with multiplicities of 1 and $n-1$, respectively. □

Summary of properties of $\mathcal{K}$

Here we provide a useful summary of the properties discussed above.

$$\begin{aligned}
&\mathcal{D}_e(y) := ye^T + ey^T & &\mathcal{D}_e^*(D) = 2De \\
&\mathcal{K}(Y) := \mathcal{D}_e(\operatorname{diag}(Y)) - 2Y & &\mathcal{K}^*(D) = 2(\operatorname{Diag}(De) - D)
\end{aligned}$$

$$\boxed{\mathcal{K}^\dagger(D) = -\frac{1}{2}J\operatorname{offDiag}(D)J, \quad \text{where } J := I - \frac{1}{n}ee^T}$$

$$\begin{aligned}
&\mathcal{S}_H^n := \{D \in \mathcal{S}^n : \operatorname{diag}(D) = 0\} & &\mathcal{K}\mathcal{K}^\dagger(D) = \operatorname{offDiag}(D) \\
&\mathcal{S}_C^n := \{Y \in \mathcal{S}^n : Ye = 0\} & &\mathcal{K}^\dagger\mathcal{K}(Y) = JYJ
\end{aligned}$$

$$\begin{aligned}
\operatorname{range}(\mathcal{K}) &= \mathcal{S}_H^n & \operatorname{null}(\mathcal{K}^\dagger) &= \operatorname{range}(\operatorname{Diag}) \\
\operatorname{range}(\mathcal{K}^\dagger) &= \mathcal{S}_C^n & \operatorname{null}(\mathcal{K}) &= \operatorname{range}(\mathcal{D}_e)
\end{aligned}$$

$$\begin{aligned}
\mathcal{K}(\mathcal{S}_C^n) &= \mathcal{S}_H^n & \mathcal{K}^\dagger(\mathcal{S}_H^n) &= \mathcal{S}_C^n \\
\mathcal{K}(\mathcal{S}_+^n \cap \mathcal{S}_C^n) &= \mathcal{E}^n & \mathcal{K}^\dagger(\mathcal{E}^n) &= \mathcal{S}_+^n \cap \mathcal{S}_C^n
\end{aligned}$$

$$\operatorname{embdim}(D) = \operatorname{rank}\mathcal{K}^\dagger(D), \quad \text{for } D \in \mathcal{E}^n$$

$$\|\mathcal{K}\|_F = 2\sqrt{n}$$

2.5.2 The cone of Euclidean distance matrices is closed

It is well-known that $\mathcal{E}^n$ is closed; see, for example, (Critchley, 1988, p. 103), (Hayden et al., 1991, p. 155-6), and (Dattorro, 2008, p. 477). However, we now present our own proof of this result that is based on the linear map $\mathcal{K}$ and its null space. Recall that the cone of Euclidean distance matrices satisfies $\mathcal{E}^n = \mathcal{K}(\mathcal{S}_+^n)$. To show that $\mathcal{E}^n$ is closed, we begin with a well-known result due to R. A. Abrams.

Theorem 2.40 (R. A. Abrams (Berman, 1973, Lemma 3.1))**.** *Let $S \subseteq \mathbb{E}$ be an arbitrary set, and $A\colon \mathbb{E} \to \mathbb{Y}$ be a linear map. Then*

$$A(S) \text{ is closed } \Leftrightarrow \; S + \operatorname{null}(A) \text{ is closed.}$$

Proof. (Berman, 1973, Lemma 3.1) First, we assume that $A(S)$ is closed. Let $z_k \in S + \operatorname{null}(A)$ such that $z_k \to z$, where we aim to show that $z \in S + \operatorname{null}(A)$. Since there exists $x_k \in S$ and $y_k \in \operatorname{null}(A)$ such that $z_k = x_k + y_k$, we have that

$$Az_k = Ax_k + Ay_k = Ax_k \in A(S), \quad \text{for all } k.$$

Since A is a continuous map, $Az_k \to Az$, and since $A(S)$ is closed, we have that $Az \in A(S)$. Therefore, there exists $x \in S$ such that $Az = Ax$, which implies that $z = x + y$, for some $y \in \operatorname{null}(A)$. Thus, $z \in S + \operatorname{null}(A)$, so $S + \operatorname{null}(A)$ is closed.

Now, we assume that $S + \operatorname{null}(A)$ is closed. Let $z_k \in A(S)$ such that $z_k \to z$; we aim to show that $z \in A(S)$. Note that there exists $x_k \in S$ such that $z_k = Ax_k$, implying that $x_k = A^\dagger z_k + y_k$, for some $y_k \in \operatorname{null}(A)$. Therefore,

$$A^\dagger z_k = x_k - y_k \in S + \operatorname{null}(A), \quad \text{for all } k.$$

Since $A^\dagger$ is a continuous map, $A^\dagger z_k \to A^\dagger z$, and since $S + \operatorname{null}(A)$ is closed, we have that $A^\dagger z \in S + \operatorname{null}(A)$. Therefore, $A^\dagger z = x - y$, for some $x \in S$ and $y \in \operatorname{null}(A)$. Thus, $Ax = AA^\dagger z$, and since $z_k \in \operatorname{range}(A)$, we have $AA^\dagger z_k = z_k$, implying that $AA^\dagger z = z$. Therefore, $Ax = z$, so $z \in A(S)$, as required. □

Recall that $\operatorname{null}(\mathcal{K}) = \operatorname{range}(\mathcal{D}_e)$. By Theorem 2.40,

$$\mathcal{E}^n = \mathcal{K}(\mathcal{S}^n_+) \text{ is closed } \Leftrightarrow \; \mathcal{S}^n_+ + \operatorname{range}(\mathcal{D}_e) \text{ is closed.}$$

Therefore, to show that $\mathcal{E}^n$ is closed, we only need to show that $\mathcal{S}^n_+ + \operatorname{range}(\mathcal{D}_e)$ is closed. We begin with the following lemma.

Lemma 2.41. *Let $Q := \begin{bmatrix} V & \frac{1}{\sqrt{n}}e \end{bmatrix} \in \mathbb{R}^{n \times n}$ be orthogonal. Let*

$$\Sigma := \left\{ Y \in \mathcal{S}^n : Y = \begin{bmatrix} 0 & v \\ v^T & t \end{bmatrix}, v \in \mathbb{R}^{n-1}, t \in \mathbb{R} \right\}.$$

Then

$$Q^T \operatorname{range}(\mathcal{D}_e) Q = \Sigma.$$

Proof. Let $Y \in \operatorname{range}(\mathcal{D}_e)$. Then $Y = ve^T + ev^T$, so

$$\begin{aligned} Q^T Y Q &= Q^T e v^T Q + Q^T v e^T Q \\ &= \begin{bmatrix} 0 \\ \sqrt{n} \end{bmatrix} v^T Q + Q^T v \begin{bmatrix} 0 \\ \sqrt{n} \end{bmatrix}^T \\ &= \begin{bmatrix} 0 \\ \sqrt{n} v^T Q \end{bmatrix} + \begin{bmatrix} 0 & \sqrt{n} Q^T v \end{bmatrix} \in \Sigma. \end{aligned}$$

Conversely, let $Y \in \Sigma$. Then $Y = \begin{bmatrix} 0 & v \\ v^T & t \end{bmatrix}$, so

$$\begin{aligned} QYQ^T &= \begin{bmatrix} V & \frac{1}{\sqrt{n}}e \end{bmatrix} \begin{bmatrix} 0 & v \\ v^T & t \end{bmatrix} \begin{bmatrix} V^T \\ \frac{1}{\sqrt{n}}e^T \end{bmatrix} \\ &= \frac{1}{\sqrt{n}} Vve^T + \frac{1}{\sqrt{n}} ev^TV^T + \frac{1}{n} tee^T \\ &= ye^T + ey^T, \end{aligned}$$

where $y := \frac{1}{\sqrt{n}}Vv + \frac{1}{2n}te$. Therefore, $Y = Q^T\mathcal{D}_e(y)Q$. □

We are now in a position to prove the main result of this section.

Proposition 2.42 (Critchley (1988)). *The cone of Euclidean distance matrices $\mathcal{E}^n$ is closed.*

Proof. We now present our own proof of this well-known result. We show that $\mathcal{S}^n_+ + \operatorname{range}(\mathcal{D}_e)$ is closed; this argument is inspired by the proof of Lemma 2.2 in Ramana et al. (1997). Let $Z_k \in \mathcal{S}^n_+ + \operatorname{range}(\mathcal{D}_e)$ such that $Z_k \to Z$. Then there exists $X_k \in \mathcal{S}^n_+$ and $Y_k \in \operatorname{range}(\mathcal{D}_e)$ such that $Z_k = X_k + Y_k$. By Lemma 2.41, $Q^TY_kQ \in \Sigma$. Thus, letting

$$\begin{bmatrix} H_k & u_k \\ u_k^T & t_k \end{bmatrix} := Q^TZ_kQ = Q^TX_kQ + Q^TY_kQ,$$

we have $H_k \succeq 0$, for all k. Thus

$$Q^TZ_kQ \to Q^TZQ =: \begin{bmatrix} H & u \\ u^T & t \end{bmatrix}$$

satisfies $H \succeq 0$. Therefore, $Z = Q\begin{bmatrix} H & 0 \\ 0 & 0 \end{bmatrix}Q^T + Q\begin{bmatrix} 0 & u \\ u^T & t \end{bmatrix}Q^T \in \mathcal{S}^n_+ + Q\Sigma Q^T = \mathcal{S}^n_+ + \operatorname{range}(\mathcal{D}_e)$, with the last equality following from Lemma 2.41. Therefore, $\mathcal{S}^n_+ + \operatorname{range}(\mathcal{D}_e)$ is closed, implying by Theorem 2.40 that $\mathcal{E}^n = \mathcal{K}(\mathcal{S}^n_+)$ is closed. □

We can also use the fact that $\mathcal{S}^n_+$ is a nice cone to obtain an alternate proof of Proposition 2.42. A cone $K \subseteq \mathbb{E}$ is called *nice* if $K^* + F^\perp$ is closed, for all $F \trianglelefteq K$.

Lemma 2.43 ((Tunçel and Wolkowicz, 2008, Lemma 2.25)). *Let $\{0\} \neq F \trianglelefteq \mathcal{S}^n_+$. Then $F^* = \mathcal{S}^n_+ + F^\perp = \operatorname{cl}(\mathcal{S}^n_+ + \operatorname{span} F^c)$ and $\mathcal{S}^n_+ + \operatorname{span} F^c$ is not closed.* □

Alternate proof of Proposition 2.42. Let $F := \mathcal{S}^n_+ \cap \mathcal{S}^n_C$. Then $\{0\} \neq F \trianglelefteq \mathcal{S}^n_+$. By Lemma 2.43, we have that $\mathcal{S}^n_+ + F^\perp$ is closed. Moreover,

$$F^\perp = \left(\mathcal{S}^n_+ \cap \mathcal{S}^n_C\right)^\perp = \operatorname{span}(\mathcal{S}^n_+ \cap \mathcal{S}^n_C)^\perp = (\mathcal{S}^n_C)^\perp = \operatorname{range}(\mathcal{D}_e).$$

Therefore, $\mathcal{S}^n_+ + \operatorname{range}(\mathcal{D}_e)$ is closed, implying by Theorem 2.40 that $\mathcal{E}^n = \mathcal{K}(\mathcal{S}^n_+)$ is closed. □

2.5.3 The dual cone of the Euclidean distance matrix cone

Since $\mathcal{E}^n = \mathcal{K}(\mathcal{S}^n_+)$, the dual cone of the Euclidean distance matrix cone $\mathcal{E}^n$ is given by

$$\begin{aligned}(\mathcal{E}^n)^* &= \{\Lambda \in \mathcal{S}^n : \langle \Lambda, D\rangle \geq 0, \text{ for all } D \in \mathcal{E}^n\} \\ &= \left\{\Lambda \in \mathcal{S}^n : \langle \Lambda, D\rangle \geq 0, \text{ for all } D \in \mathcal{K}(\mathcal{S}^n_+)\right\} \\ &= \left\{\Lambda \in \mathcal{S}^n : \langle \Lambda, \mathcal{K}(Y)\rangle \geq 0, \text{ for all } Y \in \mathcal{S}^n_+\right\} \\ &= \left\{\Lambda \in \mathcal{S}^n : \langle \mathcal{K}^*(\Lambda), Y\rangle \geq 0, \text{ for all } Y \in \mathcal{S}^n_+\right\} \\ &= \{\Lambda \in \mathcal{S}^n : \mathcal{K}^*(\Lambda) \succeq 0\}.\end{aligned}$$

It is shown that in Dattorro (2008) that

$$(\mathcal{E}^n)^* = \operatorname{range}(\operatorname{Diag}) - \mathcal{S}^n_+ \cap \mathcal{S}^n_C.$$

We now give a proof of this result using the linear maps $\mathcal{K}^*$ and $\mathcal{K}^{\dagger *}$. Recall that

$$\mathcal{K}^*(D) = 2(\operatorname{Diag}(De) - D) \quad \text{and} \quad \mathcal{K}^{\dagger *}(Y) = -\frac{1}{2}\operatorname{offDiag}(JYJ).$$

Proposition 2.44. *The map $\mathcal{K}^* \colon \mathcal{S}^n_H \to \mathcal{S}^n_C$ is a bijection and $\mathcal{K}^{\dagger *} \colon \mathcal{S}^n_C \to \mathcal{S}^n_H$ is its inverse. That is,*

$$\mathcal{K}^*(\mathcal{S}^n_H) = \mathcal{S}^n_C, \quad \mathcal{S}^n_H = \mathcal{K}^{\dagger *}(\mathcal{S}^n_C).$$

Proof. Result follows from Proposition 2.5 and the facts that $\operatorname{range}(\mathcal{K}^*) = \mathcal{S}^n_C$ and $\operatorname{range}(\mathcal{K}^{\dagger *}) = \mathcal{S}^n_H$. □

Proposition 2.45. *The map $\mathcal{K}^* \colon (\mathcal{E}^n)^* \cap \mathcal{S}^n_H \to \mathcal{S}^n_+ \cap \mathcal{S}^n_C$ is a bijection and $\mathcal{K}^{\dagger *} \colon \mathcal{S}^n_+ \cap \mathcal{S}^n_C \to (\mathcal{E}^n)^* \cap \mathcal{S}^n_H$ is its inverse. That is,*

$$\mathcal{K}^*((\mathcal{E}^n)^* \cap \mathcal{S}^n_H) = \mathcal{S}^n_+ \cap \mathcal{S}^n_C, \quad (\mathcal{E}^n)^* \cap \mathcal{S}^n_H = \mathcal{K}^{\dagger *}(\mathcal{S}^n_+ \cap \mathcal{S}^n_C).$$

Proof. Since $\operatorname{range}(\mathcal{K}^*) = \mathcal{S}^n_C$, we clearly have $\mathcal{K}^*((\mathcal{E}^n)^*) \subseteq \mathcal{S}^n_+ \cap \mathcal{S}^n_C$. Hence, $\mathcal{K}^*((\mathcal{E}^n)^* \cap \mathcal{S}^n_H) \subseteq \mathcal{S}^n_+ \cap \mathcal{S}^n_C$. Now suppose $Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C$, and let $\Lambda := \mathcal{K}^{\dagger *}(Y) \in \mathcal{S}^n_H$. Then

$$\mathcal{K}^*(\Lambda) = \mathcal{K}^*\mathcal{K}^{\dagger *}(Y) = JYJ = Y,$$

since $\mathcal{K}^*\mathcal{K}^{\dagger *}$ is the orthogonal projector onto $\operatorname{range}(\mathcal{K}^*) = \mathcal{S}^n_C$. Thus, $\mathcal{K}^*(\Lambda) \succeq 0$, so $\Lambda \in (\mathcal{E}^n)^*$, implying that $Y \in \mathcal{K}^*((\mathcal{E}^n)^* \cap \mathcal{S}^n_H)$. Therefore, $\mathcal{K}^* \colon (\mathcal{E}^n)^* \cap \mathcal{S}^n_H \to \mathcal{S}^n_+ \cap \mathcal{S}^n_C$ is surjective. The fact that $\mathcal{K}^* \colon (\mathcal{E}^n)^* \cap \mathcal{S}^n_H \to \mathcal{S}^n_+ \cap \mathcal{S}^n_C$ is injective and that $\mathcal{K}^{\dagger *} \colon \mathcal{S}^n_+ \cap \mathcal{S}^n_C \to (\mathcal{E}^n)^* \cap \mathcal{S}^n_H$ is its inverse follows from Proposition 2.44. □

However, $(\mathcal{E}^n)^* \cap \mathcal{S}^n_H \neq (\mathcal{E}^n)^*$, since $\operatorname{null}(\mathcal{K}^*) = \operatorname{range}(\operatorname{Diag})$ implies that $\operatorname{Diag}(\lambda) + \Lambda \in (\mathcal{E}^n)^*$, for all $\lambda \in \mathbb{R}^n$ and $\Lambda \in (\mathcal{E}^n)^*$. Therefore, we have the following result, which is clearly equivalent to the result from Dattorro (2008).

Corollary 2.46. *The dual cone of the Euclidean distance matrix cone $\mathcal{E}^n$ satisfies*

$$(\mathcal{E}^n)^* = \operatorname{range}(\operatorname{Diag}) \oplus \mathcal{K}^{\dagger *}(\mathcal{S}^n_+) = \operatorname{range}(\operatorname{Diag}) \ominus \operatorname{offDiag}(\mathcal{S}^n_+ \cap \mathcal{S}^n_C).$$

Proof. Follows from $(\mathcal{E}^n)^* = \operatorname{range}(\operatorname{Diag}) \oplus [(\mathcal{E}^n)^* \cap \mathcal{S}^n_H]$ and

$$(\mathcal{E}^n)^* \cap \mathcal{S}^n_H = \mathcal{K}^{\dagger *}(\mathcal{S}^n_+ \cap \mathcal{S}^n_C) = \mathcal{K}^{\dagger *}(\mathcal{S}^n_+) = -\operatorname{offDiag}(\mathcal{S}^n_+ \cap \mathcal{S}^n_C).$$ □

2.6 The Euclidean distance matrix completion problem

Following Bakonyi and Johnson (1995), we say that an n-by-n matrix D is a *partial Euclidean distance matrix* if every entry of D is either "specified" or "unspecified", $\operatorname{diag}(D) = 0$, and every fully specified principal submatrix of D is a Euclidean distance matrix. Note that this definition implies that every specified entry of D is nonnegative. In addition, if every fully specified principal submatrix of D has embedding dimension less than or equal to r, then we say that D is a partial Euclidean distance matrix in $\mathbb{R}^r$.

Associated with an n-by-n partial Euclidean distance matrix D is a *weighted undirected graph* $G = (N, E, \omega)$ with node set $N := \{1, \dots, n\}$, edge set

$$E := \{ij : i \neq j, \text{ and } D_{ij} \text{ is specified}\},$$

and edge weights $\omega \in \mathbb{R}^E_+$ with $\omega_{ij} = \sqrt{D_{ij}}$ for all $ij \in E$. We say that H is the 0–1 *adjacency matrix* of G if $H \in \mathcal{S}^n$ with

$$H_{ij} = \begin{cases} 1, & ij \in E \\ 0, & ij \notin E. \end{cases}$$

The *Euclidean distance matrix completion (EDMC) problem* asks to find a completion of a partial Euclidean distance matrix D; that is, if $G = (N, E, \omega)$ is the weighted graph associated with D, the Euclidean distance matrix completion problem can be posed as

$$\begin{array}{cc} \text{find} & \hat{D} \in \mathcal{E}^n \\ \text{s.t.} & \hat{D}_{ij} = D_{ij}, \ \forall ij \in E. \end{array} \tag{2.11}$$

Letting $H \in \mathcal{S}^n$ be the 0–1 adjacency matrix of G, the *Euclidean distance matrix completion problem* can be stated as

$$\begin{array}{cc} \text{find} & \hat{D} \in \mathcal{E}^n \\ \text{s.t.} & H \circ \hat{D} = H \circ D, \end{array} \tag{2.12}$$

where "$\circ$" represents the component-wise (or *Hadamard*) matrix product.

Using the linear map $\mathcal{K}$, we can substitute $\hat{D} = \mathcal{K}(Y)$, where $Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C$, in the EDM completion problem (2.12) to obtain the equivalent problem

$$\begin{array}{cc} \text{find} & Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C \\ \text{such that} & H \circ \mathcal{K}(Y) = H \circ D. \end{array} \tag{2.13}$$

2.6.1 The low-dimensional EDM completion problem

If D is a partial Euclidean distance matrix in $\mathbb{R}^r$, one is often interested in finding a Euclidean distance matrix completion of D that has embedding dimension r. The *low-dimensional Euclidean distance matrix completion problem* is

$$\begin{array}{rc}\text{find} & \hat{D} \in \mathcal{E}^n \\ \text{such that} & H \circ \hat{D} = H \circ D \\ & \operatorname{embdim}(\hat{D}) = r,\end{array} \tag{2.14}$$

where H is the 0–1 adjacency matrix of the graph G associated with D.

Using the linear map $\mathcal{K}$, we can state the low-dimensional EDM completion problem (2.14) as the following rank constrained semidefinite optimization problem:

$$\begin{array}{rc}\text{find} & Y \in \mathcal{S}^n_+ \\ \text{such that} & H \circ \mathcal{K}(Y) = H \circ D \\ & Ye = 0 \\ & \operatorname{rank}(Y) = r.\end{array} \tag{2.15}$$

2.6.2 Chordal EDM completions

Let G be a graph and C be a cycle in the graph. We say that C has a *chord* if there are two vertices on C that are connected by an edge which is not contained in C. Note that it is necessary that a cycle with a chord have length more than three. The graph G is called *chordal* if every cycle of the graph with length three or more has a *chord*. In the landmark paper Grone et al. (1984), they show the strong result that any partial semidefinite matrix with a chordal graph has a semidefinite completion; moreover, if a graph is not chordal, then there exists a partial semidefinite matrix with that graph, but having no semidefinite completion.

Due to the strong connection between Euclidean distance matrices and semidefinite matrices, it is not surprising that the result of chordal semidefinite completions of Grone et al. (1984) extends to the case of chordal Euclidean distance matrix completion. Indeed, Bakonyi and Johnson (1995) show the following:

1. any partial Euclidean distance matrix in $\mathbb{R}^r$ with a chordal graph can be completed to a distance matrix in $\mathbb{R}^r$;

2. every nonchordal graph has a partial Euclidean distance matrix that does not admit any distance matrix completions;

3. if the graph G of a partial Euclidean distance matrix D in $\mathbb{R}^r$ is chordal, then the completion of D is unique if and only if

$$\operatorname{rank}\left(\begin{bmatrix} 0 & e^T \\ e & D[S] \end{bmatrix}\right) = r + 2, \qquad \text{for all minimal vertex seperators } S \text{ of } G.$$

A set of vertices S in a graph G is a *minimal vertex seperator* if removing the vertices S from G separates some vertices u and v in G, and no proper subset of S separates u and v. It is discussed in, for example, Kumar and Madhavan (1998) how the *maximum cardinality search (MCS)* can be used to test in linear time if a graph G is chordal; moreover, Kumar and Madhavan (1998) show that MCS can also be used to compute all minimal vertex separators of a chordal graph in linear time. See also Alfakih (2003, 2005), for example, for more on the topic of uniqueness of Euclidean distance matrix completions.

2.6.3 Graph realization and graph rigidity

We note here the deep connection between the Euclidean distance matrix completion problem and the problem of *graph realization*. Let $N := \{1, \ldots, n\}$. Given a graph $G = (N, E, \omega)$ with edge weights $\omega \in \mathbb{R}^E_+$, the graph realization problem asks to find a mapping $p\colon N \to \mathbb{R}^r$ such that

$$\|p_i - p_j\| = \omega_{ij}, \qquad \text{for all } ij \in E;$$

in this case, we say that G has an *r-realization*, p. Clearly the graph realization problem is equivalent to the problem of Euclidean distance matrix completion, and the problem of the r-realizability of a weighted graph is equivalent to the low-dimensional Euclidean distance matrix completion problem.

A related problem is that of graph rigidity. Again, let $N := \{1, \ldots, n\}$. An unweighted graph $G = (N, E)$ together a mapping $p\colon N \to \mathbb{R}^r$ is called a *framework* (also *bar framework*) in $\mathbb{R}^r$, and is denoted by (G, p). Frameworks (G, p) in $\mathbb{R}^r$ and (G, q) in $\mathbb{R}^s$ are called *equivalent* if

$$\|p_i - p_j\| = \|q_i - q_j\|, \qquad \text{for all } ij \in E.$$

Furthermore, p and q are called *congruent* if

$$\|p_i - p_j\| = \|q_i - q_j\|, \qquad \text{for all } i, j = 1, \ldots, n. \tag{2.16}$$

Note that condition (2.16) can be stated as

$$\mathcal{K}(PP^T) = \mathcal{K}(QQ^T),$$

where $P \in \mathbb{R}^{n\times r}$ and $Q \in \mathbb{R}^{n\times s}$ are defined as

$$P := \begin{bmatrix} p_1^T \\ \vdots \\ p_n^T \end{bmatrix} \qquad \text{and} \qquad Q := \begin{bmatrix} q_1^T \\ \vdots \\ q_n^T \end{bmatrix}.$$

Thus, if $P^T e = Q^T e = 0$, then we have $PP^T = QQ^T$. By Proposition 3.2 (presented in the next chapter), $PP^T = QQ^T$ implies that:

- $P\bar{Q} = \begin{bmatrix} Q & 0 \end{bmatrix}$, for some orthogonal $\bar{Q} \in \mathbb{R}^r$, if $r \geq s$;
- $\begin{bmatrix} P & 0 \end{bmatrix} \bar{Q} = Q$, for some orthogonal $\bar{Q} \in \mathbb{R}^s$, if $s \geq r$.

A framework (G,p) in $\mathbb{R}^r$ is called *globally rigid* in $\mathbb{R}^r$ if all equivalent frameworks (G,q) in $\mathbb{R}^r$ satisfy condition (2.16). Similarly, a framework (G,p) in $\mathbb{R}^r$ is called *universally rigid* in $\mathbb{R}^r$ if, for all $s = 1,\ldots,n-1$, all equivalent frameworks (G,q) in $\mathbb{R}^s$ satisfy condition (2.16). Note that a framework (G,p) in $\mathbb{R}^r$ corresponds to the pair (H,D), where H is the 0–1 adjacency matrix of G, and D is a Euclidean distance matrix $\operatorname{embdim}(D) \leq r$. Therefore, the framework (G,p) given by (H,D) is globally rigid if

$$H \circ \hat{D} = H \circ D \quad \Rightarrow \quad \hat{D} = D,$$

for all $\hat{D} \in \mathcal{E}^n$ with $\operatorname{embdim}(\hat{D}) \leq r$; equivalently, (G,p) is globally rigid if

$$H \circ \mathcal{K}(Y) = H \circ D \quad \Rightarrow \quad \mathcal{K}(Y) = D,$$

for all $Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C$ with $\operatorname{rank}(Y) \leq r$. Moreover, the framework (G,p) given by (H,D) is universally rigid if

$$H \circ \hat{D} = H \circ D \quad \Rightarrow \quad \hat{D} = D,$$

for all $\hat{D} \in \mathcal{E}^n$; equivalently, (G,p) is universally rigid if

$$H \circ \mathcal{K}(Y) = H \circ D \quad \Rightarrow \quad \mathcal{K}(Y) = D,$$

for all $Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C$.

Graph realization and graph rigidity is a vast area of research, so we keep our discussion brief. More information can be found in, for example, Hendrickson (1992), Connelly (2005), So (2007), the recent survey Alfakih et al. (2009), and the references therein.

2.6.4 Low-dimensional EDM completion is NP-hard

We will now discuss the complexity of the low-dimensional Euclidean distance matrix completion decision problem (2.14) (equivalently, problem (2.15)). It was independently discovered by Saxe (1979) and Yemini (1979) that the problem of graph embeddability with integer edge weights, in $r = 1$ or $r = 2$ dimensions, is NP-complete by reduction from the NP-complete problem Partition. The Partition problem is defined as follows: Given a set of S of n integers, determine if there is a partition of S into two sets $S_1, S_2 \subseteq S$ such that the sum of the integers in S_1 equals the sum of the integers in S_2. Partition is one of Karp's 21 NP-complete problems in his landmark paper Karp (1972).

Theorem 2.47 ((Saxe, 1979, Theorem 3.2))**.** *The problem of* 1*-embeddability of graphs with integer weights is NP-complete.*

Proof. We follow the proof given in Saxe (1979). Let an instance of Partition be given by a subset of integers $S = \{s_1, \ldots, s_n\}$; we may assume, without loss of generality, that these integers are nonnegative by shifting them if necessary. Let

G be the cycle on n vertices with the n edge weights given by $s_1, \ldots, s_n$. A 1-embedding of G corresponds to folding this cycle flat onto the real number line. Walking along such a folded cycle, the edges partition into edges traversed from left-to-right, and edges traversed from right-to-left. We would then let S_1 be the edges traversed from left-to-right, and let S_2 be the edges traversed from right-to-left. Then the $\{S_1, S_2\}$ would form a partition of S, and since G is a cycle, the sum of the integers in S_1 would equal the sum of the integers in S_2, solving this instance of the PARTITION problem. Moreover, if no such 1-embedding of G is possible, then S cannot be partitioned into equal summing subsets (otherwise such a partition would give us a 1-embedding of G). Therefore, the NP-complete PARTITION problem reduces to the problem of 1-embeddability of graphs with integer weights, so we have shown the 1-embeddability with integer weights is NP-hard. Now, since checking a candidate solution for 1-embeddability with integer weights can be done in polynomial time, 1-embeddability with integer weights is in NP, and hence is NP-complete. □

Furthermore, by showing that for any r, r-embeddability of $\{1, 2\}$-weighted graphs is NP-hard, Saxe (1979) proves that the problem of r-embeddability of integer weighted graphs is strongly NP-hard.

In our tests, we work with so-called *unit disk graphs* which have realization in some Euclidean space satisfying their edge-weights such that the distance between vertices that are not connected is greater than some *radio range* R, and R is greater than all the edge weights. Thus, vertices are connected if and only if they are within radio range. However, Bădoiu et al. (2006) have shown that realizability of unit disk graphs is, in fact, NP-hard (again, by reduction from PARTITION). In the Bădoiu et al. (2006) proof, they again work with cycles, which are typically not uniquely realizable. In the applications we consider, we are most interested in unit disk graphs that have a unique realization. However, it is shown in Aspnes et al. (2004) and Aspnes et al. (2006) that there is no efficient algorithm for solving the unit disk graph localization problem, even if that graph has a unique realization, unless RP = NP (RP is the class of randomized polynomial time solvable problems). Problems in RP can be solved in polynomial time with high probability using a randomized algorithm. See also Laurent (2001), for example, for more on the topic of the complexity of Euclidean distance matrix completion and related problems.

Due to these hardness results for the low-dimensional Euclidean distance matrix problem, we turn to convex relaxations which can be solved efficiently, but may not solve our original problem.

2.6.5 Semidefinite relaxation of the low-dimensional EDM completion problem

The semidefinite relaxation of the low-dimensional EDM completion problem (2.15) is given by relaxing the hard $\operatorname{rank}(Y) = r$ constraint. Thus, we have the following tractable convex relaxation

$$\begin{array}{cc} \text{find} & Y \in \mathcal{S}_+^n \\ \text{such that} & H \circ \mathcal{K}(Y) = H \circ D \\ & Ye = 0. \end{array} \tag{2.17}$$

This semidefinite relaxation essentially allows the points to move into $\mathbb{R}^k$, where $k > r$. That is, if a solution Y of problem 2.17 has $\operatorname{rank}(Y) = k > r$, then we have found a Euclidean distance matrix completion of D with embedding dimension k; this is even possible if D has a completion with embedding dimension r, or even if D has a *unique* completion with embedding dimension r.

We can view this relaxation as a Lagrangian relaxation of the low-dimensional EDM completion problem.

Proposition 2.48. *Relaxation* (2.17) *is the Lagrangian relaxation of Problem* (2.15).

Proof. Problem (2.15) is equivalent to

$$\begin{array}{cc} \text{minimize} & 0 \\ \text{subject to} & H \circ \mathcal{K}\left(PP^T\right) = H \circ D \\ & P^T e = 0 \\ & P \in \mathbb{R}^{n \times r}. \end{array}$$

Following the recipe in Poljak et al. (1995), we square the linear constraints to obtain the equivalent problem

$$\begin{array}{cc} \text{minimize} & 0 \\ \text{subject to} & H \circ \mathcal{K}\left(PP^T\right) = H \circ D \\ & \|P^T e\|_2^2 = 0 \\ & P \in \mathbb{R}^{n \times r}. \end{array}$$

The Lagrangian of this problem is

$$\begin{array}{rcl} L(P, \Lambda, t) & = & \left\langle \Lambda, H \circ \left(\mathcal{K}\left(PP^T\right) - D\right)\right\rangle + t\|P^T e\|_2^2 \\ & = & \left\langle \mathcal{K}^*(H \circ \Lambda), PP^T \right\rangle - \langle H \circ \Lambda, D\rangle + te^T PP^T e \\ & = & \left\langle \mathcal{K}^*(H \circ \Lambda) + tee^T, PP^T \right\rangle - \langle H \circ \Lambda, D\rangle \\ & = & \operatorname{trace}\left[P^T\left(\mathcal{K}^*(H \circ \Lambda) + tee^T\right) P\right] - \langle H \circ \Lambda, D\rangle . \end{array}$$

Therefore, the Lagrangian dual

$$\sup_{\Lambda, t} \inf_P L(P, \Lambda, t)$$

is given by

$$\begin{array}{cc} \text{maximize} & -\langle H \circ \Lambda, D\rangle \\ \text{subject to} & \mathcal{K}^*(H \circ \Lambda) + tee^T \succeq 0. \end{array}$$

This is a semidefinite optimization problem with dual

$$\begin{array}{rc} \text{minimize} & 0 \\ \text{subject to} & H \circ \mathcal{K}(Y) = H \circ D \\ & \left\langle Y, ee^T \right\rangle = 0 \\ & Y \in \mathcal{S}^n_+. \end{array}$$

This last problem is equivalent to problem (2.17). □

2.6.6 Duality of the semidefinite relaxation

Problem (2.17) is equivalent to

$$\begin{array}{rc} \text{minimize} & 0 \\ \text{subject to} & H \circ \mathcal{K}(Y) = H \circ D \\ & Ye = 0 \\ & Y \in \mathcal{S}^n_+ \end{array}$$

The Lagrangian of this problem is given by

$$\begin{aligned} L(Y, \Lambda, v) &= \langle \Lambda, H \circ \mathcal{K}(Y) - H \circ D \rangle + \langle v, Ye \rangle \\ &= \langle \mathcal{K}^*(H \circ \Lambda), Y \rangle - \langle H \circ \Lambda, D \rangle + \left\langle ev^T, Y \right\rangle \\ &= \left\langle \mathcal{K}^*(H \circ \Lambda) + \frac{1}{2}\left(ev^T + ve^T\right), Y \right\rangle - \langle H \circ \Lambda, D \rangle \\ &= \left\langle \mathcal{K}^*(H \circ \Lambda) + \frac{1}{2}\mathcal{D}_e(v), Y \right\rangle - \langle H \circ \Lambda, D \rangle . \end{aligned}$$

Therefore, the Lagrangian dual problem is

$$\sup_{\Lambda, v} \inf_{Y \in \mathcal{S}^n_+} L(Y, \Lambda, v),$$

which is equivalent to

$$\sup \left\{ - \langle H \circ \Lambda, D \rangle : \mathcal{K}^*(H \circ \Lambda) + \frac{1}{2}\mathcal{D}_e(v) \succeq 0 \right\}.$$

From this dual problem, we obtain the following partial description of the conjugate face of the minimal face of the EDM completion problem (2.17).

Proposition 2.49. *Let $F := \operatorname{face}(\mathcal{F})$, where*

$$\mathcal{F} := \left\{ Y \in \mathcal{S}^n_+ : H \circ \mathcal{K}(Y) = H \circ D, Ye = 0 \right\}.$$

If $\mathcal{F} \neq \emptyset$, then

$$\operatorname{face}\left\{ S \in \mathcal{S}^n_+ : S = \mathcal{K}^*(H \circ \Lambda) + \frac{1}{2}\mathcal{D}_e(v), \langle H \circ \Lambda, D \rangle = 0 \right\} \trianglelefteq F^c.$$

Proof. Proof follows as in the proof of Proposition 2.19. □

Using Proposition 2.49, we obtain the following partial description of the minimal face of the EDM completion problem (2.17).

Corollary 2.50. *If* $\mathcal{F} := \left\{ Y \in \mathcal{S}^n_+ : H \circ \mathcal{K}(Y) = H \circ D, Ye = 0 \right\} \neq \emptyset$ *and there exists* $S \succeq 0$ *such that*

$$S = \mathcal{K}^*(H \circ \Lambda) + \frac{1}{2}\mathcal{D}_e(v) \quad \textit{and} \quad \langle H \circ \Lambda, D \rangle = 0,$$

for some $\Lambda \in \mathcal{S}^n$ *and* $v \in \mathbb{R}^n$*, then*

$$\operatorname{face}(\mathcal{F}) \trianglelefteq \mathcal{S}^n_+ \cap \{S\}^\perp.$$

Proof. By Proposition 2.49, we have that $S \in \operatorname{face}(\mathcal{F})^c$. By Proposition 2.11, we have that $\operatorname{face}(\mathcal{F}) \trianglelefteq \mathcal{S}^n_+ \cap \{S\}^\perp$. □

For example, we can apply Corollary 2.50 as follows. Let $\Lambda := 0$ and $v := e$. Then,

$$\mathcal{K}^*(H \circ \Lambda) + \frac{1}{2}\mathcal{D}_e(v) = ee^T \succeq 0, \quad \text{and} \quad \langle H \circ \Lambda, D \rangle = 0.$$

Thus,

$$\operatorname{face}\left\{ Y \in \mathcal{S}^n_+ : H \circ \mathcal{K}(Y) = H \circ D, Ye = 0 \right\} \trianglelefteq \mathcal{S}^n_+ \cap \left\{ ee^T \right\}^\perp = V\mathcal{S}^{n-1}_+ V^T,$$

where $\begin{bmatrix} V & \frac{1}{\sqrt{n}}e \end{bmatrix} \in \mathbb{R}^{n \times n}$ is orthogonal. Note that we have $V^TV = I$ and $VV^T = J$, where J is the orthogonal projector onto $\{e\}^\perp$. Therefore, Problem (2.17) is equivalent to the reduced problem

$$\begin{array}{cc} \text{find} & Z \in \mathcal{S}^{n-1}_+ \\ \text{such that} & H \circ \mathcal{K}_V(Z) = H \circ D, \end{array}$$

where $\mathcal{K}_V : \mathcal{S}^{n-1} \to \mathcal{S}^n$ is defined by

$$\mathcal{K}_V(Z) := \mathcal{K}(VZV^T).$$

We will return to this topic of facial reduction again in Chapter 4.

2.6.7 Rank minimization heuristics for the EDM completion problem

In order to encourage having a solution of the semidefinite relaxation with low rank, the following heuristic has been suggested by Weinberger et al. (2004) and used with great success by Biswas et al. (2006) on the sensor network localization problem. The idea is that we can try to “flatten” the graph associated with a partial Euclidean distance matrix by pushing the nodes of the graph away from each other as much as possible. This flattening of the graph then corresponds to reducing the rank of the semidefinite solution of the relaxation. Geometrically, this makes a lot of sense, and Weinberger et al. (2004) gives this nice analogy: a loose string on the table can occupy two dimensions, but the same string pulled taut occupies just one dimension.

Therefore, we would like to maximize the objective function

$$\sum_{i,j=1}^{n} \|p_i - p_j\|^2 = e^T \mathcal{K}(PP^T)e,$$

where

$$P := \begin{bmatrix} p_1^T \\ \vdots \\ p_n^T \end{bmatrix},$$

subject to the distance constraints holding. Moreover, if we include the constraint that $P^T e = 0$, then

$$\begin{aligned} e^T \mathcal{K}(PP^T)e &= \left\langle ee^T, \mathcal{K}(PP^T) \right\rangle \\ &= \left\langle \mathcal{K}^*(ee^T), PP^T \right\rangle \\ &= \left\langle 2(\operatorname{Diag}(ee^T e) - ee^T), PP^T \right\rangle \\ &= \left\langle 2(nI - ee^T), PP^T \right\rangle \\ &= \left\langle 2nI, PP^T \right\rangle - \left\langle ee^T, PP^T \right\rangle \\ &= 2n \cdot \operatorname{trace}(PP^T). \end{aligned}$$

Note that

$$\operatorname{trace}(PP^T) = \sum_{i=1}^{n} \|p_i\|^2,$$

so pushing the nodes away from each other is equivalent to pushing the nodes away from the origin, under the assumption that the points are centred at the origin. Normalizing this objective function by dividing by the constant $2n$, substituting $Y = PP^T$, and relaxing the rank constraint on Y, we obtain the following *regularized* semidefinite relaxation of the low-dimensional Euclidean distance matrix completion problem:

$$\begin{array}{rc} \text{maximize} & \operatorname{trace}(Y) \\ \text{subject to} & H \circ \mathcal{K}(Y) = H \circ D \\ & Y \in \mathcal{S}_+^n \cap \mathcal{S}_C^n. \end{array} \tag{2.18}$$

It is very interesting to compare this heuristic with the *nuclear norm* rank minimization heuristic that has received much attention lately; see, for example, Fazel (2002), Recht et al. (2008a), Recht et al. (2008b), Candès and Recht (2008), Candès and Plan (2009), and Ames and Vavasis (2009). The *nuclear norm* of a matrix $X \in \mathbb{R}^{m\times n}$ is given by

$$\|X\|_* := \sum_{i=1}^{k} \sigma_i(X),$$

where $\sigma_i(X)$ is the i^{th} largest singular value of X, and $\operatorname{rank}(X) = k$. The nuclear norm of a symmetric matrix $Y \in \mathcal{S}^n$ is then given by

$$\|Y\|_* = \sum_{i=1}^{n} |\lambda_i(Y)|.$$

Furthermore, for $Y \in \mathcal{S}^n_+$, we have $\|Y\|_* = \operatorname{trace}(Y)$.

Since we are interested in solving the *rank minimization* problem,

$$\begin{array}{cc} \text{minimize} & \operatorname{rank}(Y) \\ \text{subject to} & H \circ \mathcal{K}(Y) = H \circ D \\ & Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C, \end{array} \tag{2.19}$$

the nuclear norm heuristic gives us the problem

$$\begin{array}{cc} \text{minimize} & \operatorname{trace}(Y) \\ \text{subject to} & H \circ \mathcal{K}(Y) = H \circ D \\ & Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C. \end{array} \tag{2.20}$$

However, this is geometrically interpreted as trying to bring all the nodes of the graph as close to the origin as possible. Intuition tells us that this approach would produce a solution with a high embedding dimension.

A comparison between these two approaches of trace minimization and trace maximization appears in the two related papers Göring et al. (2008) and Göring et al. (2009) on graph realization. In a recent talk by Helmberg, it is observed in numerical tests that trace minimization produces graph realizations in high dimensions, while the graph realizations obtained from trace maximization tend to be in a much lower dimensional space.

Another rank minimization heuristic that has been considered recently in Fazel et al. (2003) is the log-det maximization heuristic. There they successfully computed solutions to Euclidean distance matrix problems with very low embedding dimension via this heuristic.

Chapter 3

Applications of Euclidean Distance Matrices

There are many applications of Euclidean distance matrices, including wireless sensor network localization, molecular conformation in chemistry and bioinformatics, and multidimensional scaling in statistics and machine learning. Here we will focus on the first two applications.

3.1 Sensor network localization

The *sensor network localization (SNL)* problem is a low-dimensional Euclidean distance matrix completion problem in which the position of a subset of the nodes is specified. In the sensor network localization problem, we have $n-m$ unknown points

$$x_1, \ldots, x_{n-m} \in \mathbb{R}^r,$$

called *sensors*, and m known points

$$a_1, \ldots, a_m \in \mathbb{R}^r,$$

called *anchors*. We let

$$X := \begin{bmatrix} x_1^T \\ \vdots \\ x_{n-m}^T \end{bmatrix} \in \mathbb{R}^{(n-m)\times r}, \quad A := \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix} \in \mathbb{R}^{m\times r}, \quad \text{and} \quad P := \begin{bmatrix} X \\ A \end{bmatrix} \in \mathbb{R}^{n\times r}.$$

We will partition the n-by-n partial Euclidean distance matrix as

$$D =: \begin{array}{c c} & \begin{array}{cc} \scriptstyle n-m & \scriptstyle m \end{array} \\ \begin{array}{c} \scriptstyle n-m \\ \scriptstyle m \end{array} & \left[\begin{array}{cc} D_{11} & D_{12} \\ D_{12}^T & D_{22} \end{array}\right] \end{array}.$$

The sensor network localization problem can then be stated as follows: Given $A \in \mathbb{R}^{m\times r}$ and an n-by-n partial Euclidean distance matrix D satisfying $D_{22} = \mathcal{K}(AA^T)$, with corresponding 0–1 adjacency matrix H,

$$\begin{array}{rl} \text{find} & X \in \mathbb{R}^{(n-m)\times r} \\ \text{such that} & H \circ \mathcal{K}\left(PP^T\right) = H \circ D \\ & P = \begin{bmatrix} X \\ A \end{bmatrix} \in \mathbb{R}^{n\times r}. \end{array} \tag{3.1}$$

Before discussing the semidefinite relaxation of the sensor network localization problem, we first discuss the important assumption having anchors whose affine hull has full dimension, and how we may, in fact, remove the constraint on P that its bottom block must equal the anchor positions A.

3.1.1 Anchors and the Procrustes problem

A key assumption that we need to make is that the affine hull of the anchors $A \in \mathbb{R}^{m\times r}$ is full-dimensional. That is, we need

$$\operatorname{aff}\{a_1, \ldots, a_m\} = \mathbb{R}^r.$$

This holds if, for all $x \in \mathbb{R}^r$, there exists $\lambda_1, \ldots, \lambda_m \in \mathbb{R}$ such that $\sum_{i=1}^m \lambda_i = 1$ and

$$x = \lambda_1 a_1 + \cdots + \lambda_m a_m.$$

Another way to say this is that

$$\mathbb{R}^r \times \{1\} \subseteq \operatorname{range}\left(\begin{bmatrix} A^T \\ e^T \end{bmatrix}\right),$$

implying that

$$\mathbb{R}^{r+1} = \operatorname{span}(\mathbb{R}^r \times \{1\}) \subseteq \operatorname{range}\left(\begin{bmatrix} A^T \\ e^T \end{bmatrix}\right) \subseteq \mathbb{R}^{r+1},$$

which further implies that $\operatorname{null}\left(\begin{bmatrix} A & e \end{bmatrix}\right) = \{0\}$. Thus, we have that the affine hull of the anchors is full-dimensional if and only if $\begin{bmatrix} A & e \end{bmatrix} \in \mathbb{R}^{m\times(r+1)}$ has full column rank. We immediately see that this assumption implies that we must have $m \geq r+1$, and we also must have that A itself has full column rank. The reason we make this assumption is due to the fact that when the affine hull of the anchors is not full dimensional, we may reflect the position of the anchors across any hyperplane containing the affine hull of the anchors, giving us another distinct solution of the sensor network localization problem (3.1).

On the other hand, from the following observation, we can discard the constraint $P_2 = A$ in problem (3.1), where $P \in \mathbb{R}^{n\times r}$ is partitioned as

$$P =: \begin{array}{c} \\ {\scriptstyle n-m} \\ {\scriptstyle m} \end{array} \!\!\begin{array}{c} {\scriptstyle r} \\ \left[\begin{array}{c} P_1 \\ P_2 \end{array}\right] \end{array}. \tag{3.2}$$

Suppose that P satisfies $H \circ \mathcal{K}\left(PP^T\right) = H \circ D$ in problem (3.1). Then

$$\mathcal{K}(P_2 P_2^T) = \mathcal{K}(AA^T).$$

Assuming, without loss of generality, that $P_2^T e = 0$ and $A^T e = 0$, we have that $P_2 P_2^T = AA^T$. As we will see, this implies that there exists an orthogonal $Q \in \mathbb{R}^r$ such that $P_2 Q = A$. Since such an orthogonal transformation does not change the distances between points, we have that PQ is a feasible solution of the sensor network localization problem (3.1), with sensor positions $X := P_1 Q$.

To show the existence of such an orthogonal transformation, we will now discuss the following classical problem. Given $A, B \in \mathbb{R}^{m \times n}$, the *Procrustes problem* asks to find an orthogonal $Q \in \mathbb{R}^r$ that minimizes $\|BQ - A\|_F$; that is

$$\begin{array}{rl} \text{minimize} & \|BQ - A\|_F \\ \text{subject to} & Q^T Q = I. \end{array} \tag{3.3}$$

The general solution to this problem was first given in Schönemann (1966) (see also Green (1952), Higham (1986), and Golub and Van Loan (1996)).

Theorem 3.1 (Schönemann (1966))**.** *Let $A, B \in \mathbb{R}^{m \times n}$. Then $Q := UV^T$ is an optimal solution of the Procrustes problem* (3.3)*, where $B^T A = U \Sigma V^T$ is the singular value decomposition of $B^T A$.*

Proof. Here we follow the proof given in (Golub and Van Loan, 1996, Section 12.4.1). We first observe that if $Q \in \mathbb{R}^r$ is orthogonal, then

$$\|BQ - A\|_F^2 = \langle BQ - A, BQ - A \rangle = \|B\|_F^2 - 2\left\langle Q, B^T A \right\rangle + \|A\|_F^2 .$$

Therefore, we solve the Procrustes problem by finding Q orthogonal that maximizes

$$\left\langle Q, B^T A \right\rangle = \left\langle Q, U \Sigma V^T \right\rangle = \left\langle U^T Q V, \Sigma \right\rangle = \sum_{i=1}^{p} (U^T Q V)_{ii} \sigma_{ii},$$

where $\sigma_1, \ldots, \sigma_p$ are the nonzero singular values of $B^T A$. Since $U^T Q V$ is itself orthogonal, we have $|(U^T Q V)_{ii}| \leq 1$, for all i. Thus,

$$\left\langle Q, B^T A \right\rangle \leq \sum_{i=1}^{p} \sigma_{ii},$$

and this upper bound is attained by choosing $U^T Q V = I$. Therefore, $Q := UV^T$ is optimal for the Procrustes problem (3.3). □

From Theorem 3.1, we have the following useful consequence.

Proposition 3.2. *Let $A, B \in \mathbb{R}^{m \times n}$. Then $AA^T = BB^T$ if and only if there exists an orthogonal $Q \in \mathbb{R}^{n \times n}$ such that $BQ = A$.*

Proof. Clearly, if $BQ = A$ for some orthogonal $Q \in \mathbb{R}^{n \times n}$, then $AA^T = BQQ^TB^T = BB^T$. Now suppose that $AA^T = BB^T$. Let $B^TA = U\Sigma V^T$ be the singular value decomposition of B^TA. From Theorem 3.1, we have that $Q := UV^T$ minimizes $\|BQ - A\|_F$. Therefore, we need only show that $\|BQ - A\|_F = 0$. Observe that

$$\begin{aligned}\|BQ - A\|_F^2 &= \langle BQ - A, BQ - A\rangle \\ &= \operatorname{trace}\left((BQ - A)^T(BQ - A)\right) \\ &= \operatorname{trace}\left(B^TB - Q^TB^TA - A^TBQ + A^TA\right).\end{aligned}$$

We now claim that $Q^TB^TA = A^TA$. First we observe that, since $AA^T = BB^T$, we have

$$\left(A^TA\right)^2 = A^TAA^TA = A^TBB^TA = V\Sigma U^TU\Sigma V^T = V\Sigma^2V^T.$$

Thus, $A^TA = V\Sigma V^T$, since the positive semidefinite root of a positive semidefinite matrix is unique. Therefore, $Q^TB^TA = VU^TU\Sigma V^T = V\Sigma V^T = A^TA$. Similarly, we can show that $A^TBQ = A^TA$. Thus,

$$\begin{aligned}\|BQ - A\|_F^2 &= \operatorname{trace}\left(B^TB - Q^TB^TA - A^TBQ + A^TA\right) \\ &= \operatorname{trace}\left(B^TB - A^TA - A^TA + A^TA\right) \\ &= \operatorname{trace}\left(B^TB - A^TA\right) \\ &= \operatorname{trace}\left(BB^T - AA^T\right) \\ &= 0,\end{aligned}$$

so $BQ = A$. □

Therefore, the method of discarding the constraint $P_2 = A$ discussed above is justified. This suggests a simple approach for solving the sensor network localization problem. First we solve the equivalent low-dimensional Euclidean distance matrix completion problem,

$$\begin{array}{rl} \text{find} & \bar{Y} \in \mathcal{S}_+^n \cap \mathcal{S}_C^n \\ \text{such that} & H \circ \mathcal{K}\left(\bar{Y}\right) = H \circ D \\ & \operatorname{rank}(\bar{Y}) = r. \end{array} \tag{3.4}$$

Note that without the anchor constraint, we can now assume that our points are centred at the origin, hence the constraint $\bar{Y} \in \mathcal{S}_C^n$. Factoring $\bar{Y} = PP^T$, for some $P \in \mathbb{R}^{n \times r}$, we then translate P so that $P_2^Te = 0$. Similarly, we translate the anchors so that $A^Te = 0$. We then apply an orthogonal transformation to align P_2 with the anchors positions in A by solving a Procrustes problem using the solution technique in Theorem 3.1. Finally, if we translated to centre the anchors, we simply translate everything back accordingly.

However, as problems with rank constraints are often NP-hard, problem (3.4) may be very hard to solve. In fact, we saw in Section 2.6.4 that the general problem (3.4) can be reduced to the NP-complete problem Partition. Therefore, we now turn to investigating semidefinite relaxations of the sensor network localization problem.

Based on our discussion in this section, the relaxation that immediately comes to mind is the semidefinite relaxation of the low-dimensional Euclidean distance matrix completion problem (2.15), which is given by relaxing the rank constraint in problem (3.4); that is, we get the relaxation

$$\begin{array}{rl} \text{find} & \bar{Y} \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C \\ \text{such that} & H \circ \mathcal{K}\left(\bar{Y}\right) = H \circ D. \end{array} \tag{3.5}$$

However, as we will see in the next section, it is possible to take advantage of the structure available in the anchor constraints. We will show how this structure allows us to reduce the size of the semidefinite relaxation.

3.1.2 Semidefinite relaxation of the SNL problem

To get a semidefinite relaxation of the sensor network localization problem, we start by writing problem (3.1) as,

$$\begin{array}{rl} \text{find} & X \in \mathbb{R}^{(n-m)\times r} \\ \text{such that} & H \circ \mathcal{K}\left(\bar{Y}\right) = H \circ D \\ & \bar{Y} = \begin{bmatrix} XX^T & XA^T \\ AX^T & AA^T \end{bmatrix}. \end{array} \tag{3.6}$$

Next we show that the nonlinear second constraint on $\bar{Y}$, the block-matrix constraint, may be replaced by a semidefinite constraint, a linear constraint, and a rank constraint on $\bar{Y}$.

Proposition 3.3. *Let $A \in \mathbb{R}^{m\times r}$ have full column rank, and let $\bar{Y} \in \mathcal{S}^n$ be partitioned as*

$$\bar{Y} =: \begin{array}{c c} & \begin{array}{cc} n-m & m \end{array} \\ \begin{array}{c} n-m \\ m \end{array} & \left[\begin{array}{cc} \bar{Y}_{11} & \bar{Y}_{12} \\ \bar{Y}_{12}^T & \bar{Y}_{22} \end{array}\right] \end{array}.$$

Then the following hold:

1. *If $\bar{Y}_{22} = AA^T$ and $\bar{Y} \succeq 0$, then there exists $X \in \mathbb{R}^{(n-m)\times r}$ such that*

$$\bar{Y}_{12} = XA^T,$$

and X is given uniquely by $X = \bar{Y}_{12}A^{\dagger T}$.

2. *There exists $X \in \mathbb{R}^{(n-m)\times r}$ such that $\bar{Y} = \begin{bmatrix} XX^T & XA^T \\ AX^T & AA^T \end{bmatrix}$ if and only if $\bar{Y}$ satisfies*

$$\left\{ \begin{array}{r} \bar{Y} \succeq 0 \\ \bar{Y}_{22} = AA^T \\ \operatorname{rank}(\bar{Y}) = r \end{array} \right\}.$$

Proof. Suppose that $\bar{Y}_{22} = AA^T$ and $\bar{Y} \succeq 0$; let $X := \bar{Y}_{12}A^{\dagger T}$. Since $\operatorname{rank}(A) = r$ and $\operatorname{range}(A) = \operatorname{null}(A^T)^\perp$, there exists a full column rank matrix $V \in \mathbb{R}^{m\times(m-r)}$ such that $\operatorname{range}(V) = \operatorname{null}(A^T)$. Then, $A^TV = 0$, so we have

$$\begin{bmatrix} I & 0 \\ 0 & V \end{bmatrix}^T \begin{bmatrix} \bar{Y}_{11} & \bar{Y}_{12} \\ \bar{Y}_{12}^T & AA^T \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & V \end{bmatrix} = \begin{bmatrix} \bar{Y}_{11} & \bar{Y}_{12}V \\ V^T\bar{Y}_{12}^T & 0 \end{bmatrix} \succeq 0,$$

since $\bar{Y} \succeq 0$ and $\bar{Y}_{22} = AA^T$. It is well-known that if a positive semidefinite matrix has a zero diagonal entry, then the entire row and column corresponding to that diagonal entry are also zero. Therefore, $V^T\bar{Y}_{12}^T = 0$, implying that

$$\operatorname{range}(\bar{Y}_{12}^T) \subseteq \operatorname{null}(V^T) = \operatorname{range}(V)^\perp = \operatorname{range}(A).$$

Thus, there exists $\Phi \in \mathbb{R}^{r\times(m-r)}$ such that $\bar{Y}_{12}^T = A\Phi$. Since A is full column rank, we have

$$\Phi = (A^TA)^{-1}A^T\bar{Y}_{12}^T = A^\dagger\bar{Y}_{12}^T = X^T.$$

Therefore, X satisfies

$$\bar{Y} = \begin{bmatrix} \bar{Y}_{11} & XA^T \\ AX^T & AA^T \end{bmatrix}. \tag{3.7}$$

Moreover, if $\bar{X}$ also satisfies (3.7), then $A(\bar{X}^T - X^T) = 0$, so $\bar{X} = X$, since A has full column rank. Thus, we have shown Item 1.

To show Item 2, we first note that if there exists $X \in \mathbb{R}^{(n-m)\times r}$ such that

$$\bar{Y} = \begin{bmatrix} XX^T & XA^T \\ AX^T & AA^T \end{bmatrix},$$

then $\bar{Y} \succeq 0$, $\bar{Y}_{22} = AA^T$, and $\operatorname{rank}(\bar{Y}) = r$. Now suppose that $\bar{Y} \succeq 0$, $\bar{Y}_{22} = AA^T$, and $\operatorname{rank}(\bar{Y}) = r$. Based on the result in Item 1, it remains to show that $\bar{Y}_{11} = XX^T$. Since $\bar{Y} \succeq 0$ and $\operatorname{rank}(\bar{Y}) = r$, there exists $\bar{P} \in \mathbb{R}^{n\times r}$ such that $\bar{Y} = \bar{P}\bar{P}^T$. Let $\bar{P}$ be partitioned as

$$\bar{P} =: \begin{array}{c} \\ {\scriptstyle n-m} \\ {\scriptstyle m} \end{array} \begin{array}{c} {\scriptstyle r} \\ \left[\begin{array}{c} \bar{P}_1 \\ \bar{P}_2 \end{array}\right] \end{array}.$$

Then we have:

$$\begin{aligned} \bar{Y}_{11} &= \bar{P}_1\bar{P}_1^T; \\ XA^T = \bar{Y}_{12} &= \bar{P}_1\bar{P}_2^T; \\ AA^T = \bar{Y}_{22} &= \bar{P}_2\bar{P}_2^T. \end{aligned}$$

Since $AA^T = \bar{P}_2\bar{P}_2^T$, by Proposition 3.2 there exists $Q \in \mathbb{R}^{r\times r}$ such that $\bar{P}_2Q = A$. Thus,

$$AX^T = \bar{P}_2\bar{P}_1^T = AQ^T\bar{P}_1^T,$$

so $X^T = Q^T\bar{P}_1^T$, since A has full column rank. Therefore, we have

$$\bar{Y}_{11} = \bar{P}_1\bar{P}_1^T = XQ^TQX^T = XX^T,$$

completing the proof of Item 2. □

Now, by Proposition 3.3, we have that problem (3.1) is equivalent to

$$\begin{array}{rl} \text{find} & \bar{Y} \in \mathcal{S}_+^n \\ \text{such that} & H \circ \mathcal{K}\left(\bar{Y}\right) = H \circ D \\ & \bar{Y}_{22} = AA^T \\ & \operatorname{rank}(\bar{Y}) = r. \end{array} \tag{3.8}$$

Relaxing the hard rank constraint, we obtain the semidefinite relaxation of the sensor network localization problem:

$$\begin{array}{rl} \text{find} & \bar{Y} \in \mathcal{S}_+^n \\ \text{such that} & H \circ \mathcal{K}\left(\bar{Y}\right) = H \circ D \\ & \bar{Y}_{22} = AA^T \end{array} \tag{3.9}$$

As in Proposition 2.48, this relaxation is equivalent to the Lagrangian relaxation of the sensor network localization problem (3.8). Moreover, this relaxation essentially allows the sensors to move into a higher dimension. To obtain a solution in $\mathbb{R}^r$, we may either project the positions in the higher dimension onto $\mathbb{R}^r$, or we can try a best rank-r approximation approach; we will say more on this later.

3.1.3 Further transformations of the semidefinite relaxation of the SNL problem

From Proposition 3.3, we have that $\bar{Y} \succeq 0$ and $\bar{Y}_{22} = AA^T$ implies that

$$\bar{Y} = \begin{bmatrix} Y & XA^T \\ AX^T & AA^T \end{bmatrix}$$

for some $Y \in \mathcal{S}_+^{n-m}$ and $X = \bar{Y}_{12} A^{\dagger T} \in \mathbb{R}^{(n-m)\times r}$. Now we make the key observation that having anchors in a Euclidean distance matrix problem implies that our feasible points are restricted to a face of the semidefinite cone. This is because, if $\bar{Y}$ is feasible for problem (3.9), then

$$\bar{Y} = \begin{bmatrix} I & 0 \\ 0 & A \end{bmatrix} \begin{bmatrix} Y & X \\ X^T & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & A \end{bmatrix}^T \in U_A \mathcal{S}_+^{n-m+r} U_A^T,$$

where

$$U_A := \begin{array}{c c} & \begin{array}{cc} n-m & r \end{array} \\ \begin{array}{c} n-m \\ m \end{array} & \left[\begin{array}{cc} I & 0 \\ 0 & A \end{array} \right] \end{array} \in \mathbb{R}^{n \times (n-m+r)}. \tag{3.10}$$

The fact that we must have

$$\begin{array}{c c} & \begin{array}{cc} n-m & r \end{array} \\ \begin{array}{c} n-m \\ r \end{array} & \left[\begin{array}{cc} Y & X \\ X^T & I \end{array} \right] \end{array} \in \mathcal{S}_+^{n-m+r}$$

follows from $\bar{Y} \succeq 0$ and the assumption that A has full column rank (hence U_A has full column rank). Therefore, we obtain the following reduced problem:

$$\begin{array}{rl} \text{find} & Z \in \mathcal{S}_+^{n-m+r} \\ \text{such that} & H \circ \mathcal{K}\left(U_A Z U_A^T\right) = H \circ D \\ & Z_{22} = I, \end{array} \tag{3.11}$$

where Z is partitioned as

$$Z =: \begin{array}{c c} & \begin{array}{cc} n-m & r \end{array} \\ \begin{array}{c} n-m \\ r \end{array} & \left[\begin{array}{cc} Z_{11} & Z_{12} \\ Z_{12}^T & Z_{22} \end{array}\right] \end{array}. \tag{3.12}$$

Since $Y - XX^T$ is the *Schur complement* of the matrix

$$\begin{bmatrix} Y & X \\ X^T & I \end{bmatrix}$$

with respect to the positive definite identity block, we have that

$$Y \succeq XX^T \quad \Leftrightarrow \quad \begin{bmatrix} Y & X \\ X^T & I \end{bmatrix} \succeq 0.$$

Therefore, we have a choice of a larger linear semidefinite constraint, or a smaller quadratic semidefinite constraint. See Chua and Tunçel (2008) for a theoretical discussion on the barriers associated with these two represenations; see Ding et al. (2008a) and Ding et al. (2008b) for a numerical comparison.

3.1.4 The Biswas-Ye formulation

We now present the Biswas-Ye formulation Biswas and Ye (2004); Biswas et al. (2006); Biswas and Ye (2006); Biswas (2007); Biswas et al. (2008) of the semidefinite relaxation of the sensor network localization problem. First we let $Y := XX^T$ be the Gram matrix of the rows of the matrix $X \in \mathbb{R}^{(n-m)\times r}$. Letting $e_{ij} \in \mathbb{R}^{n-m}$ be the vector with 1 in the i^{th} position, -1 in the j^{th} position, and zero elsewhere, we have

$$\begin{aligned} \|x_i - x_j\|^2 &= x_i^T x_i + x_j^T x_j - 2x_i^T x_j \\ &= e_{ij}^T X X^T e_{ij} \\ &= \left\langle e_{ij} e_{ij}^T, Y \right\rangle, \end{aligned}$$

for all $i, j = 1, \ldots, n-m$. Furthermore, letting $e_i \in \mathbb{R}^{n-m}$ be the vector with 1 in the i^{th} position, and zero elsewhere. Then

$$\begin{aligned} \|x_i - a_j\|^2 &= x_i^T x_i + a_j^T a_j - 2x_i^T a_j \\ &= \begin{bmatrix} e_i \\ a_j \end{bmatrix}^T \begin{bmatrix} X \\ I \end{bmatrix} \begin{bmatrix} X \\ I \end{bmatrix}^T \begin{bmatrix} e_i \\ a_j \end{bmatrix} \\ &= \left\langle \begin{bmatrix} e_i \\ a_j \end{bmatrix} \begin{bmatrix} e_i \\ a_j \end{bmatrix}^T, \begin{bmatrix} Y & X \\ X^T & I \end{bmatrix} \right\rangle, \end{aligned}$$

for all $i = 1, \ldots, n-m$, and $j = n-m+1, \ldots, n$, where we are now considering the anchors a_j to be indexed by $j \in \{n-m+1, \ldots, n\}$. Let $G = (N, E)$ be the graph corresponding to the partial Euclidean distance matrix D. Let

$$E_x := \{ij \in E : 1 \leq i < j \leq n-m\}$$

be the set of edges between sensors, and let

$$E_a := \{ij \in E : 1 \leq i \leq n-m,\ n-m+1 \leq j \leq n\}$$

be the set of edges between sensors and anchors. The sensor network localization problem can then be stated as:

$$\begin{array}{ll} \text{find} & X \in \mathbb{R}^{(n-m)\times r} \\ \text{such that} & \left\langle \begin{bmatrix} e_{ij} \\ 0 \end{bmatrix} \begin{bmatrix} e_{ij} \\ 0 \end{bmatrix}^T, Z \right\rangle = D_{ij}, \quad \forall ij \in E_x \\ & \left\langle \begin{bmatrix} e_i \\ a_j \end{bmatrix} \begin{bmatrix} e_i \\ a_j \end{bmatrix}^T, Z \right\rangle = D_{ij}, \quad \forall ij \in E_a \\ & Z = \begin{bmatrix} Y & X \\ X^T & I \end{bmatrix} \\ & Y = XX^T. \end{array} \tag{3.13}$$

The Biswas-Ye semidefinite relaxation of the sensor network localization problem is then formed by relaxing the hard constraint $Y = XX^T$ to the convex constraint $Y \succeq XX^T$. As mentioned above, $Y \succeq XX^T$ is equivalent to

$$\begin{bmatrix} Y & X \\ X^T & I \end{bmatrix} \succeq 0.$$

Therefore, the Biswas-Ye relaxation is the *linear* semidefinite optimization problem

$$\begin{array}{ll} \text{find} & Z \in \mathcal{S}_+^{n-m+r} \\ \text{such that} & \left\langle \begin{bmatrix} e_{ij} \\ 0 \end{bmatrix} \begin{bmatrix} e_{ij} \\ 0 \end{bmatrix}^T, Z \right\rangle = D_{ij}, \quad \forall ij \in E_x \\ & \left\langle \begin{bmatrix} e_i \\ a_j \end{bmatrix} \begin{bmatrix} e_i \\ a_j \end{bmatrix}^T, Z \right\rangle = D_{ij}, \quad \forall ij \in E_a \\ & Z_{22} = I, \end{array} \tag{3.14}$$

where Z is partitioned as in equation (3.12). Clearly, we have that the Biswas-Ye formulation (3.14) is equivalent to the Euclidean distance matrix formulation (3.11).

3.1.5 Unique localizability

The sensor network localization problem (3.1)/(3.13) is called *uniquely localizable* if there is a unique solution $X \in \mathbb{R}^{(n-m)\times r}$ for problem (3.1)/(3.13) and if $\bar{X} \in$

$\mathbb{R}^{(n-m)\times h}$ is a solution to the problem with anchors $\bar{A} := \begin{bmatrix} A & 0 \end{bmatrix} \in \mathbb{R}^{m\times h}$, then $\bar{X} = \begin{bmatrix} X & 0 \end{bmatrix}$. The following theorem from So and Ye (2007) shows that the semidefinite relaxation is tight if and only if the sensor network localization problem is uniquely localizable.

Theorem 3.4 ((So and Ye, 2007, Theorem 2)). *Let $A \in \mathbb{R}^{m\times r}$ such that $\begin{bmatrix} A & e \end{bmatrix}$ has full column rank. Let D be an n-by-n partial Euclidean distance matrix satisfying $D_{22} = \mathcal{K}(AA^T)$, with corresponding graph G and* 0–1 *adjacency matrix H. If G is connected, the following are equivalent.*

1. *The sensor network localization problem* (3.1)/(3.13) *is uniquely localizable.*
2. *The max-rank solution of the relaxation* (3.11)/(3.14) *has rank r.*
3. *The solution matrix Z of the relaxation* (3.11)/(3.14) *satisfies $Y = XX^T$, where*
$$Z = \begin{bmatrix} Y & X \\ X^T & I \end{bmatrix}.$$

□

Therefore, Theorem 3.4 implies that we can solve *uniquely localizable* instances of the sensor network localization problem in polynomial time by solving the semidefinite relaxation (3.11)/(3.14). However, it is important to point out an instance of the sensor network localization problem (3.1)/(3.13) which has a unique solution in $\mathbb{R}^r$ need not be uniquely localizable. This is especially important to point out in light of the complexity result in Aspnes et al. (2004) and Aspnes et al. (2006) in which it is proved that there is no efficient algorithm to solve instances of the sensor network localization problem having a unique solution in $\mathbb{R}^r$, unless $RP = NP$. This means we have two types of sensor network localization problem instances that have a unique solution in $\mathbb{R}^r$: (i) uniquely localizable, having no non-congruent solution in a higher dimension; (ii) not uniquely localizable, having a non-congruent solution in a higher dimension. Type (i) instances can be solved in polynomial time. Type (ii) cannot be solved in polynomial time, unless RP = NP.

3.1.6 Obtaining sensor positions from the semidefinite relaxation

Often it can be difficult to obtain a low rank solution from the semidefinite relaxation of a combinatorial optimization problem. An example of a successful semidefinite rounding technique is the impressive result in Goemans and Williamson (1995) for the Max-Cut problem; however, this is not always possible. For the sensor network localization problem we must obtain sensor positions from a solution of the semidefinite relaxation.

In the case of the Biswas-Ye formulation (3.14), or equivalently the Euclidean distance matrix formulation (3.11), the sensor positions $X \in \mathbb{R}^{(n-m)\times r}$ are obtained from a solution $Z \in \mathcal{S}_+^{n-m+r}$ by letting $X := Z_{12}$, where Z is partitioned as in equation (3.12). By Proposition 3.3, under the constraints that $\bar{Y} \succeq 0$ and $\bar{Y}_{22} = AA^T$, we may also compute the sensor positions as $X := \bar{Y}_{12}A^{\dagger T}$. Clearly, these are equivalent methods for computing the sensor positions.

Another method for computing the sensor positions was briefly discussed in Section 3.1.1 for the uniquely localizable case when $\operatorname{rank}(\bar{Y}) = r$. Now suppose that $\operatorname{rank}(\bar{Y}) > r$. In this case we find a best rank-r approximation of $\bar{Y}$. For this, we turn to the classical result of Eckart and Young (1936).

Theorem 3.5 ((Björck, 1996, Theorem 1.2.3)). *Let $A \in \mathbb{R}^{m\times n}$ and $k := \operatorname{rank}(A)$. Let*

$$A = U\Sigma V^T = \sum_{i=1}^{k} \sigma u_i v_i^T$$

be the singular value decomposition of A. Then the unique optimal solution of

$$\min\left\{\|A - X\|_F : \operatorname{rank}(X) = r\right\}$$

is given by

$$X := \sum_{i=1}^{r} \sigma u_i v_i^T,$$

with $\|A - X\|_F^2 = \sum_{i=r+1}^{k} \sigma_i^2$. □

For a symmetric matrix A, we can use the eigenvalue decomposition $A = UDU^T$ instead of the singular value decomposition; this is because the singular values of A are equal to the absolute values of the eigenvalues of A, so the singular value decomposition is $A = U\Sigma V^T$, where $\Sigma = |D|$ and V is computed from U by multiplying appropriate columns by -1. When $A \succeq 0$, its singular value decomposition and eigenvalue decomposition are identical.

Comparing two methods

Let $A \in \mathbb{R}^{m\times r}$ and D be an n-by-n partial Euclidean distance matrix with

$$D = \begin{bmatrix} D_{11} & D_{12} \\ D_{12}^T & \mathcal{K}(AA^T) \end{bmatrix}.$$

Let H be the 0–1 adjacency matrix corresponding to D. Suppose D has a completion $\bar{D} \in \mathcal{E}^n$ having $\operatorname{embdim}(\bar{D}) = r$. Suppose Z is a feasible solution of the semidefinite relaxation (3.14) (or equivalently, the Euclidean distance matrix formulation (3.11)).

Using a path-following interior-point method to find Z can result in a solution with high rank. Indeed, Goldfarb and Scheinberg (1998) show that for semidefinite optimization problems having strict complementarity, the central path converges to the analytic centre of the optimal solution set (that is, the optimal solution with maximum determinant); also see, for example, Güler and Ye (1993) for a similar result in linear optimization. However, it is also demonstrated in Halická et al. (2002) that there are examples of semidefinite optimization problems without strict complementarity having a central path converging to an optimal solution that is not the analytic centre of the optimal set.

Let $\bar{Y} := U_A Z U_A^T$, where U_A is as defined in equation (3.10). Suppose that $k := \operatorname{rank}(\bar{Y}) > r$. Then $\mathcal{K}(\bar{Y})$ is a Euclidean distance matrix completion of the partial Euclidean distance matrix D, with

$$\operatorname{embdim} \mathcal{K}(\bar{Y}) = k.$$

Moreover, $\bar{Y} \in \mathcal{S}_+^n$ and $\bar{Y}_{22} = AA^T$, so by Proposition 3.3, we have that

$$\bar{Y} = \begin{bmatrix} Y & XA^T \\ AX^T & AA^T \end{bmatrix},$$

where $Y := Z_{11} = \bar{Y}_{11}$ and $X := Z_{12} = \bar{Y}_{12} A^{\dagger T}$. Let $\bar{Y}_r \in \mathcal{S}_+^n$ be the nearest rank-r matrix to $\bar{Y} \in \mathcal{S}_+^n$, in the sense of Theorem 3.5. Let $\bar{P} \in \mathbb{R}^{n \times k}$ and $\bar{P}_r \in \mathbb{R}^{n \times r}$ such that

$$\bar{Y} = \bar{P}\bar{P}^T \qquad \text{and} \qquad \bar{Y}_r = \bar{P}_r \bar{P}_r^T.$$

Let $\bar{P}$ and $\bar{P}_r$ be partitioned as

$$\bar{P} =: \begin{array}{cc} & \begin{array}{cc} r & k-r \end{array} \\ \begin{array}{c} n-m \\ m \end{array} & \begin{bmatrix} \bar{P}_{11} & \bar{P}_{12} \\ \bar{P}_{12}^T & \bar{P}_{22} \end{bmatrix} \end{array} \qquad \text{and} \qquad \bar{P}_r =: \begin{array}{cc} & r \\ \begin{array}{c} n-m \\ m \end{array} & \begin{bmatrix} \bar{P}_r' \\ \bar{P}_r'' \end{bmatrix} \end{array}.$$

For the first method, we have that

$$\begin{bmatrix} \bar{P}_{12}^T & \bar{P}_{22} \end{bmatrix} \begin{bmatrix} \bar{P}_{12}^T & \bar{P}_{22} \end{bmatrix}^T = \bar{Y}_{22} = AA^T = \begin{bmatrix} A & 0 \end{bmatrix} \begin{bmatrix} A & 0 \end{bmatrix}^T.$$

Therefore, by Proposition 3.2, there exists an orthogonal matrix $\bar{Q} \in \mathbb{R}^{k \times k}$ such that

$$\begin{bmatrix} \bar{P}_{12}^T & \bar{P}_{22} \end{bmatrix} \bar{Q} = \begin{bmatrix} A & 0 \end{bmatrix}.$$

Let $\hat{P} := \bar{P}\bar{Q}$ and define the partition

$$\hat{P} =: \begin{array}{cc} & \begin{array}{cc} r & k-r \end{array} \\ \begin{array}{c} n-m \\ m \end{array} & \begin{bmatrix} \hat{P}_{11} & \hat{P}_{12} \\ A & 0 \end{bmatrix} \end{array}.$$

Therefore, we see that the semidefinite relaxation of the sensor network localization problem has allowed the sensors to move into the higher dimension of $\mathbb{R}^k$ instead of fixing the sensors to the space $\mathbb{R}^r$. Now we have that

$$\bar{Y} = \hat{P}\hat{P}^T = \begin{bmatrix} \hat{P}_{11}\hat{P}_{11}^T + \hat{P}_{12}\hat{P}_{12}^T & \hat{P}_{11}A^T \\ A\hat{P}_{11}^T & AA^T \end{bmatrix},$$

implying that $\hat{P}_{11} = X$, and $Y = XX^T + \hat{P}_{12}\hat{P}_{12}^T$, so

$$Y - XX^T = \hat{P}_{12}\hat{P}_{12}^T \succeq 0.$$

Therefore, we have that

$$\left\| \bar{Y} - \begin{bmatrix} X \\ A \end{bmatrix} \begin{bmatrix} X \\ A \end{bmatrix}^T \right\|_F = \left\| \hat{P}_{12}\hat{P}_{12}^T \right\|_F = \left\| Y - XX^T \right\|_F .$$

Since $\operatorname{rank}(PP^T) = r$, by the definition of $\bar{P}_r$ we have that

$$\left(\sum_{i=r+1}^{k} \lambda_i^2(\bar{Y}) \right)^{1/2} = \left\| \bar{Y} - \begin{bmatrix} \bar{P}_r' \\ \bar{P}_r'' \end{bmatrix} \begin{bmatrix} \bar{P}_r' \\ \bar{P}_r'' \end{bmatrix}^T \right\|_F \leq \left\| \bar{Y} - \begin{bmatrix} X \\ A \end{bmatrix} \begin{bmatrix} X \\ A \end{bmatrix}^T \right\|_F , \tag{3.15}$$

with equality if and only if

$$\begin{bmatrix} \bar{P}_r' \\ \bar{P}_r'' \end{bmatrix} \begin{bmatrix} \bar{P}_r' \\ \bar{P}_r'' \end{bmatrix}^T = \begin{bmatrix} X \\ A \end{bmatrix} \begin{bmatrix} X \\ A \end{bmatrix}^T ;$$

in other words, equality holds in (3.15) if and only if there exists $Q \in \mathbb{R}^{r\times r}$ orthogonal such that

$$\begin{bmatrix} \bar{P}_r' \\ \bar{P}_r'' \end{bmatrix} Q = \begin{bmatrix} X \\ A \end{bmatrix} .$$

For the second method, we let $Q_r \in \mathbb{R}^{r\times r}$ be an orthogonal matrix that minimizes $\left\| \bar{P}_r'' Q_r - A \right\|_F$. Since we may not have $\bar{P}_r''\bar{P}_r''^T = AA^T$, it is possible that

$$\left\| \bar{P}_r'' Q_r - A \right\|_F > 0.$$

Let $\bar{P}_r Q_r$ be partitioned as

$$\bar{P}_r Q_r =: \begin{array}{c} \\ n-m \\ n \end{array} \begin{array}{c} r \\ \left[\begin{array}{c} \bar{X} \\ \bar{A} \end{array} \right] \end{array} .$$

Therefore, we have the following relationship between the two different approaches for computing sensor positions:

$$\left\| \bar{Y} - \begin{bmatrix} \bar{X} \\ \bar{A} \end{bmatrix} \begin{bmatrix} \bar{X} \\ \bar{A} \end{bmatrix}^T \right\|_F \leq \left\| \bar{Y} - \begin{bmatrix} X \\ A \end{bmatrix} \begin{bmatrix} X \\ A \end{bmatrix}^T \right\|_F ,$$

with equality if and only if there exists a $Q \in \mathbb{R}^{r\times r}$ such that

$$\begin{bmatrix} \bar{X} \\ \bar{A} \end{bmatrix} Q = \begin{bmatrix} X \\ A \end{bmatrix} .$$

The second method can be seen getting a better approximation of $\bar{Y}$ at the expense of allowing the anchors positions to move slightly.

Note that we have no guarantee that X from the first method, or $\bar{X}$ from the second method, satisfy the distance constraints. That is, we may have

$$H \circ \mathcal{K}\left(PP^T\right) \neq H \circ D,$$

for

$$P = \begin{bmatrix} X \\ A \end{bmatrix} \qquad \text{or} \qquad P = \begin{bmatrix} \bar{X} \\ \bar{A} \end{bmatrix}.$$

Moreover, we have the following bounds on the approximation of the Euclidean distance matrix $\mathcal{K}\left(\bar{Y}\right)$:

$$\begin{aligned}
\left\| \mathcal{K}\left(\bar{Y}\right) - \mathcal{K}\left(\begin{bmatrix} X \\ A \end{bmatrix} \begin{bmatrix} X \\ A \end{bmatrix}^T \right) \right\|_F &\leq \|\mathcal{K}\|_F \left\| \bar{Y} - \begin{bmatrix} X \\ A \end{bmatrix} \begin{bmatrix} X \\ A \end{bmatrix}^T \right\|_F \\
&= 2\sqrt{n} \left\| Y - XX^T \right\|_F ; \qquad (3.16) \\
\left\| \mathcal{K}\left(\bar{Y}\right) - \mathcal{K}\left(\begin{bmatrix} \bar{X} \\ \bar{A} \end{bmatrix} \begin{bmatrix} \bar{X} \\ \bar{A} \end{bmatrix}^T \right) \right\|_F &\leq \|\mathcal{K}\|_F \left\| \bar{Y} - \begin{bmatrix} \bar{X} \\ \bar{A} \end{bmatrix} \begin{bmatrix} \bar{X} \\ \bar{A} \end{bmatrix}^T \right\|_F \\
&= 2\sqrt{n} \left(\sum_{i=r+1}^{k} \lambda_i^2(\bar{Y}) \right)^{1/2}. \qquad (3.17)
\end{aligned}$$

Here we have used the fact from Lemma 2.39 that $\|\mathcal{K}\|_F = 2\sqrt{n}$. Clearly, the upper bound (3.17) for the second method is lower than upper bound (3.16) for the first method, but this need not imply that the second method gives a better r-dimensional approximation of the k-dimensional Euclidean distance matrix $\mathcal{K}(\bar{Y})$. Indeed, we ran numerical tests in Ding et al. (2008b) to compare these two methods. We often found better results using $\bar{X}$ than when using X, but this was not always the case.

3.2 Molecular conformation

While the sensor network localization problem we have just described is a low-dimensional Euclidean distance matrix completion problem with anchor nodes, the molecular conformation problem can be considered as a regular *anchor-free* low-dimensional Euclidean distance matrix completion problem for which we would like to compute relative positions of the points in the low-dimensional space $\mathbb{R}^3$.

Nuclear magnetic resonance (NMR) can be used to measure interatomic distances in the range of 5–6Å, where 1Å $= 10^{-10}$m; longer ranges of up to 8Å are possible, but come at a cost of structural modification of the molecule Yuen et al. (2010). Using the partial distance information obtained by NMR, we would like to compute relative atom positions within the molecule. Our approach to the molecular conformation problem is simple: we treat it exactly as a low-dimensional Euclidean distance matrix completion problem (2.14) and solve the corresponding semidefinite relaxation (2.17) using the facial reduction techniques and algorithm we develop in Chapter 4. While this simple approach is reasonable, it may also be desirable to incorporate inequality constraints for flexible atomic bonds, or even angle constraints into the model; however, we only consider simple distance equality constraints here and note that this is a topic of ongoing research. See Section 5.2 for the results of our numerical tests on protein data obtained from the Protein Data Bank Berman et al. (2002).

Chapter 4

Facial Reduction for the Euclidean Distance Matrix Completion Problem

When strict feasibility fails, the Supporting Hyperplane Theorem 2.9 tells us that there exists an exposed proper face of the cone containing your feasible set. When this face is the minimal face, Proposition 2.13 implies that we obtain a strictly feasible problem by restricting our problem to this face. Before discussing facial reduction for the semidefinite relaxation of the Euclidean distance matrix completion problem,

$$\begin{array}{rc} \text{find} & Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C \\ \text{subject to} & H \circ \mathcal{K}(Y) = H \circ D \end{array},$$

or the regularized semidefinite optimization problem (see, for example, Weinberger et al. (2004) or Biswas et al. (2006)),

$$\begin{array}{rc} \text{maximize} & \operatorname{trace}(Y) \\ \text{subject to} & H \circ \mathcal{K}(Y) = H \circ D \\ & Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C \end{array},$$

we first consider facial reduction on general linear optimization and semidefinite optimization problems.

4.1 Facial reduction in linear optimization

First we look at the case of facial reduction on the *linear optimization (LO)* problem:

$$\begin{array}{rl} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \in \mathbb{R}^n_+ \end{array}, \tag{4.1}$$

where $A \in \mathbb{R}^{m\times n}$ has full row rank, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$. Faces of the cone $\mathbb{R}^n_+$ are giving by

$$F = \left\{x \in \mathbb{R}^n_+ : x_i = 0, \forall i \in \mathcal{I}\right\},$$

for some $\mathcal{I} \subseteq 1{:}n$. As in Theorem 2.21, we have that if problem (4.1) is feasible, then problem (4.1) is not strictly feasible if and only if there exists $y \in \mathbb{R}^m$ such that

$$\begin{aligned} b^T y &= 0, \\ 0 \neq A^T y &\geq 0. \end{aligned} \tag{4.2}$$

Indeed, if $y \in \mathbb{R}^m$ satisfies conditions (4.2), then for all x feasible we have

$$\left(A^T y\right) x = b^T y = 0,$$

implying that $x_i = 0$ for all $i \in 1{:}n$ such that $\left(A^T y\right)_i \neq 0$. Therefore, if condition (4.2) holds, then

$$\left\{x \in \mathbb{R}^n_+ : Ax = b\right\} \subseteq \left\{x \in \mathbb{R}^n_+ : x_i = 0,\ \forall i \in \mathcal{I}\right\},$$

where $\mathcal{I} := \operatorname{supp}(A^T y) = \left\{i : (A^T y)_i \neq 0\right\}$. In this case, we have

$$\operatorname{face}\left\{x \in \mathbb{R}^n_+ : Ax = b\right\} \trianglelefteq \left\{x \in \mathbb{R}^n_+ : x_i = 0,\ \forall i \in \mathcal{I}\right\}.$$

Let $\mathcal{J} := (1{:}n) \setminus \mathcal{I}$ and $t := |\mathcal{J}|$. Then problem (4.1) reduces to

$$\begin{array}{ll} \text{minimize} & c[\mathcal{J}]^T \bar{x} \\ \text{subject to} & A[:, \mathcal{J}]\bar{x} = b[\mathcal{J}] \\ & \bar{x} \in \mathbb{R}^t_+ \end{array}. \tag{4.3}$$

Note that in problem (4.3), $A[:, \mathcal{J}] \in \mathbb{R}^{m\times t}$ may not have full row rank. This implies that we can also reduce the number of constraints by removing linearly dependent rows, leaving us with $\bar{A} \in \mathbb{R}^{k\times t}$ and $\bar{b} \in \mathbb{R}^k$. Letting $\bar{c} := c[\mathcal{J}] \in \mathbb{R}^t$, we now have the equivalent reduced problem

$$\begin{array}{ll} \text{minimize} & \bar{c}^T \bar{x} \\ \text{subject to} & \bar{A}\bar{x} = \bar{b} \\ & \bar{x} \in \mathbb{R}^t_+ \end{array}. \tag{4.4}$$

If the face we identified was minimal, then problem (4.4) is also strictly feasible.

The technique discussed here is reminiscent of the linear optimization preprocessing techniques discussed in, for example, Brearley et al. (1975), Andersen and Andersen (1995), and more recently in Mészáros and Suhl (2003).

4.2 Facial reduction in semidefinite optimization

In the case of *semidefinite optimization (SDO)*,

$$\begin{array}{ll} \text{minimize} & \langle C, X\rangle \\ \text{subject to} & \langle A_i, X\rangle = b_i,\ \forall i = 1,\dots,m \\ & X \in \mathcal{S}^n_+ \end{array}, \tag{4.5}$$

where $A_i \in \mathcal{S}^n$, for $i = 1,\dots,m$, are linearly independent, $b \in \mathbb{R}^m$, and $C \in \mathcal{S}^n$. Let $U \in \mathbb{R}^{n\times t}$ have full column rank and satisfy

$$\operatorname{face}\left\{X \in \mathcal{S}^n_+ : \langle A_i, X\rangle = b_i,\ \forall i\right\} = U\mathcal{S}^t_+U^T.$$

Then problem (4.5) is equivalent to the strictly feasible problem

$$\begin{array}{ll} \text{minimize} & \left\langle C, UZU^T\right\rangle \\ \text{subject to} & \left\langle A_i, UZU^T\right\rangle = b_i,\ \forall i = 1,\dots,m \\ & Z \in \mathcal{S}^t_+ \end{array},$$

which, in turn, is equivalent to

$$\begin{array}{ll} \text{minimize} & \left\langle \bar{C}, Z\right\rangle \\ \text{subject to} & \left\langle \bar{A}_i, Z\right\rangle = b_i,\ \forall i = 1,\dots,m \\ & Z \in \mathcal{S}^t_+ \end{array}, \tag{4.6}$$

where $\bar{A}_i := U^TA_iU \in \mathcal{S}^t$, for $i = 1,\dots,m$, and $\bar{C} := U^TCU \in \mathcal{S}^t$. A very nice example of this type of reduction can be seen in (Tunçel, 2001, Section 3.1) for semidefinite relaxations of nonconvex sets which are not full dimensional; as we see here, the reduction in Tunçel (2001) has the effect of producing a semidefinite relaxation that is strictly feasible.

Let $\mathcal{A}\colon \mathcal{S}^n \to \mathbb{R}^m$ be defined by $\mathcal{A}X := \left(\left\langle \bar{A}_i, Z\right\rangle\right)_{i=1}^m$. As seen in Theorem 2.21, if problem (4.5) is feasible, then problem (4.5) is not strictly feasible if and only if there exists some $y \in \mathbb{R}^m$ such that $0 \neq \mathcal{A}^*y \succeq 0$ and $b^Ty = 0$. Moreover, if $y \in \mathbb{R}^m$ such that $0 \neq \mathcal{A}^*y \succeq 0$ and $b^Ty = 0$, then

$$\operatorname{face}\left\{X \in \mathcal{S}^n_+ : \mathcal{A}X = b\right\} \trianglelefteq \mathcal{S}^n_+ \cap \left\{\mathcal{A}^*y\right\}^\perp = U\mathcal{S}^t_+U^T \triangleleft \mathcal{S}^n_+,$$

where $U \in \mathbb{R}^{n\times t}$ has full column rank and satisfies $\operatorname{range}(U) = \operatorname{null}(\mathcal{A}^*y)$.

As in the case of linear optimization, we may now have that the set of matrices $\left\{\bar{A}_i : i = 1,\dots,m\right\}$ is linearly dependent. Therefore, we can determine a maximal subset $\mathcal{I} \subseteq 1{:}m$ such that $\left\{\bar{A}_i : i \in \mathcal{I}\right\}$ is linearly independent. Letting $k := |\mathcal{I}|$ and $\bar{b} := b[\mathcal{I}] \in \mathbb{R}^k$, we have

$$\begin{array}{ll} \text{minimize} & \left\langle \bar{C}, Z\right\rangle \\ \text{subject to} & \bar{\mathcal{A}}(Z) = \bar{b} \\ & Z \in \mathcal{S}^t_+ \end{array}, \tag{4.7}$$

where $\bar{\mathcal{A}}\colon \mathcal{S}^t \to \mathbb{R}^k$ is defined by $\bar{\mathcal{A}}(Z) := \left(\left\langle \bar{A}_i, Z\right\rangle\right)_{i\in\mathcal{I}}$.

More information on semidefinite facial reduction can by found in, for example, Ramana et al. (1997) and Cheung et al. (2010).

4.3 Single clique facial reduction

Recall that the feasible set of the semidefinite relaxation of the Euclidean distance matrix problem is given by

$$\mathcal{F} := \left\{ Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C : H \circ \mathcal{K}(Y) = H \circ D \right\},$$

where D is an n-by-n partial Euclidean distance matrix, and H is the corresponding 0–1 adjacency matrix.

For a general semidefinite optimization problem, it is generally not possible to easily identify the minimal face containing the feasible set. However, because of the structure available to us, we now show how we can identify the minimal face containing the semidefinite matrices satisfying a subset of the constraints corresponding to a clique in the graph.

Theorem 4.1 (Single Clique Facial Reduction Theorem)**.** *Let D be an n-by-n partial Euclidean distance matrix and let $C \subseteq 1{:}n$ be a clique in the corresponding graph. Let $t := \operatorname{embdim}(D[C])$. Assume, without loss of generality, that $C = 1{:}k$, and let $U \in \mathbb{R}^{n \times (n-|C|+t+1)}$ be defined as follows:*

- *let $U_C \in \mathbb{R}^{|C| \times t}$ have full column rank and satisfy $\operatorname{range}(U_C) = \operatorname{range}(\mathcal{K}^\dagger(D[C]))$;*
- *let $\bar{U} := \begin{bmatrix} U_C & e \end{bmatrix} \in \mathbb{R}^{|C| \times (t+1)}$;*
- *let $U := \begin{array}{c} \\ {\scriptstyle |C|} \\ {\scriptstyle n-|C|} \end{array} \begin{array}{c} \begin{array}{cc} {\scriptstyle t+1} & {\scriptstyle n-|C|} \end{array} \\ \left[\begin{array}{cc} \bar{U} & 0 \\ 0 & I \end{array} \right] \end{array} \in \mathbb{R}^{n \times (n-|C|+t+1)}$.*

Then U has full column rank, $e \in \operatorname{range}(U)$, and

$$\operatorname{face}\left\{ Y \in \mathcal{S}^n_+ : \mathcal{K}(Y[C]) = D[C] \right\} = U \mathcal{S}^{n-|C|+t+1}_+ U^T.$$

Proof. First we note that since $\mathcal{K}^\dagger(D[C]) \succeq 0$ and $\operatorname{range}(U_C) = \operatorname{range}(\mathcal{K}^\dagger(D[C]))$, there exists some $\Sigma_C \in \mathcal{S}^t_{++}$ such that

$$\mathcal{K}^\dagger(D[C]) = U_C \Sigma_C U_C^T.$$

Since $\mathcal{K}^\dagger(D[C]) \in \mathcal{S}^{|C|}_C$, we have $U_C^T e = 0$. Therefore, $e \notin \operatorname{range}(U_C)$, so $\bar{U}$ has full column rank, implying that U has full column rank. Moreover, $e \in \operatorname{range}(U)$ clearly holds. Let $\bar{V} \in \mathbb{R}^{|C| \times (|C|-t-1)}$ have full column rank and satisfy $\operatorname{range}(\bar{V}) = \operatorname{range}(\bar{U})^\perp = \operatorname{null}(\bar{U}^T)$; in particular, note that $\bar{V}^T e = 0$, since $e \in \operatorname{range}(\bar{U})$ and $\operatorname{range}(\bar{U}) = \operatorname{null}(\bar{V}^T)$. Let

$$V := \begin{array}{c} \\ {\scriptstyle |C|} \\ {\scriptstyle n-|C|} \end{array} \begin{array}{c} {\scriptstyle |C|-t-1} \\ \left[\begin{array}{c} \bar{V} \\ 0 \end{array} \right] \end{array} \in \mathbb{R}^{n \times (|C|-t-1)}.$$

Then $\operatorname{range}(V) = \operatorname{range}(U)^\perp = \operatorname{null}(U^T)$. Now we will show that $VV^T \in F^c$, where

$$F := \operatorname{face}\left\{Y \in \mathcal{S}^n_+ : \mathcal{K}(Y[C]) = D[C]\right\}.$$

The dual problem of

$$\begin{array}{cc} \text{minimize} & 0 \\ \text{subject to} & \mathcal{K}(Y[C]) = D[C] \\ & Y \succeq 0 \end{array}$$

is given by

$$\begin{array}{cc} \text{maximize} & -\langle \Lambda, D[C]\rangle \\ \text{subject to} & \begin{bmatrix} \mathcal{K}^*(\Lambda) & 0 \\ 0 & 0 \end{bmatrix} \succeq 0. \end{array}$$

As expected, Λ is dual feasible if and only if $\Lambda \in (\mathcal{E}^{|C|})^*$. Therefore, by Proposition 2.19, we have that

$$\operatorname{face}\left\{S \in \mathcal{S}^n_+ : S = \begin{bmatrix} \mathcal{K}^*(\Lambda) & 0 \\ 0 & 0 \end{bmatrix}, \langle \Lambda, D[C]\rangle = 0\right\} \trianglelefteq F^c.$$

Let $\Lambda := \mathcal{K}^{\dagger *}(\bar{V}\bar{V}^T)$. Then $\mathcal{K}^*(\Lambda) = \mathcal{K}^*\mathcal{K}^{\dagger *}(\bar{V}\bar{V}^T) = \bar{V}\bar{V}^T$, since $\mathcal{K}^*\mathcal{K}^{\dagger *}$ is the orthogonal projection onto $\operatorname{range}(\mathcal{K}^*) = \mathcal{S}^{|C|}_C$, and since $\bar{V}^T e = 0$. Moreover,

$$\begin{aligned} \langle \Lambda, D[C]\rangle &= \left\langle \mathcal{K}^{\dagger *}(\bar{V}\bar{V}^T), D[C]\right\rangle \\ &= \left\langle \bar{V}\bar{V}^T, \mathcal{K}^{\dagger}(D[C])\right\rangle \\ &= \left\langle \bar{V}\bar{V}^T, U_C\Sigma_C U_C^T\right\rangle \\ &= 0, \end{aligned}$$

since $\bar{V}^T U_C = 0$. Thus,

$$S := \begin{bmatrix} \mathcal{K}^*(\Lambda) & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \bar{V}\bar{V}^T & 0 \\ 0 & 0 \end{bmatrix} = VV^T \in F^c,$$

as claimed.

Since $VV^T \in F^c$, by Proposition 2.11, we have that

$$\operatorname{face}\left\{Y \in \mathcal{S}^n_+ : \mathcal{K}(Y[C]) = D[C]\right\} \trianglelefteq \mathcal{S}^n_+ \cap \left\{VV^T\right\}^\perp = U\mathcal{S}^{n-|C|+t+1}_+ U^T.$$

By Proposition 2.13, to show that

$$\operatorname{face}\left\{Y \in \mathcal{S}^n_+ : \mathcal{K}(Y[C]) = D[C]\right\} = U\mathcal{S}^{n-|C|+t+1}_+ U^T,$$

we simply must show that there exists $\bar{Y} \in \left\{Y \in \mathcal{S}^n_+ : \mathcal{K}(Y[C]) = D[C]\right\}$ such that $\bar{Y} = UZU^T$ for some $Z \in \mathcal{S}^{n-|C|+t+1}_{++}$. To achieve this, we begin by letting

$$\bar{\Xi} := \begin{bmatrix} \Sigma_C & 0 \\ 0 & 1 \end{bmatrix} \in \mathcal{S}^{t+1}_{++} \quad \text{and} \quad \bar{P} := \bar{U}\bar{\Xi}^{1/2} \in \mathbb{R}^{|C|\times(t+1)}.$$

Note that

$$\begin{aligned}\bar{P}\bar{P}^T &= \bar{U}\bar{\Xi}\bar{U}^T \\ &= \begin{bmatrix} U_C & e \end{bmatrix} \begin{bmatrix} \Sigma_C & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} U_C & e \end{bmatrix}^T \\ &= U_C \Sigma_C U_C^T + ee^T, \end{aligned}$$

implying that

$$\mathcal{K}\left(\bar{P}\bar{P}^T\right) = \mathcal{K}\left(U_C\Sigma_C U_C^T + ee^T\right) = \mathcal{K}\left(U_C\Sigma_C U_C^T\right) = \mathcal{K}\left(\mathcal{K}^\dagger(D[C])\right) = D[C].$$

Furthermore, let

$$P := \begin{array}{ccc} & t+1 & n-|C| \\ |C| & \bar{P} & 0 \\ n-|C| & 0 & I \end{array} \in \mathbb{R}^{n\times(n-|C|+t+1)} \quad \text{and} \quad \bar{Y} := PP^T.$$

Then

$$\bar{Y} = \begin{array}{ccc} & |C| & n-|C| \\ |C| & \bar{P}\bar{P}^T & 0 \\ n-|C| & 0 & I \end{array} = \begin{bmatrix} \bar{U}\bar{\Xi}\bar{U}^T & 0 \\ 0 & I \end{bmatrix} = UZU^T \in \mathcal{S}^n_+,$$

where Z is defined as

$$Z := \begin{array}{ccc} & t+1 & n-|C| \\ t+1 & \bar{\Xi} & 0 \\ n-|C| & 0 & I \end{array} \in \mathcal{S}^{n-|C|+t+1}_{++}.$$

Finally, note that $\mathcal{K}\left(\bar{Y}[C]\right) = \mathcal{K}\left(\bar{P}\bar{P}^T\right) = D[C]$, implying that

$$\bar{Y} \in \left\{Y \in \mathcal{S}^n_+ : \mathcal{K}(Y[C]) = D[C]\right\} \cap \operatorname{relint}\left(U\mathcal{S}^{n-|C|+t+1}_+ U^T\right),$$

so we have face $\left\{Y \in \mathcal{S}^n_+ : \mathcal{K}(Y[C]) = D[C]\right\} = U\mathcal{S}^{n-|C|+t+1}_+ U^T$. □

4.3.1 An alternate proof

Our alternative proof of Theorem 4.1 requires the following technical lemmas. We will also need these lemmas for results we prove in later sections. The first lemma is inspired by the following observation: if $Y \in \mathcal{S}^n_+$ and $D := \mathcal{K}(Y) \in \mathcal{E}^n$, then $\mathcal{K}(Y) = \mathcal{K}(\mathcal{K}^\dagger(D))$, so $Y \in \mathcal{K}^\dagger(D) + \operatorname{null}(\mathcal{K})$, implying that

$$Y = \mathcal{K}^\dagger(D) + \mathcal{D}_e(y), \quad \text{for some } y \in \mathbb{R}^n.$$

If $B := \mathcal{K}^\dagger(D)$, then $Be = 0$, and $Y = B + \mathcal{D}_e(y) \succeq 0$. However, a more general result is possible using the linear map $\mathcal{D}_v \colon \mathbb{R}^n \to \mathcal{S}^n$ defined by

$$\mathcal{D}_v(y) := yv^T + vy^T,$$

where $v \in \mathbb{R}^n$.

Lemma 4.2 (Range Lemma)**.** *Let $B \in \mathcal{S}^n$, $Bv = 0$, $v \neq 0$, $y \in \mathbb{R}^n$ and $Y := B + \mathcal{D}_v(y)$. If $Y \succeq 0$, then*

$$y \in \operatorname{range}(B) + \operatorname{cone}\{v\}$$

and

$$\operatorname{range}(Y) \subseteq \operatorname{range}\left(\begin{bmatrix} B & v \end{bmatrix}\right).$$

Proof. First suppose, for the sake of contradiction, that $y \notin \operatorname{range}(B) + \operatorname{span}\{v\} = \operatorname{range}\left(\begin{bmatrix} B & v \end{bmatrix}\right)$. Then y can be written as the orthogonal decomposition

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

where $0 \neq \bar{y} \in \operatorname{range}\left(\begin{bmatrix} B & v \end{bmatrix}\right)^\perp = \operatorname{null}\left(\begin{bmatrix} B & v \end{bmatrix}^T\right)$. Note that $\bar{y}$ satisfies

$$B\bar{y} = 0 \quad \text{and} \quad v^T\bar{y} = 0.$$

To get a contradiction with the assumption that $Y \succeq 0$, we let

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

and observe that:

$$\begin{aligned} Bz &= 0; \\ v^T z &= \frac{1}{2}; \\ \bar{y}^T z &= -(1 + |\beta|). \end{aligned}$$

Then

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

which gives us the desired contradiction. Therefore, $y \in \operatorname{range}(B) + \operatorname{span}\{v\}$.

To show that $y \in \operatorname{range}(B) + \operatorname{cone}\{v\}$, we suppose that $y = Bu + \beta v$, and aim to show that $\beta \geq 0$. Since $Bv = 0$, we have $v^T y = \beta v^T v$. Then,

$$\begin{aligned} v^T Y v &= v^T \mathcal{D}_v(y) v \\ &= v^T \left(y v^T + v y^T \right) v \\ &= 2 v^T y v^T v \\ &= 2\beta (v^T v)^2. \end{aligned}$$

Since $Y \succeq 0$, we have $2\beta(v^T v)^2 \geq 0$. This implies that $\beta \geq 0$, since $v \neq 0$.

Finally, since $y = Bu + \beta v$, we have

$$\begin{aligned} Y &= B + v y^T + y v^T \\ &= B + v u^T B + B u v^T + 2\beta v v^T \\ &= B\left(I + u v^T\right) + v\left(Bu + 2\beta v\right)^T \end{aligned}$$

so $\operatorname{range}(Y) \subseteq \operatorname{range}\left(\begin{bmatrix} B & v \end{bmatrix}\right)$. □

In the next lemma, we show that 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

$$\mathcal{S}^n_+(1{:}k, \bar{Y}) := \left\{ Y \in \mathcal{S}^n_+ : Y[1{:}k] = \bar{Y} \right\}.$$

Lemma 4.3 (Face Lemma)**.** *Let $\bar{U} \in \mathbb{R}^{k \times t}$ have full column rank, and let*

$$U := \begin{array}{cc} & \begin{array}{cc} t & n-k \end{array} \\ \begin{array}{c} k \\ n-k \end{array} & \begin{bmatrix} \bar{U} & 0 \\ 0 & I \end{bmatrix} \end{array}.$$

If $\operatorname{face}\left\{\bar{Y}\right\} \trianglelefteq \bar{U}\mathcal{S}^t_+\bar{U}^T$*, then*

$$\operatorname{face}\mathcal{S}^n_+(1{:}k, \bar{Y}) \trianglelefteq U \mathcal{S}^{n-k+t}_+ U^T.$$

Furthermore, if $\operatorname{face}\left\{\bar{Y}\right\} = \bar{U}\mathcal{S}^t_+\bar{U}^T$*, then*

$$\operatorname{face}\mathcal{S}^n_+(1{:}k, \bar{Y}) = U \mathcal{S}^{n-k+t}_+ U^T.$$

Proof. First, suppose that $\operatorname{face}\left\{\bar{Y}\right\} \trianglelefteq \bar{U}\mathcal{S}^t_+\bar{U}^T$. Since $\bar{Y} \in \bar{U}\mathcal{S}^t_+\bar{U}^T$, then $\bar{Y} = \bar{U}\bar{Z}\bar{U}^T$, for some $\bar{Z} \in \mathcal{S}^t_+$. Let $Y \in \mathcal{S}^n_+(1{:}k, \bar{Y})$ and choose $\bar{V} \in \mathbb{R}^{k \times (k-t)}$ with full column rank satisfying $\operatorname{range}(\bar{V}) = \operatorname{range}(\bar{U})^\perp = \operatorname{null}(\bar{U}^T)$; note that $\bar{V}^T \bar{Y} \bar{V} = 0$. Let

$$Y =: \begin{array}{cc} & \begin{array}{cc} k & n-k \end{array} \\ \begin{array}{c} k \\ n-k \end{array} & \begin{bmatrix} \bar{Y} & Y_{21}^T \\ Y_{21} & Y_{22} \end{bmatrix} \end{array} \quad \text{and} \quad V := \begin{array}{cc} & \begin{array}{cc} k-t & n-k \end{array} \\ \begin{array}{c} k \\ n-k \end{array} & \begin{bmatrix} \bar{V} & 0 \\ 0 & I \end{bmatrix} \end{array}.$$

Then

$$V^T Y V = \begin{bmatrix} 0 & \bar{V}^T Y_{21}^T \\ Y_{21}\bar{V} & Y_{22} \end{bmatrix} \succeq 0,$$

implying that $\bar{V}^T Y_{21}^T = 0$. Thus, $\mathrm{range}(Y_{21}^T) \subseteq \mathrm{null}(\bar{V}^T) = \mathrm{range}(\bar{V})^\perp = \mathrm{range}(\bar{U})$, so $Y_{21}^T = \bar{U}X$, for some $X \in \mathbb{R}^{t\times(n-k)}$. Therefore, we can write

$$Y = \begin{bmatrix} \bar{U} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \bar{Z} & X \\ X^T & Y_{22} \end{bmatrix} \begin{bmatrix} \bar{U} & 0 \\ 0 & I \end{bmatrix}^T =: UMU^T.$$

Since $Y \succeq 0$, we have $v^T M v \geq 0$, for all $v \in \mathrm{range}(U^T)$. However, $U \in \mathbb{R}^{n\times(n-k+t)}$ has full column rank, so $\mathrm{range}(U^T) = \mathbb{R}^{n-k+t}$. Therefore, $M \succeq 0$, implying that $Y \in U\mathcal{S}_+^{n-k+t}U^T$. Thus, $\mathcal{S}_+^n(1{:}k, \bar{Y}) \subseteq U\mathcal{S}_+^{n-k+t}U^T$, so

$$\mathrm{face}\mathcal{S}_+^n(1{:}k, \bar{Y}) \trianglelefteq U\mathcal{S}_+^{n-k+t}U^T.$$

Now suppose that $\mathrm{face}\left\{\bar{Y}\right\} = \bar{U}\mathcal{S}_+^t\bar{U}^T$. Then, by Proposition 2.13, $\bar{Y} \in \mathrm{relint}\left(\bar{U}\mathcal{S}_+^t\bar{U}^T\right)$, so $\bar{Y} = \bar{U}\bar{Z}\bar{U}^T$, for some $\bar{Z} \in \mathcal{S}_{++}^t$. Letting

$$\hat{Y} := \begin{bmatrix} \bar{U} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \bar{Z} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \bar{U} & 0 \\ 0 & I \end{bmatrix}^T = \begin{bmatrix} \bar{Y} & 0 \\ 0 & I \end{bmatrix},$$

we see that $\hat{Y} \in \mathcal{S}_+^n(1{:}k, \bar{Y}) \cap \mathrm{relint}\left(U\mathcal{S}_+^{n-k+t}U^T\right)$. Since $\mathcal{S}_+^n(1{:}k, \bar{Y}) \subseteq U\mathcal{S}_+^{n-k+t}U^T$, Proposition 2.13 implies that

$$\mathrm{face}\mathcal{S}_+^n(1{:}k, \bar{Y}) = U\mathcal{S}_+^{n-k+t}U^T. \qquad \square$$

We are now in a position to provide our alternative proof of the Single Clique Facial Reduction Theorem.

Alternate Proof of Theorem 4.1. Let $Y \in \mathcal{S}_+^n$ such that $\mathcal{K}(Y[C]) = D[C]$. Then $Y[C] \in \mathcal{K}^\dagger(D[C]) + \mathrm{null}(\mathcal{K}) = \mathcal{K}^\dagger(D[C]) + \mathrm{range}(\mathcal{D}_e)$. Therefore, there exists $y \in \mathbb{R}^{|C|}$ such that

$$Y[C] = \mathcal{K}^\dagger(D[C]) + \mathcal{D}_e(y).$$

Since $Y \succeq 0$, we have $Y[C] \succeq 0$. Thus, by Lemma 4.2,

$$\mathrm{range}(Y[C]) \subseteq \mathrm{range}\left(\begin{bmatrix} \mathcal{K}^\dagger(D[C]) & e \end{bmatrix}\right) = \mathrm{range}\left(\bar{U}\right).$$

Therefore, by Lemma 2.14, we have that $Y[C] \in \bar{U}\mathcal{S}_+^{t+1}\bar{U}^T$, implying that

$$\mathrm{face}\left\{Y[C]\right\} \trianglelefteq \bar{U}\mathcal{S}_+^{t+1}\bar{U}^T.$$

Since we are assuming that $C = 1{:}k$, by Lemma 4.3, we have

$$\mathrm{face}\mathcal{S}_+^n(C, Y[C]) \trianglelefteq U\mathcal{S}_+^{n-|C|+t+1}U^T,$$

so $Y \in U\mathcal{S}_+^{n-|C|+t+1}U^T$. Therefore,

$$\left\{Y \in \mathcal{S}_+^n : \mathcal{K}(Y[C]) = D[C]\right\} \subseteq U\mathcal{S}_+^{n-|C|+t+1}U^T,$$

implying that

$$\mathrm{face}\left\{Y \in \mathcal{S}_+^n : \mathcal{K}(Y[C]) = D[C]\right\} \trianglelefteq U\mathcal{S}_+^{n-|C|+t+1}U^T.$$

The remainder of the proof continues as in the original proof of Theorem 4.1. $\square$

4.4 Facial reduction algorithm overview

Let D be an n-by-n partial Euclidean distance matrix, let G be the corresponding graph, and let

$$\mathcal{F} := \left\{ Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C : H \circ \mathcal{K}(Y) = H \circ D \right\} \tag{4.8}$$

be the feasible set of the Euclidean distance matrix completion problem (2.13), where H is the 0–1 adjacency matrix of G. For each node $i = 1, \ldots, n$, let C_i be a clique in the graph G such that

$$i \in C_i, \quad \text{for all } i = 1, \ldots, n.$$

Note that $\cup_{i=1}^n C_i = 1{:}n$. For each clique C_i, let

$$\mathcal{F}_i := \left\{ Y \in \mathcal{S}^n_+ : \mathcal{K}(Y[C_i]) = D[C_i] \right\}.$$

Then

$$\mathcal{F} \subseteq \left(\bigcap_{i=1}^n \mathcal{F}_i \right) \cap \mathcal{S}^n_C.$$

For $i = 1, \ldots, n$, let

$$F_i := \text{face}(\mathcal{F}_i), \quad t_i := \text{embdim}(D[C_i]),$$

and, as in Theorem 4.1, let $U_i \in \mathbb{R}^{n \times (n-|C_i|+t_i+1)}$ with full column rank satisfy $e \in \text{range}(U_i)$ and

$$F_i = U_i \mathcal{S}_+^{n-|C_i|+t_i+1} U_i^T.$$

Therefore, we have

$$\begin{aligned} \text{face}(\mathcal{F}) &\trianglelefteq \left(\bigcap_{i=1}^n F_i \right) \cap \mathcal{S}^n_C \\ &= \left(\bigcap_{i=1}^n U_i \mathcal{S}_+^{n-|C_i|+t_i+1} U_i^T \right) \cap \mathcal{S}^n_C \\ &= \left(U \mathcal{S}_+^{t+1} U^T \right) \cap \mathcal{S}^n_C, \end{aligned}$$

where $U \in \mathbb{R}^{n \times (t+1)}$ is full column rank and satisfies

$$\text{range}(U) = \bigcap_{i=1}^n \text{range}(U_i). \tag{4.9}$$

This implies that the feasible set $\mathcal{F}$ in equation (4.8) satisifies

$$\mathcal{F} \subseteq \left\{ Y \in \left(U \mathcal{S}_+^{t+1} U^T \right) \cap \mathcal{S}^n_C : H \circ \mathcal{K}(Y) = H \circ D \right\}.$$

4.4.1 Facial reduction algorithm for a fixed embedding dimension

When D is a partial Euclidean distance matrix in $\mathbb{R}^r$, we may take the following approach. Let

$$\mathcal{C} := \{i : \operatorname{embdim}(D[C_i]) = r\}.$$

For $i, j \in \mathcal{C}$, we have $U_i \in \mathbb{R}^{n \times (n - |C_i| + r + 1)}$ and $U_i \in \mathbb{R}^{n \times (n - |C_j| + r + 1)}$. We will show that, when $|C_i \cap C_j| \geq r + 1$, and this intersection is "rigid", then

$$\dim(\operatorname{range}(U_i) \cap \operatorname{range}(U_j)) = n - |C_i \cup C_j| + r + 1.$$

Therefore, we can iteratively intersect faces in a stable way that maintains a certain dimension. Having successfully combined the faces for the cliques in $\mathcal{B} \subseteq \mathcal{C}$, we have that

$$\operatorname{face}(\mathcal{F}) \trianglelefteq \left(U \mathcal{S}_+^{n-|C|+r+1} U^T\right) \cap \mathcal{S}_C^n,$$

where $C := \cup_{i \in \mathcal{B}} C_i$.

4.5 Single subset facial reduction

We now show that if we have determined a face containing the feasible set of a subproblem, then we have determined a face containing the feasible set of the whole problem, even if that subproblem is not a clique.

Theorem 4.4 (Single Subset Facial Reduction Theorem). *Let D be an n-by-n partial Euclidean distance matrix, let H be the corresponding* 0–1 *adjacency matrix, and let*

$$\mathcal{F} := \left\{Y \in \mathcal{S}_+^n \cap \mathcal{S}_C^n : H \circ \mathcal{K}(Y) = H \circ D\right\}.$$

Let $\alpha \subseteq 1{:}n$ be an arbitrary subset,

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

and

$$\bar{\mathcal{F}}_\alpha := \left\{Y \in \mathcal{S}_+^{|\alpha|} \cap \mathcal{S}_C^{|\alpha|} : H[\alpha] \circ \mathcal{K}(Y) = H[\alpha] \circ D[\alpha]\right\}.$$

Assume, without loss of generality, that $\alpha = 1{:}k$. Let $\bar{U} \in \mathbb{R}^{|\alpha| \times (t+1)}$ with full column rank satisfy $e \in \operatorname{range}(\bar{U})$ and

$$\operatorname{face}(\bar{\mathcal{F}}_\alpha) \trianglelefteq \left(\bar{U} \mathcal{S}_+^{t+1} \bar{U}^T\right) \cap \mathcal{S}_C^{|\alpha|}. \tag{4.10}$$

Let

$$U := \begin{array}{c} \\ {\scriptstyle |\alpha|} \\ {\scriptstyle n-|\alpha|} \end{array} \!\! \begin{array}{c} \begin{array}{cc} {\scriptstyle t+1} & {\scriptstyle n-|\alpha|} \end{array} \\ \left[\begin{array}{cc} \bar{U} & 0 \\ 0 & I \end{array}\right] \end{array} \in \mathbb{R}^{n \times (n - |\alpha| + t + 1)}.$$

Then

$$\operatorname{face}(\mathcal{F}_\alpha) \trianglelefteq \left(U \mathcal{S}_+^{n-|\alpha|+t+1} U^T\right) \cap \mathcal{S}_C^n. \tag{4.11}$$

Proof. Let $Y \in \mathcal{F}_\alpha$. Then $Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C$ and $H[\alpha] \circ \mathcal{K}(Y[\alpha]) = H[\alpha] \circ D[\alpha]$. Note that $\mathcal{K}(Y[\alpha]) = \mathcal{K}\mathcal{K}^\dagger \mathcal{K}(Y[\alpha]) = \mathcal{K}(JY[\alpha]J)$. Moreover, $JY[\alpha]J \in \mathcal{S}^{|\alpha|}_+ \cap \mathcal{S}^{|\alpha|}_C$. Thus, since $JY[\alpha]J$ satisfies

$$H[\alpha] \circ \mathcal{K}(JY[\alpha]J) = H[\alpha] \circ D[\alpha],$$

we have that $JY[\alpha]J \in \bar{\mathcal{F}}_\alpha$, so $\text{range}(JY[\alpha]J) \subseteq \text{range}(\bar{U})$. Moreover, since $\text{null}(\mathcal{K}) = \text{range}(\mathcal{D}_e)$ and $\mathcal{K}(Y[\alpha] - JY[\alpha]J) = 0$, we have $Y[\alpha] = JY[\alpha]J + \mathcal{D}_e(y)$, for some $y \in \mathbb{R}^{|\alpha|}$. Since $Y[\alpha] \succeq 0$ and $JY[\alpha]Je = 0$, by Lemma 4.2 we have that

$$\text{range}(Y[\alpha]) \subseteq \text{range}(\begin{bmatrix} JY[\alpha]J & e \end{bmatrix}) \subseteq \text{range}(\bar{U}),$$

since $\text{range}(JY[\alpha]J) \subseteq \text{range}(\bar{U})$ and $e \in \text{range}(\bar{U})$. Therefore, $\text{face}\,\{Y[\alpha]\} \trianglelefteq \bar{U}\mathcal{S}^{t+1}_+\bar{U}^T$. Since $\bar{U}$ has full column rank, Lemma 4.3 implies that

$$\text{face}\mathcal{S}^n_+(\alpha, Y[\alpha]) \trianglelefteq U\mathcal{S}^{n-|\alpha|+t+1}_+ U^T,$$

where $\mathcal{S}^n_+(\alpha, Y[\alpha]) = \left\{X \in \mathcal{S}^n_+ : X[\alpha] = Y[\alpha]\right\}$. Since $Y \in \mathcal{S}^n_+(\alpha, Y[\alpha])$, we now have that $Y \in \left(U\mathcal{S}^{n-|\alpha|+t+1}_+ U^T\right) \cap \mathcal{S}^n_C$, completing the proof that equation (4.11) holds. □

4.6 Disjoint subsets facial reduction

In some situations, we are not able to collect all nodes of the graph into a single set described by a small dimensional face. Instead we may have a collection of disjoint sets of nodes, the distance constraints given by each set constraining us to some determined face. In this case, it is easy to determine a face containing the entire problem. We find that this face is simply given by a block-diagonal matrix, which we now give in the following theorem.

Theorem 4.5 (Disjoint Subsets Facial Reduction Theorem)**.** *Let D be an n-by-n partial Euclidean distance matrix, let H be the corresponding* 0–1 *adjacency matrix, and let*

$$\mathcal{F} := \left\{Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C : H \circ \mathcal{K}(Y) = H \circ D\right\}.$$

Let $\alpha_1, \ldots, \alpha_\ell \subseteq 1{:}n$ be a collection of pairwise disjoint sets, and assume, without loss of generality, that $\alpha_i = (k_{i-1}+1){:}k_i$, for $i = 1, \ldots, \ell$, with $k_0 = 0$. Let

$$\alpha := \bigcup_{i=1}^{\ell} \alpha_i = 1{:}|\alpha|.$$

For $i = 1, \ldots, \ell$, let

$$\mathcal{F}_i := \left\{Y \in \mathcal{S}^n_+ : H[\alpha_i] \circ \mathcal{K}(Y[\alpha_i]) = H[\alpha_i] \circ D[\alpha_i]\right\},$$

and let $\bar{U}_i \in \mathbb{R}^{|\alpha_i| \times (t_i+1)}$ *with full column rank satisfy* $e \in \operatorname{range}(\bar{U}_i)$ *and*

$$U_i := \begin{array}{c|ccc|} & k_{i-1} & t_i+1 & n-k_i \\ \hline k_{i-1} & I & 0 & 0 \\ |\alpha_i| & 0 & \bar{U}_i & 0 \\ n-k_i & 0 & 0 & I \\ \hline \end{array} \in \mathbb{R}^{n \times (n-|\alpha_i|+t_i+1)}$$

such that

$$\operatorname{face}(\mathcal{F}_i) \trianglelefteq U_i \mathcal{S}_+^{n-|\alpha_i|+t_i+1} U_i^T.$$

Let

$$U := \begin{array}{c|cccc|} & t_1+1 & \ldots & t_\ell+1 & n-|\alpha| \\ \hline |\alpha_1| & \bar{U}_1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ |\alpha_\ell| & 0 & \ldots & \bar{U}_\ell & 0 \\ n-|\alpha| & 0 & \ldots & 0 & I \\ \hline \end{array} \in \mathbb{R}^{n \times (n-|\alpha|+t+1)},$$

where $t := \sum_{i=1}^{\ell} t_i + \ell - 1$*. Then* $e \in \operatorname{range}(U)$ *and*

$$\operatorname{range}(U) = \bigcap_{i=1}^{\ell} \operatorname{range}(U_i)$$

and

$$\operatorname{face}(\mathcal{F}) \trianglelefteq \left(U \mathcal{S}_+^{n-|\alpha|+t+1} U^T \right) \cap \mathcal{S}_C^n.$$

Proof. Clearly we have $e \in \operatorname{range}(U)$. Suppose $x \in \operatorname{range}(U)$. Then

$$x = \begin{bmatrix} \bar{U}_1 v_1 \\ \vdots \\ \bar{U}_\ell v_\ell \\ v_{\ell+1} \end{bmatrix},$$

for some $v = \left[v_1; \cdots; v_\ell; v_{\ell+1}\right] \in \mathbb{R}^{n-|\alpha|+t+1}$. This clearly implies that $x \in \operatorname{range}(U_i)$, for all $i = 1, \ldots, \ell$, so $x \in \cap_{i=1}^{\ell} \operatorname{range}(U_i)$. Conversely, suppose $x \in \cap_{i=1}^{\ell} \operatorname{range}(U_i)$. Then, for $i = 1, \ldots, \ell$, there exist $w_i \in \mathbb{R}^{n-|\alpha_i|+t_i+1}$ such that $x = U_i w_i$. Therefore,

$$x = U_1 w_1 = \cdots = U_\ell w_\ell = \begin{bmatrix} \bar{U}_1 v_1 \\ \vdots \\ \bar{U}_\ell v_\ell \\ v_{\ell+1} \end{bmatrix},$$

where $v_i \in \mathbb{R}^{t_i+1}$ is chosen to be the appropriate subvector of w_i, for $i = 1, \ldots, \ell$, and $v_{\ell+1} \in \mathbb{R}^{n-|\alpha|}$ is the last $n - |\alpha|$ entries of w_1, for example. Therefore, $x \in \operatorname{range}(U)$.

Finally, since we now have that range$(U) = \cap_{i=1}^{\ell}$range(U_i),

$$\begin{aligned}
\operatorname{face}(\mathcal{F}) &\trianglelefteq \left(\bigcap_{i=1}^{n} \operatorname{face}(\mathcal{F}_i)\right) \cap \mathcal{S}_C^n \\
&= \left(\bigcap_{i=1}^{n} U_i \mathcal{S}_+^{n-|\alpha_i|+t_i+1} U_i^T\right) \cap \mathcal{S}_C^n \\
&= \left(U \mathcal{S}_+^{n-|\alpha|+t+1} U^T\right) \cap \mathcal{S}_C^n. \qquad \square
\end{aligned}$$

4.7 Rigid intersection facial reduction

Let D be an n-by-n partial Euclidean distance matrix, let H be the corresponding 0–1 adjacency matrix, and let

$$\mathcal{F} := \left\{Y \in \mathcal{S}_+^n \cap \mathcal{S}_C^n : H \circ \mathcal{K}(Y) = H \circ D\right\}.$$

We now show how to combine the facial representation of two subproblems together, when these two subproblems have an intersection that is rigid. Consider the situation in Figure 4.1. In Figure 4.1a, we have two cliques C_i and C_j, each with embedding dimension $r = 2$, such that embdim$(D[C_i \cap C_j]) = r$. Figure 4.1b shows a special case of this where we have a clique C_i with embedding dimension $r = 2$, and a node j such that embdim$(D[C_i \cap \mathcal{N}(j)]) = r$. We use $\mathcal{N}(j)$ to represent the *neighbours* of node j; that is,

$$\mathcal{N}(j) := \{i \in N : ij \in E\},$$

where the graph is given by $G = (N, E)$, N is the node set, and E is the edge set.

In general, we say that the subproblems given by node subsets α_1 and α_2 have a *rigid intersection* if, for $i = 1, 2$, there exists $\bar{U}_i \in \mathbb{R}^{|\alpha_i| \times (r+1)}$ with full column rank satisfying $e \in$ range$(\bar{U}_i)$ and

$$\operatorname{face}\left\{Y \in \mathcal{S}_+^{|\alpha_i|} \cap \mathcal{S}_C^{|\alpha_i|} : H[\alpha_i] \circ \mathcal{K}(Y) = H[\alpha_i] \circ D[\alpha_i]\right\} \trianglelefteq \left(\bar{U}_i \mathcal{S}_+^{r+1} \bar{U}_i^T\right) \cap \mathcal{S}_C^{|\alpha_i|},$$

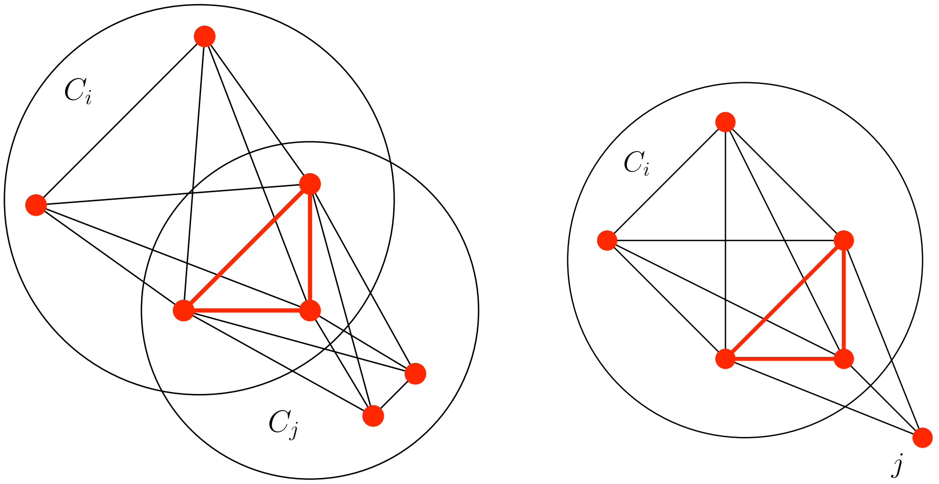
and there exists $\bar{Y} \in \mathcal{S}_+^n \cap \mathcal{S}_C^n$ satisfying

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

with embdim$(\mathcal{K}(\bar{Y}[\alpha_1 \cap \alpha_2])) = r$.

4.7.1 Rigid face intersection

We compute the matrix U satisfying condition (4.9) by computing the intersection of two range spaces at a time using the following lemma. Later we will prove in Theorem 4.8 that we can always satisfy the assumptions of this lemma, allowing us to recursively apply it in our algorithm.



(a) Clique union (b) Node absorb

Figure 4.1: Rigid intersection

Lemma 4.6 (Rigid Face Intersection Lemma)**.** *Let* $\alpha_1, \alpha_2 \subseteq 1{:}n$, $\alpha := \alpha_1 \cup \alpha_2$, *and* $k := |\alpha_1 \cap \alpha_2|$. *Let* $\bar{U}_1 \in \mathbb{R}^{|\alpha_1|\times(r+1)}$ *and* $\bar{U}_2 \in \mathbb{R}^{|\alpha_2|\times(r+1)}$ *be partitioned as follows:*

$$\bar{U}_1 =: \begin{array}{rc} & r{+}1 \\ \begin{array}{r} |\alpha_1|{-}k \\ k \end{array} & \begin{bmatrix} U_1' \\ U_1'' \end{bmatrix} \end{array}; \quad \bar{U}_2 =: \begin{array}{rc} & r{+}1 \\ \begin{array}{r} k \\ |\alpha_2|{-}k \end{array} & \begin{bmatrix} U_2'' \\ U_2' \end{bmatrix} \end{array}.$$

Let

$$U_1 := \begin{array}{rc} & \begin{array}{ccc} r{+}1 & |\alpha_2|{-}k & n{-}|\alpha| \end{array} \\ \begin{array}{r} |\alpha_1|{-}k \\ k \\ |\alpha_2|{-}k \\ n{-}|\alpha| \end{array} & \begin{bmatrix} U_1' & 0 & 0 \\ U_1'' & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \end{array} \in \mathbb{R}^{n\times(n-|\alpha_1|+r+1)}$$

and

$$U_2 := \begin{array}{rc} & \begin{array}{ccc} |\alpha_1|{-}k & r{+}1 & n{-}|\alpha| \end{array} \\ \begin{array}{r} |\alpha_1|{-}k \\ k \\ |\alpha_2|{-}k \\ n{-}|\alpha| \end{array} & \begin{bmatrix} I & 0 & 0 \\ 0 & U_2'' & 0 \\ 0 & U_2' & 0 \\ 0 & 0 & I \end{bmatrix} \end{array} \in \mathbb{R}^{n\times(n-|\alpha_2|+r+1)}.$$

Let $\bar{U} \in \mathbb{R}^{|\alpha|\times(r+1)}$ *be defined by either of the two following expressions:*

$$\bar{U} := \begin{array}{rc} & r{+}1 \\ \begin{array}{r} |\alpha_1|{-}k \\ |\alpha_2| \end{array} & \begin{bmatrix} U_1'(U_1'')^\dagger U_2'' \\ \bar{U}_2 \end{bmatrix} \end{array}; \tag{4.12a}$$

$$\bar{U} := \begin{array}{rc} & r{+}1 \\ \begin{array}{r} |\alpha_1| \\ |\alpha_2|{-}k \end{array} & \begin{bmatrix} \bar{U}_1 \\ U_2'(U_2'')^\dagger U_1'' \end{bmatrix} \end{array}. \tag{4.12b}$$

Let

$$U := \begin{array}{cc} & \begin{array}{cc} \scriptstyle r+1 & \scriptstyle n-|\alpha| \end{array} \\ \begin{array}{c} \scriptstyle |\alpha| \\ \scriptstyle n-|\alpha| \end{array} & \left[\begin{array}{cc} \bar{U} & 0 \\ 0 & I \end{array}\right] \end{array} \in \mathbb{R}^{n\times(n-|\alpha|+r+1)}.$$

If U_1'' *and* U_2'' *are full column rank and* $\operatorname{range}(U_1'') = \operatorname{range}(U_2'')$*, then*

1. $|\alpha_1 \cap \alpha_2| \geq r+1$*;*

2. $\operatorname{range}(U) = \operatorname{range}(U_1) \cap \operatorname{range}(U_2)$*;*

3. $e \in \operatorname{range}(\bar{U}_1) \cap \operatorname{range}(\bar{U}_2)$ *implies* $e \in \operatorname{range}(U)$*.*

Proof. Suppose U_1'' and U_2'' are full column rank, and $\operatorname{range}(U_1'') = \operatorname{range}(U_2'')$. Clearly we have $|\alpha_1 \cap \alpha_2| \geq r+1$, since $U_1'', U_2'' \in \mathbb{R}^{|\alpha_1\cap\alpha_2|\times(r+1)}$. First we let $\bar{U}$ be given by equation (4.12a). Suppose $x \in \operatorname{range}(U_1) \cap \operatorname{range}(U_2)$. Then

$$x = \begin{bmatrix} U_1'v_1 \\ U_1''v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} w_1 \\ U_2''w_2 \\ U_2'w_2 \\ w_3 \end{bmatrix}, \tag{4.13}$$

for some $v = \begin{bmatrix} v_1; v_2; v_3 \end{bmatrix} \in \mathbb{R}^{n-|\alpha_1|+r+1}$ and $w = \begin{bmatrix} w_1; w_2; w_3 \end{bmatrix} \in \mathbb{R}^{n-|\alpha_2|+r+1}$, partitioned appropriately. Since U_1'' is full column rank, we have $(U_1'')^\dagger U_1 = I$. Therefore,

$$v_1 = (U_1'')^\dagger U_2'' w_2,$$

implying that

$$x = \begin{bmatrix} U_1'(U_1'')^\dagger U_2''w_2 \\ U_2''w_2 \\ U_2'w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} \bar{U} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} w_2 \\ w_3 \end{bmatrix}. \tag{4.14}$$

Therefore, $x \in \operatorname{range}(U)$. Now suppose that $x \in \operatorname{range}(U)$. Then there exist $w_2 \in \mathbb{R}^{r+1}$ and $w_3 \in \mathbb{R}^{n-|\alpha|}$ satisfying equation (4.14). Let

$$\begin{aligned} v_1 &:= (U_1'')^\dagger U_2'' w_2, & w_1 &:= U_1' v_1, \\ v_2 &:= U_2' w_2, \\ v_3 &:= w_3. \end{aligned}$$

Since $\operatorname{range}(U_1'') = \operatorname{range}(U_2'')$, we have $U_1''v_1 = U_1''(U_1'')^\dagger U_2''w_2 = U_2''w_2$. Therefore, equation (4.13) holds, so $x \in \operatorname{range}(U_1) \cap \operatorname{range}(U_2)$. Thus,

$$\operatorname{range}(U) = \operatorname{range}(U_1) \cap \operatorname{range}(U_2)$$

when $\bar{U}$ is given by equation (4.12a). A similar argument gives the same result when $\bar{U}$ is given by equation (4.12b). Finally we note that $e \in \operatorname{range}(\bar{U}_1) \cap \operatorname{range}(\bar{U}_2)$ implies that

$$e \in \operatorname{range}(U_1) \cap \operatorname{range}(U_2),$$

so we have $e \in \operatorname{range}(U)$. □

4.7.2 Rigid intersection

We now show that if subproblems α_1 and α_2 have a rigid intersection, then their face representations satisfy the assumptions of Lemma 4.6. First we prove the following lemma.

Lemma 4.7. *Let $U \in \mathbb{R}^{k\times(r+1)}$. If $e \in \operatorname{range}(U)$, then*

$$\operatorname{range}(U) = \operatorname{range}(JU) \oplus \mathbb{R}\{e\} \quad \textit{and} \quad \operatorname{rank}(U) = \operatorname{rank}(JU) + 1,$$

where $J \in \mathbb{R}^{k\times k}$ is the orthogonal projector onto $\{e\}^\perp$.

Proof. Suppose $e = Uz \in \operatorname{range}(U)$. First we let $x = Uy \in \operatorname{range}(U)$. Then

$$x = Jx + \frac{\langle x, e\rangle}{\langle e, e\rangle}e = JUy + \frac{\langle x, e\rangle}{\langle e, e\rangle}e \in \operatorname{range}(JU) \oplus \mathbb{R}\{e\}.$$

Conversely, let $x = JUy + \alpha e \in \operatorname{range}(JU) \oplus \mathbb{R}\{e\}$. Then

$$Uy = JUy + \frac{\langle Uy, e\rangle}{\langle e, e\rangle}e = JUy + \frac{\langle Uy, e\rangle}{\langle e, e\rangle}Uz,$$

so we have $JUy = U\left(y - \frac{\langle Uy,e\rangle}{\langle e,e\rangle}z\right)$. Therefore,

$$x = U\left(y - \frac{\langle Uy, e\rangle}{\langle e, e\rangle}z\right) + \alpha Uz \in \operatorname{range}(U).$$

Finally, if $\{u_1, \ldots, u_\ell\}$ is a basis of $\operatorname{range}(JU)$, then $\{u_1, \ldots, u_\ell, e\}$ is a basis of $\operatorname{range}(JU) \oplus \mathbb{R}\{e\}$. Therefore, $\dim(\operatorname{range}(JU) \oplus \mathbb{R}\{e\}) = \operatorname{rank}(JU) + 1$. □

We now give the main result of Section 4.7 which, under the assumption of rigid intersection, provides the guarantee the assumptions of Lemma 4.6 are satisfied.

Theorem 4.8 (Rigid Intersection Facial Reduction Theorem)**.** *Let D be an n-by-n partial Euclidean distance matrix, let H be the corresponding* 0–1 *adjacency matrix, and let*

$$\mathcal{F} := \left\{Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C : H \circ \mathcal{K}(Y) = H \circ D\right\}.$$

Let $\alpha_1, \alpha_2 \subseteq 1{:}n$ be arbitrary subsets, $\alpha := \alpha_1 \cup \alpha_2$, and $k := |\alpha_1 \cap \alpha_2|$. For $i = 1, 2$, let

$$\mathcal{F}_i := \left\{Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C : H[\alpha_i] \circ \mathcal{K}(Y[\alpha_i]) = H[\alpha_i] \circ D[\alpha_i]\right\},$$

and let $\bar{U}_i =\in \mathbb{R}^{|\alpha_i|\times(r+1)}$ with full column rank satisfy $e \in \operatorname{range}(\bar{U}_i)$ and

$$\operatorname{face}\left\{Y \in \mathcal{S}^{|\alpha_i|}_+ \cap \mathcal{S}^{|\alpha_i|}_C : H[\alpha_i] \circ \mathcal{K}(Y) = H[\alpha_i] \circ D[\alpha_i]\right\} \trianglelefteq \left(\bar{U}_i \mathcal{S}^{r+1}_+ \bar{U}_i^T\right) \cap \mathcal{S}^{|\alpha_i|}_C.$$

Let $\bar{U}_1$ and $\bar{U}_2$ be partitioned as follows:

$$\bar{U}_1 =: \begin{array}{c c} & {\scriptstyle r+1} \\ {\scriptstyle |\alpha_1|-k} & \left[\begin{array}{c} U_1' \\ U_1'' \end{array}\right] \\ {\scriptstyle k} & \end{array}; \quad \bar{U}_2 =: \begin{array}{c c} & {\scriptstyle r+1} \\ {\scriptstyle k} & \left[\begin{array}{c} U_2'' \\ U_2' \end{array}\right] \\ {\scriptstyle |\alpha_2|-k} & \end{array}.$$

If there exists $\bar{Y} \in \mathcal{F}_1 \cap \mathcal{F}_2$ such that $\operatorname{embdim}(\bar{D}) = r$*, where $\bar{D} := \mathcal{K}(\bar{Y}[\alpha_1 \cap \alpha_2])$, then*

1. $|\alpha_1 \cap \alpha_2| \geq r+1$*;*

2. $U_1'', U_2'' \in \mathbb{R}^{k \times (r+1)}$ *have full column rank;*

3. $\operatorname{range}(U_1'') = \operatorname{range}(U_2'')$*.*

Proof. Let

$$U_1 := \begin{array}{c} \\ {\scriptstyle |\alpha_1|-k} \\ {\scriptstyle k} \\ {\scriptstyle |\alpha_2|-k} \\ {\scriptstyle n-|\alpha|} \end{array} \begin{array}{c} \begin{array}{ccc} {\scriptstyle r+1} & {\scriptstyle |\alpha_2|-k} & {\scriptstyle n-|\alpha|} \end{array} \\ \left[\begin{array}{ccc} U_1' & 0 & 0 \\ U_1'' & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{array}\right] \end{array} \in \mathbb{R}^{n \times (n-|\alpha_1|+r+1)}$$

and

$$U_2 := \begin{array}{c} \\ {\scriptstyle |\alpha_1|-k} \\ {\scriptstyle k} \\ {\scriptstyle |\alpha_2|-k} \\ {\scriptstyle n-|\alpha|} \end{array} \begin{array}{c} \begin{array}{ccc} {\scriptstyle |\alpha_1|-k} & {\scriptstyle r+1} & {\scriptstyle n-|\alpha|} \end{array} \\ \left[\begin{array}{ccc} I & 0 & 0 \\ 0 & U_2'' & 0 \\ 0 & U_2' & 0 \\ 0 & 0 & I \end{array}\right] \end{array} \in \mathbb{R}^{n \times (n-|\alpha_2|+r+1)}.$$

Since $\bar{U}_1$ and $\bar{U}_2$ have full column rank, from Theorem 4.4, we have

$$\operatorname{face}(\mathcal{F}_i) \trianglelefteq \left(U_i \mathcal{S}_+^{n-|\alpha_i|+r+1} U_i^T\right) \cap \mathcal{S}_C^n, \quad \text{for } i = 1, 2. \tag{4.15}$$

Now, suppose there exists $\bar{Y} \in \mathcal{F}_1 \cap \mathcal{F}_2$ such that $\operatorname{embdim}(\bar{D}) = r$, where $\bar{D} := \mathcal{K}(\bar{Y}[\alpha_1 \cap \alpha_2])$. Since $\bar{D} \in \mathcal{E}^{|\alpha_1 \cap \alpha_2|}$, by Theorem 2.37 we have $r \leq |\alpha_1 \cap \alpha_2| - 1$. Moreover, since

$$\mathcal{K}^\dagger(\bar{D}) = \mathcal{K}^\dagger \mathcal{K}(\bar{Y}[\alpha_1 \cap \alpha_2]) = J\bar{Y}[\alpha_1 \cap \alpha_2]J,$$

Theorem 2.37 implies that $\operatorname{rank}(J\bar{Y}[\alpha_1 \cap \alpha_2]J) = r$. Now, since $\bar{Y} \in \mathcal{F}_1$, we have, from equation (4.15), that $\operatorname{range}(\bar{Y}) \subseteq \operatorname{range}(U_1)$. Therefore,

$$\operatorname{range}(\bar{Y}[\alpha_1 \cap \alpha_2]) \subseteq \operatorname{range}(U_1''),$$

implying that

$$\operatorname{range}(J\bar{Y}[\alpha_1 \cap \alpha_2]J) \subseteq \operatorname{range}(JU_1'').$$

Since $e \in \operatorname{range}(U_1'')$ we have from Lemma 4.7 that $\operatorname{rank}(JU_1'') = \operatorname{rank}(U_1'') - 1$. Since $U_1'' \in \mathbb{R}^{|\alpha_1 \cap \alpha_2| \times (r+1)}$, we have $\operatorname{rank}(U_1'') \leq r+1$, so

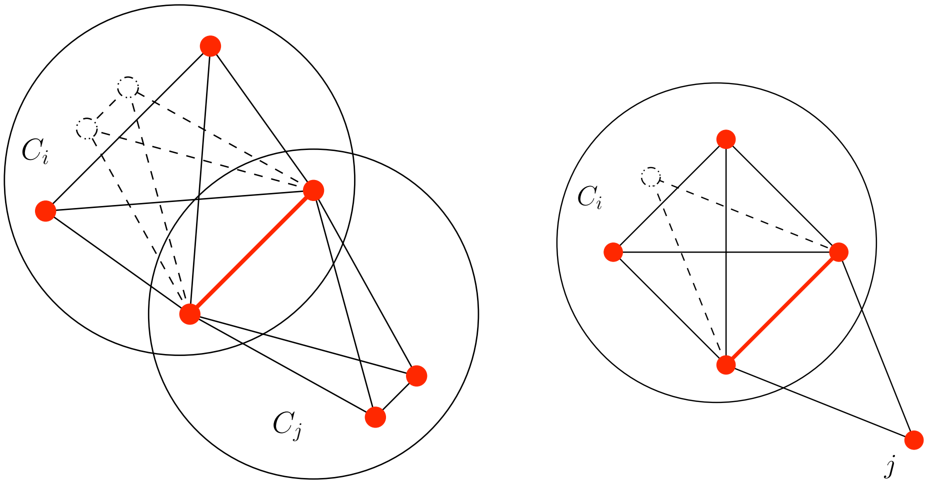
$$r = \operatorname{rank}(J\bar{Y}[\alpha_1 \cap \alpha_2]J) \leq \operatorname{rank}(JU_1'') \leq r.$$

Therefore, $\operatorname{rank}(JU_1'') = r$ and

$$\operatorname{range}(J\bar{Y}[\alpha_1 \cap \alpha_2]J) = \operatorname{range}(JU_1''). \tag{4.16}$$

Now, by Lemma 4.7, we have that

$$\operatorname{range}(U_1'') = \operatorname{range}(JU_1'') \oplus \mathbb{R}\{e\}$$



(a) Clique union (b) Node absorb

Figure 4.2: Non-rigid intersection

and

$$\operatorname{rank}(U_1'') = \operatorname{rank}(JU_1'') + 1 = r + 1,$$

so U_1'' is full column rank. Similarly, since $\bar{Y} \in \mathcal{F}_2$, we have that U_2'' is full column rank and

$$\operatorname{range}(J\bar{Y}[\alpha_1 \cap \alpha_2]J) = \operatorname{range}(JU_2''). \tag{4.17}$$

Thus, by equations (4.16) and (4.17), we have that $\operatorname{range}(JU_1'') = \operatorname{range}(JU_2'')$ Therefore,

$$\operatorname{range}(U_1'') = \operatorname{range}(JU_1'') \oplus \mathbb{R}\{e\} = \operatorname{range}(JU_2'') \oplus \mathbb{R}\{e\} = \operatorname{range}(U_2''). \qquad \square$$

4.8 Non-rigid intersection facial reduction

We now consider the situation in Figure 4.2. In Figure 4.2a, we have two cliques C_i and C_j, each with embedding dimension $r = 2$, such that $\operatorname{embdim}(D[C_i \cap C_j]) = r - 1$. Figure 4.2b shows a special case of this where we have a clique C_i with embedding dimension $r = 2$, and a node j such that $\operatorname{embdim}(D[C_i \cap \mathcal{N}(j)]) = r - 1$. Recall that $\mathcal{N}(j)$ is the set of neighbours of j in the graph.

In general, we say that the subproblems given by node subsets α_1 and α_2 have a *non-rigid intersection* if, for $i = 1, 2$, there exists $\bar{U}_i \in \mathbb{R}^{|\alpha_i| \times (r+1)}$ with full column rank satisfying $e \in \operatorname{range}(\bar{U}_i)$ and

$$\operatorname{face}\left\{Y \in \mathcal{S}_+^{|\alpha_i|} \cap \mathcal{S}_C^{|\alpha_i|} : H[\alpha_i] \circ \mathcal{K}(Y) = H[\alpha_i] \circ D[\alpha_i]\right\} \trianglelefteq \left(\bar{U}_i \mathcal{S}_+^{r+1} \bar{U}_i^T\right) \cap \mathcal{S}_C^{|\alpha_i|},$$

and there exists $\bar{Y} \in \mathcal{S}_+^n \cap \mathcal{S}_C^n$ satisfying

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

with $\operatorname{embdim}(\mathcal{K}(\bar{Y}[\alpha_1 \cap \alpha_2])) = r - 1$.

We generalize this definition and the following results in Section 4.9.

4.8.1 Non-rigid face intersection

We now give the following non-rigid version of Lemma 4.6.

Lemma 4.9 (Non-Rigid Face Intersection Lemma)**.** *Let $\alpha_1, \alpha_2 \subseteq 1{:}n$, such that $|\alpha_1 \cap \alpha_2| = r$. Let $\alpha := \alpha_1 \cup \alpha_2$. Let $\bar{U}_1 \in \mathbb{R}^{|\alpha_1|\times(r+1)}$ and $\bar{U}_2 \in \mathbb{R}^{|\alpha_2|\times(r+1)}$ satisfy $\operatorname{rank}(U_1'') = \operatorname{rank}(U_2'') = r$ and $\operatorname{range}(U_1'') = \operatorname{range}(U_2'')$, where U_1'' and U_2'' are defined by the following partitions:*

$$\bar{U}_1 =: \begin{array}{cc} & r+1 \\ \begin{array}{c} |\alpha_1|-r \\ r \end{array} & \left[\begin{array}{c} U_1' \\ U_1'' \end{array}\right] \end{array}; \qquad \bar{U}_2 =: \begin{array}{cc} & r+1 \\ \begin{array}{c} r \\ |\alpha_2|-r \end{array} & \left[\begin{array}{c} U_2'' \\ U_2' \end{array}\right] \end{array}.$$

Let

$$U_1 := \begin{array}{cc} & \begin{array}{ccc} r+1 & |\alpha_2|-r & n-|\alpha| \end{array} \\ \begin{array}{c} |\alpha_1|-r \\ r \\ |\alpha_2|-r \\ n-|\alpha| \end{array} & \left[\begin{array}{ccc} U_1' & 0 & 0 \\ U_1'' & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{array}\right] \end{array} \in \mathbb{R}^{n\times(n-|\alpha_1|+r+1)}$$

and

$$U_2 := \begin{array}{cc} & \begin{array}{ccc} |\alpha_1|-r & r+1 & n-|\alpha| \end{array} \\ \begin{array}{c} |\alpha_1|-r \\ r \\ |\alpha_2|-r \\ n-|\alpha| \end{array} & \left[\begin{array}{ccc} I & 0 & 0 \\ 0 & U_2'' & 0 \\ 0 & U_2' & 0 \\ 0 & 0 & I \end{array}\right] \end{array} \in \mathbb{R}^{n\times(n-|\alpha_2|+r+1)}.$$

Let $0 \neq z_1 \in \operatorname{null}(U_1'')$, $0 \neq z_2 \in \operatorname{null}(U_2'')$, and let $\bar{U} \in \mathbb{R}^{|\alpha|\times(r+2)}$ be defined by either of the two following expressions:

$$\bar{U} := \begin{array}{cc} & \begin{array}{cc} r+1 & 1 \end{array} \\ \begin{array}{c} |\alpha_1|-r \\ |\alpha_2| \end{array} & \left[\begin{array}{cc} U_1'(U_1'')^{\dagger}U_2'' & U_1'z_1 \\ \bar{U}_2 & 0 \end{array}\right] \end{array}; \tag{4.18a}$$

$$\bar{U} := \begin{array}{cc} & \begin{array}{cc} r+1 & 1 \end{array} \\ \begin{array}{c} |\alpha_1| \\ |\alpha_2|-r \end{array} & \left[\begin{array}{cc} \bar{U}_1 & 0 \\ U_2'(U_2'')^{\dagger}U_1'' & U_2'z_2 \end{array}\right] \end{array}. \tag{4.18b}$$

Let

$$U := \begin{array}{cc} & \begin{array}{cc} r+2 & n-|\alpha| \end{array} \\ \begin{array}{c} |\alpha| \\ n-|\alpha| \end{array} & \left[\begin{array}{cc} \bar{U} & 0 \\ 0 & I \end{array}\right] \end{array} \in \mathbb{R}^{n\times(n-|\alpha|+r+2)}.$$

Then:

1. $\operatorname{range}(U) = \operatorname{range}(U_1) \cap \operatorname{range}(U_2)$*;*

2. $e \in \operatorname{range}(\bar{U}_1) \cap \operatorname{range}(\bar{U}_2)$ *implies* $e \in \operatorname{range}(U)$*.*

Proof. Since $U_i'' \in \mathbb{R}^{r \times (r+1)}$, $\operatorname{rank}(U_i'') = r$, and $0 \neq z_i \in \operatorname{null}(U_i'')$, we have that $\operatorname{null}(U_i'') = \mathbb{R}\{z_i\}$, for $i = 1, 2$. First we let $\bar{U}$ be given by equation (4.18a). Suppose $x \in \operatorname{range}(U_1) \cap \operatorname{range}(U_2)$. Then

$$x = \begin{bmatrix} U_1'v_1 \\ U_1''v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} w_1 \\ U_2''w_2 \\ U_2'w_2 \\ w_3 \end{bmatrix}, \tag{4.19}$$

for some $v = \begin{bmatrix} v_1; v_2; v_3 \end{bmatrix} \in \mathbb{R}^{n-|\alpha_1|+r+1}$ and $w = \begin{bmatrix} w_1; w_2; w_3 \end{bmatrix} \in \mathbb{R}^{n-|\alpha_2|+r+1}$, partitioned appropriately. Therefore, by Proposition 2.6, we have that

$$v_1 \in (U_1'')^\dagger U_2'' w_2 + \operatorname{null}(U_1''),$$

so $v_1 = (U_1'')^\dagger U_2'' w_2 + \mu_1 z_1$, for some $\mu_1 \in \mathbb{R}$. Thus,

$$x = \begin{bmatrix} U_1'(U_1'')^\dagger U_2'' w_2 + \mu_1 U_1' z_1 \\ U_2'' w_2 \\ U_2' w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} \bar{U} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} w_2 \\ \mu_1 \\ w_3 \end{bmatrix}. \tag{4.20}$$

Therefore, $x \in \operatorname{range}(U)$. Now suppose that $x \in \operatorname{range}(U)$. Then there exist $w_2 \in \mathbb{R}^{r+1}$, $\mu_1 \in \mathbb{R}$, and $w_3 \in \mathbb{R}^{n-|\alpha|}$ satisfying equation (4.20). Let

$$\begin{aligned} v_1 &:= (U_1'')^\dagger U_2'' w_2 + \mu_1 z_1, \qquad & w_1 &:= U_1' v_1, \\ v_2 &:= U_2' w_2, \\ v_3 &:= w_3. \end{aligned}$$

Since $\operatorname{range}(U_1'') = \operatorname{range}(U_2'')$ and $z_1 \in \operatorname{null}(U_1'')$, we have $U_1'' v_1 = U_1''(U_1'')^\dagger U_2'' w_2 + \mu_1 U_1'' z_1 = U_2'' w_2$. Therefore, equation (4.19) holds, so $x \in \operatorname{range}(U_1) \cap \operatorname{range}(U_2)$. Thus,

$$\operatorname{range}(U) = \operatorname{range}(U_1) \cap \operatorname{range}(U_2)$$

when $\bar{U}$ is given by equation (4.18a). An similar argument gives the same result when $\bar{U}$ is given by equation (4.18b). Finally we note that $e \in \operatorname{range}(\bar{U}_1) \cap \operatorname{range}(\bar{U}_2)$ implies that

$$e \in \operatorname{range}(U_1) \cap \operatorname{range}(U_2),$$

so we have $e \in \operatorname{range}(U)$. □

4.8.2 Non-rigid intersection

We now give the main result of Section 4.8 which, under the assumption of rigid intersection, provides the guarantee the assumptions of Lemma 4.9 are satisfied.

Theorem 4.10 (Non-Rigid Intersection Facial Reduction Theorem)**.** *Let D be an n-by-n partial Euclidean distance matrix, let H be the corresponding 0–1 adjacency matrix, and let*

$$\mathcal{F} := \left\{ Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C : H \circ \mathcal{K}(Y) = H \circ D \right\}.$$

Let $\alpha_1, \alpha_2 \subseteq 1{:}n$ such that $|\alpha_1 \cap \alpha_2| = r$, and let $\alpha := \alpha_1 \cup \alpha_2$. For $i = 1, 2$, let

$$\mathcal{F}_i := \left\{ Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C : H[\alpha_i] \circ \mathcal{K}(Y[\alpha_i]) = H[\alpha_i] \circ D[\alpha_i] \right\},$$

and let $\bar{U}_i =\in \mathbb{R}^{|\alpha_i| \times (r+1)}$ with full column rank satisfy $e \in \operatorname{range}(\bar{U}_i)$ and

$$\operatorname{face}\left\{ Y \in \mathcal{S}^{|\alpha_i|}_+ \cap \mathcal{S}^{|\alpha_i|}_C : H[\alpha_i] \circ \mathcal{K}(Y) = H[\alpha_i] \circ D[\alpha_i] \right\} \trianglelefteq \left(\bar{U}_i \mathcal{S}^{r+1}_+ \bar{U}_i^T \right) \cap \mathcal{S}^{|\alpha_i|}_C.$$

Let $\bar{U}_1$ and $\bar{U}_2$ be partitioned as follows:

$$\bar{U}_1 =: \begin{array}{cc} & r+1 \\ |\alpha_1|-r & \left[\begin{array}{c} U_1' \\ U_1'' \end{array}\right] \\ r & \end{array}; \quad \bar{U}_2 =: \begin{array}{cc} & r+1 \\ r & \left[\begin{array}{c} U_2'' \\ U_2' \end{array}\right] \\ |\alpha_2|-r & \end{array}.$$

If there exists $\bar{Y} \in \mathcal{F}_1 \cap \mathcal{F}_2$ such that $\operatorname{embdim}(\bar{D}) = r-1$, where $\bar{D} := \mathcal{K}(\bar{Y}[\alpha_1 \cap \alpha_2])$, then

1. $\operatorname{rank}(U_1'') = \operatorname{rank}(U_2'') = r$;
2. $\operatorname{range}(U_1'') = \operatorname{range}(U_2'')$.

Proof. Let

$$U_1 := \begin{array}{c|ccc} & r+1 & |\alpha_2|-r & n-|\alpha| \\ \hline |\alpha_1|-r & U_1' & 0 & 0 \\ r & U_1'' & 0 & 0 \\ |\alpha_2|-r & 0 & I & 0 \\ n-|\alpha| & 0 & 0 & I \end{array} \in \mathbb{R}^{n \times (n-|\alpha_1|+r+1)}$$

and

$$U_2 := \begin{array}{c|ccc} & |\alpha_1|-r & r+1 & n-|\alpha| \\ \hline |\alpha_1|-r & I & 0 & 0 \\ r & 0 & U_2'' & 0 \\ |\alpha_2|-r & 0 & U_2' & 0 \\ n-|\alpha| & 0 & 0 & I \end{array} \in \mathbb{R}^{n \times (n-|\alpha_2|+r+1)}.$$

Since $\bar{U}_1$ and $\bar{U}_2$ have full column rank, from Theorem 4.4, we have

$$\operatorname{face}(\mathcal{F}_i) \trianglelefteq \left(U_i \mathcal{S}^{n-|\alpha_i|+r+1}_+ U_i^T \right) \cap \mathcal{S}^n_C, \quad \text{for } i = 1, 2. \tag{4.21}$$

Now, suppose there exists $\bar{Y} \in \mathcal{F}_1 \cap \mathcal{F}_2$ such that $\operatorname{embdim}(\bar{D}) = r - 1$, where $\bar{D} := \mathcal{K}(\bar{Y}[\alpha_1 \cap \alpha_2])$. Since

$$\mathcal{K}^\dagger(\bar{D}) = \mathcal{K}^\dagger \mathcal{K}(\bar{Y}[\alpha_1 \cap \alpha_2]) = J\bar{Y}[\alpha_1 \cap \alpha_2]J,$$

Theorem 2.37 implies that $\operatorname{rank}(J\bar{Y}[\alpha_1 \cap \alpha_2]J) = r - 1$. Now, since $\bar{Y} \in \mathcal{F}_1$, we have, from equation (4.21), that $\operatorname{range}(\bar{Y}) \subseteq \operatorname{range}(U_1)$. Therefore,

$$\operatorname{range}(\bar{Y}[\alpha_1 \cap \alpha_2]) \subseteq \operatorname{range}(U_1''),$$

implying that

$$\operatorname{range}(J\bar{Y}[\alpha_1 \cap \alpha_2]J) \subseteq \operatorname{range}(JU_1'').$$

Since $e \in \operatorname{range}(U_1'')$ we have from Lemma 4.7 that $\operatorname{rank}(JU_1'') = \operatorname{rank}(U_1'') - 1$. Since $U_1'' \in \mathbb{R}^{r \times (r+1)}$, we have $\operatorname{rank}(U_1'') \leq r$, so

$$r - 1 = \operatorname{rank}(J\bar{Y}[\alpha_1 \cap \alpha_2]J) \leq \operatorname{rank}(JU_1'') \leq r - 1.$$

Therefore, $\operatorname{rank}(JU_1'') = r - 1$ and

$$\operatorname{range}(J\bar{Y}[\alpha_1 \cap \alpha_2]J) = \operatorname{range}(JU_1''). \tag{4.22}$$

Now, by Lemma 4.7, we have that

$$\operatorname{range}(U_1'') = \operatorname{range}(JU_1'') \oplus \mathbb{R}\{e\}$$

and

$$\operatorname{rank}(U_1'') = \operatorname{rank}(JU_1'') + 1 = r.$$

Similarly, since $\bar{Y} \in \mathcal{F}_2$, we have that $\operatorname{rank}(U_2'') = r$ and

$$\operatorname{range}(J\bar{Y}[\alpha_1 \cap \alpha_2]J) = \operatorname{range}(JU_2''). \tag{4.23}$$

Thus, by equations (4.22) and (4.23), we have that $\operatorname{range}(JU_1'') = \operatorname{range}(JU_2'')$ Therefore,

$$\operatorname{range}(U_1'') = \operatorname{range}(JU_1'') \oplus \mathbb{R}\{e\} = \operatorname{range}(JU_2'') \oplus \mathbb{R}\{e\} = \operatorname{range}(U_2''). \qquad \square$$

4.9 Level k non-rigid intersection facial reduction

We now generalize the result from Section 4.8. We say that the subproblems given by node subsets α_1 and α_2 have a *level k non-rigid intersection* if, for $i = 1, 2$, there exists $\bar{U}_i \in \mathbb{R}^{|\alpha_i| \times (r+1)}$ with full column rank satisfying $e \in \operatorname{range}(\bar{U}_i)$ and

$$\operatorname{face}\left\{Y \in \mathcal{S}_+^{|\alpha_i|} \cap \mathcal{S}_C^{|\alpha_i|} : H[\alpha_i] \circ \mathcal{K}(Y) = H[\alpha_i] \circ D[\alpha_i]\right\} \trianglelefteq \left(\bar{U}_i \mathcal{S}_+^{r+1} \bar{U}_i^T\right) \cap \mathcal{S}_C^{|\alpha_i|},$$

and there exists $\bar{Y} \in \mathcal{S}_+^n \cap \mathcal{S}_C^n$ satisfying

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

with $\operatorname{embdim}(\mathcal{K}(\bar{Y}[\alpha_1 \cap \alpha_2])) = r - k$.

4.9.1 Level k non-rigid face intersection

We now give the level k version of Lemma 4.9. The proof is straightforward to generalize from Lemma 4.9, so we have omitted it.

Lemma 4.11 (Level k Non-Rigid Face Intersection Lemma)**.** *Let $\alpha_1, \alpha_2 \subseteq 1{:}n$, such that $|\alpha_1 \cap \alpha_2| = r+1-k$. Let $\alpha := \alpha_1 \cup \alpha_2$. Let $\bar{U}_1 \in \mathbb{R}^{|\alpha_1| \times (r+1)}$ and $\bar{U}_2 \in \mathbb{R}^{|\alpha_2| \times (r+1)}$ satisfy* $\operatorname{rank}(U_1'') = \operatorname{rank}(U_2'') = r+1-k$ *and* $\operatorname{range}(U_1'') = \operatorname{range}(U_2'')$*, where U_1'' and U_2'' are defined by the following partitions:*

$$\bar{U}_1 =: \begin{array}{c c} & {\scriptstyle r+1} \\ {\scriptstyle |\alpha_1|-r-1+k} & \left[\begin{array}{c} U_1' \\ U_1'' \end{array}\right. \\ {\scriptstyle r+1-k} & \end{array}\left.\begin{array}{c} \\ \\ \end{array}\right]; \qquad \bar{U}_2 =: \begin{array}{c c} & {\scriptstyle r+1} \\ {\scriptstyle r+1-k} & \left[\begin{array}{c} U_2'' \\ U_2' \end{array}\right. \\ {\scriptstyle |\alpha_2|-r-1+k} & \end{array}\left.\begin{array}{c} \\ \\ \end{array}\right].$$

Let

$$U_1 := \begin{array}{c|ccc|} & {\scriptstyle r+1} & {\scriptstyle |\alpha_2|-r-1+k} & {\scriptstyle n-|\alpha|} \\ {\scriptstyle |\alpha_1|-r-1+k} & U_1' & 0 & 0 \\ {\scriptstyle r+1-k} & U_1'' & 0 & 0 \\ {\scriptstyle |\alpha_2|-r-1+k} & 0 & I & 0 \\ {\scriptstyle n-|\alpha|} & 0 & 0 & I \end{array} \in \mathbb{R}^{n \times (n-|\alpha_1|+r+1)}$$

and

$$U_2 := \begin{array}{c|ccc|} & {\scriptstyle |\alpha_1|-r-1+k} & {\scriptstyle r+1} & {\scriptstyle n-|\alpha|} \\ {\scriptstyle |\alpha_1|-r-1+k} & I & 0 & 0 \\ {\scriptstyle r+1-k} & 0 & U_2'' & 0 \\ {\scriptstyle |\alpha_2|-r-1+k} & 0 & U_2' & 0 \\ {\scriptstyle n-|\alpha|} & 0 & 0 & I \end{array} \in \mathbb{R}^{n \times (n-|\alpha_2|+r+1)}.$$

For $i = 1, 2$, let $Z_i \in \mathbb{R}^{(r+1-k) \times k}$ satisfy $\operatorname{range}(Z_i) = \operatorname{null}(U_i'')$*. Let $\bar{U} \in \mathbb{R}^{|\alpha| \times (r+k+1)}$ be defined by either of the two following expressions:*

$$\bar{U} := \begin{array}{c|cc|} & {\scriptstyle r+1} & {\scriptstyle k} \\ {\scriptstyle |\alpha_1|-r-1+k} & U_1'(U_1'')^\dagger U_2'' & U_1' Z_1 \\ {\scriptstyle |\alpha_2|} & \bar{U}_2 & 0 \end{array}; \tag{4.24a}$$

$$\bar{U} := \begin{array}{c|cc|} & {\scriptstyle r+1} & {\scriptstyle k} \\ {\scriptstyle |\alpha_1|} & \bar{U}_1 & 0 \\ {\scriptstyle |\alpha_2|-r-1+k} & U_2'(U_2'')^\dagger U_1'' & U_2' Z_2 \end{array}. \tag{4.24b}$$

Let

$$U := \begin{array}{c|cc|} & {\scriptstyle r+k+1} & {\scriptstyle n-|\alpha|} \\ {\scriptstyle |\alpha|} & \bar{U} & 0 \\ {\scriptstyle n-|\alpha|} & 0 & I \end{array} \in \mathbb{R}^{n \times (n-|\alpha|+r+k+1)}.$$

Then:

1. $\operatorname{range}(U) = \operatorname{range}(U_1) \cap \operatorname{range}(U_2)$;

2. $e \in \operatorname{range}(\bar{U}_1) \cap \operatorname{range}(\bar{U}_2)$ *implies* $e \in \operatorname{range}(U)$. □

4.9.2 Level k non-rigid intersection

We now generalize Theorem 4.10. Again, it is straightforward to generalize the proof, so the proof has been omitted.

Theorem 4.12 (Level k Non-Rigid Intersection Facial Reduction Theorem)**.** *Let D be an n-by-n partial Euclidean distance matrix, let H be the corresponding* 0–1 *adjacency matrix, and let*

$$\mathcal{F} := \left\{ Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C : H \circ \mathcal{K}(Y) = H \circ D \right\}.$$

Let $\alpha_1, \alpha_2 \subseteq 1{:}n$ such that $|\alpha_1 \cap \alpha_2| = r+1-k$, and let $\alpha := \alpha_1 \cup \alpha_2$. For $i = 1, 2$, let

$$\mathcal{F}_i := \left\{ Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C : H[\alpha_i] \circ \mathcal{K}(Y[\alpha_i]) = H[\alpha_i] \circ D[\alpha_i] \right\},$$

and let $\bar{U}_i =\in \mathbb{R}^{|\alpha_i| \times (r+1)}$ with full column rank satisfy $e \in \operatorname{range}(\bar{U}_i)$ and

$$\operatorname{face}\left\{ Y \in \mathcal{S}^{|\alpha_i|}_+ \cap \mathcal{S}^{|\alpha_i|}_C : H[\alpha_i] \circ \mathcal{K}(Y) = H[\alpha_i] \circ D[\alpha_i] \right\} \trianglelefteq \left(\bar{U}_i \mathcal{S}^{r+1}_+ \bar{U}_i^T \right) \cap \mathcal{S}^{|\alpha_i|}_C.$$

Let $\bar{U}_1$ and $\bar{U}_2$ be partitioned as follows:

$$\bar{U}_1 =: \begin{array}{cc} & {\scriptstyle r+1} \\ \begin{array}{c} {\scriptstyle |\alpha_1|-r-1+k} \\ {\scriptstyle r+1-k} \end{array} & \left[\begin{array}{c} U_1' \\ U_1'' \end{array} \right] \end{array}; \quad \bar{U}_2 =: \begin{array}{cc} & {\scriptstyle r+1} \\ \begin{array}{c} {\scriptstyle r+1-k} \\ {\scriptstyle |\alpha_2|-r-1+k} \end{array} & \left[\begin{array}{c} U_2'' \\ U_2' \end{array} \right] \end{array}.$$

If there exists $\bar{Y} \in \mathcal{F}_1 \cap \mathcal{F}_2$ such that $\operatorname{embdim}(\bar{D}) = r-k$, where $\bar{D} := \mathcal{K}(\bar{Y}[\alpha_1 \cap \alpha_2])$, then

1. $\operatorname{rank}(U_1'') = \operatorname{rank}(U_2'') = r+1-k$*;*

2. $\operatorname{range}(U_1'') = \operatorname{range}(U_2'')$*.* □

4.10 Constraint reduction

We now present two theorems that show we can also reduce the number of constraints in our problem. First we show how to remove the centring constraint using facial reduction. Then we show how we can easily complete a Euclidean distance matrix using the corresponding facial representation using just a few distance constraints; this result then implies that we may discard many distance constraints for each subproblem for which we have a facial representation of the proper dimension.

4.10.1 Centring constraint reduction

We can further reduce the size of the matrix variable required by one dimension if we eliminate the centring constraint, $Y \in \mathcal{S}^n_C$, using the following technique.

Theorem 4.13 (Centring Constraint Reduction Theorem)**.** *Let D be an n-by-n partial Euclidean distance matrix, let H be the corresponding* 0–1 *adjacency matrix, and let*

$$\mathcal{F} := \left\{ Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C : H \circ \mathcal{K}(Y) = H \circ D \right\}.$$

Let $U \in \mathbb{R}^{n \times (t+1)}$ satisfy

$$\operatorname{face}\left\{ Y \in \mathcal{S}^n_+ : H \circ \mathcal{K}(Y) = H \circ D \right\} \trianglelefteq U\mathcal{S}^{t+1}_+ U^T.$$

Let $V \in \mathbb{R}^{(t+1)\times t}$ have full column rank and satisfy $\operatorname{range}(V) = \left\{U^T e\right\}^\perp$*. Then*

$$\operatorname{face}(\mathcal{F}) \trianglelefteq \left(U\mathcal{S}^{t+1}_+ U^T\right) \cap \mathcal{S}^n_C = (UV)\mathcal{S}^t_+(UV)^T.$$

Proof. First we note that $\mathcal{F} \subseteq \left(U\mathcal{S}^{t+1}_+ U^T\right) \cap \mathcal{S}^n_C$. Thus,

$$\operatorname{face}(\mathcal{F}) \trianglelefteq \left(U\mathcal{S}^{t+1}_+ U^T\right) \cap \mathcal{S}^n_C.$$

Next we note that $V^T U^T e = 0$. Therefore,

$$(UV)\mathcal{S}^t_+(UV)^T \subseteq \left(U\mathcal{S}^{t+1}_+ U^T\right) \cap \mathcal{S}^n_C.$$

Now suppose $Y \in \left(U\mathcal{S}^{t+1}_+ U^T\right) \cap \mathcal{S}^n_C$. Then $Y = UZU^T$, for some $Z \in \mathcal{S}^{t+1}_+$, and $UZU^T e = 0$. Therefore, $\left\|Z^{1/2}U^T e\right\|^2 = e^T UZU^T e = 0$, so $e^T UZ^{1/2} = 0$. Thus,

$$\operatorname{range}(Z^{1/2}) \subseteq \operatorname{null}(e^T U) = \operatorname{range}(U^T e)^\perp = \operatorname{range}(V),$$

so there exists some $\Phi \in \mathbb{R}^{t \times (t+1)}$ such that $Z^{1/2} = V\Phi$. Therefore, $Z = V\Phi\Phi^T V^T$, so $Y \in (UV)\mathcal{S}^t_+(UV)^T$. □

4.10.2 Euclidean distance matrix completion

We now show that our facial reduction technique not only reduces the size of the semidefinite matrix variable we need, but it also reduces the number of constraints we need. We begin with a result that shows that if we have determined a face with low enough dimension, then most of the equality constraints may be discarded. Moreover, with this low dimensional face and the few remaining distance constraints, we can obtain the unique completion of the partial Euclidean distance matrix.

Theorem 4.14 (Euclidean Distance Matrix Completion Theorem)**.** *Let D be an n-by-n partial Euclidean distance matrix, let H be the corresponding* 0–1 *adjacency matrix, and let*

$$\mathcal{F} := \left\{ Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C : H \circ \mathcal{K}(Y) = H \circ D \right\}.$$

Let $U \in \mathbb{R}^{n \times (t+1)}$ satisfy

$$\operatorname{face}(\mathcal{F}) \trianglelefteq \left(U\mathcal{S}^{t+1}_+ U^T\right) \cap \mathcal{S}^n_C = (UV)\mathcal{S}^t_+(UV)^T,$$

where $V \in \mathbb{R}^{(t+1)\times t}$ is full column rank and satisfies $\operatorname{range}(V) = \left\{U^T e\right\}^\perp$*. If $\bar{Y} \in \mathcal{F}$ and there exists some clique $\beta \subseteq 1{:}n$ such that* $\operatorname{embdim}(D[\beta]) = t$*, then:*

1. *$\mathcal{F} = \left\{Y \in (UV)\mathcal{S}_+^t(UV)^T : \mathcal{K}(Y[\beta]) = D[\beta]\right\} = \left\{\bar{Y}\right\}$;*

2. *$\bar{Y} = (UV)\bar{Z}(UV)^T$, where $\bar{Z} \in \mathcal{S}_+^t$ is the unique solution of the linear system*
$$(JU_\beta V)Z(JU_\beta V)^T = \mathcal{K}^\dagger(D[\beta]), \tag{4.25}$$
with $U_\beta := U[\beta, :] \in \mathbb{R}^{|\beta| \times (t+1)}$;

3. *$\bar{D} := \mathcal{K}(\bar{P}\bar{P}^T) \in \mathcal{E}^n$ is the unique completion of D, where*
$$\bar{P} := UV\bar{Z}^{1/2} \in \mathbb{R}^{n \times t}.$$

Proof. Clearly we have
$$\mathcal{F} \subseteq \left\{Y \in (UV)\mathcal{S}_+^t(UV)^T : \mathcal{K}(Y[\beta]) = D[\beta]\right\}.$$
Suppose $Y = (UV)Z(UV)^T$, for some $Z \in \mathcal{S}_+^t$, and $\mathcal{K}(Y[\beta]) = D[\beta]$. Then $Y[\beta] = (U_\beta V)Z(U_\beta V)^T$, so we have
$$\mathcal{K}\left((U_\beta V)Z(U_\beta V)^T\right) = D[\beta],$$
implying that Z satisfies equation (4.25) since $\mathcal{K}^\dagger\mathcal{K}$ is the orthogonal projection onto $\mathcal{S}_C^{|\beta|}$. Define $M := JU_\beta V \in \mathbb{R}^{|\beta| \times t}$. Then $MZM^T = \mathcal{K}^\dagger(D[\beta])$ implies that $\operatorname{rank}(M) \geq \operatorname{rank}(\mathcal{K}^\dagger(D[\beta])) = t$. Therefore M has full column rank, implying that equation (4.25) has a unique solution; this can be seen since $M\hat{Z}M^T = 0$ would imply that
$$\left\|M\hat{Z}\right\|^2 = \left\langle M\hat{Z}, M\hat{Z}\right\rangle = \left\langle M\hat{Z}M^T, \hat{Z}\right\rangle = 0,$$
so $M\hat{Z} = 0$ and hence $\hat{Z} = 0$. Therefore, $Y = \bar{Y}$, so we have
$$\left\{Y \in (UV)\mathcal{S}_+^t(UV)^T : \mathcal{K}(Y[\beta]) = D[\beta]\right\} = \left\{\bar{Y}\right\}.$$
Let $\bar{Z}$ be the unique solution of equation (4.25) and let $\bar{P} := UV\bar{Z}^{1/2}$. Then $\bar{Y} = \bar{P}\bar{P}^T$, so we have that $H \circ \mathcal{K}(\bar{P}\bar{P}^T) = H \circ D$. Thus, $\bar{D} := \mathcal{K}(\bar{P}\bar{P}^T)$ is a completion of the partial Euclidean distance matrix D. Suppose that $\hat{D} \in \mathcal{E}^n$ is another completion of D. Then $\mathcal{K}^\dagger(\hat{D}) \in \mathcal{F}$, so $\mathcal{K}^\dagger(\hat{D}) = \bar{Y}$, and thus $\hat{D} = \mathcal{K}(\bar{Y}) = \bar{D}$. Therefore, $\bar{D}$ is the unique completion of D. □

Note that in this theorem, it is possible for $|\beta|$ to be as small as $t + 1$. Thus, we may only require very few distance constraints to complete the partial Euclidean distance matrix; moreover, when $|\beta|$ is very small, solving system (4.25) is very inexpensive. Surprisingly, when the conditions of Theorem 4.14 hold, we find that we can explicitly solve for the unique feasible solution without using a semidefinite optimization solver. This result is reminiscent of other results in semidefinite optimization where the optimal solution can be obtained without the need of an semidefinite optimization solver; see, for example, Vanderbei and Yang (1995), Wolkowicz (1996), and Muramatsu (2005).

We now present the non-rigid version of Theorem 4.14. Here we show that when we have a non-rigid intersection, there are at most two different completions that are possible, each of which is easily computable by solving a simple linear system and a solve a small generalized eigenvalue problem. We may then check which of these two solutions is feasible; if exactly one of the two solutions is feasible, then we have determined the unique Euclidean distance matrix completion for the partial Euclidean distance matrix.

Theorem 4.15 (Non-Rigid Euclidean Distance Matrix Completion Theorem)**.** *Let D be an n-by-n partial Euclidean distance matrix, let H be the corresponding* 0–1 *adjacency matrix, and let*

$$\mathcal{F} := \left\{ Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C : H \circ \mathcal{K}(Y) = H \circ D \right\}.$$

Let $\alpha_1, \alpha_2 \subseteq 1{:}n$ such that $\alpha_1 \cup \alpha_2 = 1{:}n$ and $|\alpha_1 \cap \alpha_2| = r$. For $i = 1, 2$, let

$$\begin{aligned} \mathcal{F}_i &:= \left\{ Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C : H[\alpha_i] \circ \mathcal{K}(Y[\alpha_i]) = H[\alpha_i] \circ D[\alpha_i] \right\}, \\ \bar{\mathcal{F}}_i &:= \left\{ Y \in \mathcal{S}^{|\alpha_i|}_+ \cap \mathcal{S}^{|\alpha_i|}_C : H[\alpha_i] \circ \mathcal{K}(Y) = H[\alpha_i] \circ D[\alpha_i] \right\}, \end{aligned}$$

and let $\bar{U}_i =\in \mathbb{R}^{|\alpha_i| \times (r+1)}$ with full column rank satisfy $e \in$ range$(\bar{U}_i)$ and

$$\operatorname{face}\left(\bar{\mathcal{F}}_i\right) \unlhd \left(\bar{U}_i \mathcal{S}^{r+1}_+ \bar{U}_i^T\right) \cap \mathcal{S}^{|\alpha_i|}_C.$$

Let $\bar{U} \in \mathbb{R}^{n \times (r+2)}$ be given by one of the two expressions in equation (4.18)*, and let $U := \bar{U}$. Then*

$$\operatorname{face}(\mathcal{F}) \unlhd \left(U \mathcal{S}^{r+2}_+ U^T\right) \cap \mathcal{S}^n_C = (UV) \mathcal{S}^{r+1}_+ (UV)^T,$$

where $V \in \mathbb{R}^{(r+2) \times (r+1)}$ is full column rank and satisfies range$(V) = \left\{U^T e\right\}^\perp$. *For $i = 1, 2$, let $U_{\alpha_i} := U[\alpha_i, :] \in \mathbb{R}^{|\alpha_i| \times (r+2)}$ and let $J_i \in \mathcal{S}^{|\alpha_i|}$ be the orthogonal projection onto $\{e\}^\perp \subseteq \mathbb{R}^{|\alpha_i|}$.*

If $\bar{Y} = (UV)\bar{Z}(UV)^T \in \mathcal{F}_1 \cap \mathcal{F}_2$ such that $\bar{Z} \in \mathcal{S}^{r+1}_{++}$ and, for $i = 1, 2$, there exists some $\bar{Y}_i \in \bar{\mathcal{F}}_i$ and some clique $\beta_i \subseteq \alpha_i$ such that embdim$(D[\beta_i]) = r$*, then:*

1. *$\bar{\mathcal{F}}_i = \left\{ Y \in \left(\bar{U}_i \mathcal{S}^{r+1}_+ \bar{U}_i^T\right) \cap \mathcal{S}^{|\alpha_i|} : \mathcal{K}(Y[\beta_i]) = D[\beta_i] \right\} = \left\{\bar{Y}_i\right\}$, for $i = 1, 2$;*
2. *$J_i U_{\alpha_i} V \in \mathbb{R}^{|\alpha_i| \times (r+1)}$ and* rank$(J_i U_{\alpha_i} V) = r$*, for $i = 1, 2$;*
3. *$Z \in \mathcal{S}^{r+1}$ is a solution of the system of equations*

$$\begin{aligned} (J_1 U_{\alpha_1} V) Z (J_1 U_{\alpha_1} V)^T &= \bar{Y}_1 \\ (J_2 U_{\alpha_2} V) Z (J_2 U_{\alpha_2} V)^T &= \bar{Y}_2 \end{aligned} \tag{4.26}$$

if and only if

$$Z = \bar{Z} + \tau \Delta Z, \qquad \text{for some } \tau \in \mathbb{R},$$

where $\Delta Z := z_1 z_2^T + z_2 z_1^T$, for some $z_1 \in$ null$(J_1 U_{\alpha_1} V)$ and $z_2 \in$ null$(J_2 U_{\alpha_2} V)$ with $\|z_1\| = \|z_2\| = 1$;

4. *there is at least one, and at most two, positive semidefinite solutions Z_1 and Z_2 of system* (4.26) *having rank r;*

5. $\{Y \in \mathcal{F} : \operatorname{rank}(Y) = r\} \subseteq \{Y_1, Y_2\}$, *where*

$$\begin{aligned} Y_1 &:= (UV)Z_1(UV)^T, \\ Y_2 &:= (UV)Z_2(UV)^T. \end{aligned}$$

Proof. Item 1 follows directly from Theorem 4.14. Since $\bar{Y} \in \mathcal{F}_1$, we have that $J_1\bar{Y}[\alpha_1]J_1 \in \bar{\mathcal{F}}_1$, so

$$(J_1U_{\alpha_1}V)\bar{Z}(J_1U_{\alpha_1}V)^T = \bar{Y}_1.$$

Note that $\operatorname{rank}(\bar{Y}_1) = r$, so $\operatorname{rank}(J_1U_{\alpha_1}V) \geq r$. However, since $\operatorname{rank}(\bar{Z}) = r + 1$, if $\operatorname{rank}(J_1U_{\alpha_1}V) = r + 1$, then $\operatorname{rank}(\bar{Y}_1) = r + 1$ would give us a contradiction. Therefore, $\operatorname{rank}(J_1U_{\alpha_1}V) = r$. Similarly, we have that $\operatorname{rank}(J_2U_{\alpha_2}V) = r$, concluding the proof of Item 2.

Item 3 then follows from the facts that $\bar{Z}$ is a particular solution of system 4.26, and that $\dim(\operatorname{null}(J_iU_{\alpha_i}V)) = 1$, for $i = 1, 2$.

Item 4 follows from the fact that the line $\left\{\bar{Z} + \tau\Delta Z : \tau \in \mathbb{R}\right\}$ intersects the interior of the cone $\mathcal{S}^{r+1}_+$, but cannot extend infinitely far in both directions within $\mathcal{S}^{r+1}_+$, since $\mathcal{S}^{r+1}_+$ is a pointed cone. The one or two points where the line intersects the boundary of $\mathcal{S}^{r+1}_+$ can be computed by solving the generalized eigenvalue problem $-\Delta Z v = \tau \bar{Z} v$.

To prove Item 5, we let $Y \in \mathcal{F}$ such that $\operatorname{rank}(Y) = r$. Then $Y = (UV)Z(UV)^T$, for some $Z \in \mathcal{S}^{r+1}_+$. Since $Y \in \mathcal{F}_1 \cap \mathcal{F}_2$, we have that Z must satisfy system 4.26. If $\operatorname{rank}(Y) = r$, then

$$r = \operatorname{rank}(Y) \leq \operatorname{rank}(Z) \leq r + 1.$$

We claim that $\operatorname{rank}(Z) = r$. Suppose, for the sake of contradiction, that $\operatorname{rank}(Z) = r + 1$. Note that since $\operatorname{rank}(\bar{U}_i) = r + 1$, for $i = 1, 2$, we must have that $\operatorname{rank}(U) \geq r+1$, since $U = \bar{U}$ from equation (4.18). Moreover, $\operatorname{rank}(V) = r+1$. Therefore, the matrix UV has full column rank. Thus, $Z \succ 0$ implies that $\operatorname{range}(Y) = \operatorname{range}(UV)$, so $\operatorname{rank}(Y) = r + 1$, a contradiction. Therefore, $\operatorname{rank}(Z) = r$, so $Z \in \{Z_1, Z_2\}$, and we conclude that $Y \in \{Y_1, Y_2\}$. □

4.10.3 Distance constraint reduction

We now extend this result to show that whenever we have determined a face with a small enough dimension for a *subgraph* of the partial Euclidean distance matrix, then we need only keep the distance constraints coming from a clique having the correct embedding dimension and contained in the subgraph.

Theorem 4.16 (Distance Constraint Reduction Theorem)**.** *Let D be an n-by-n partial Euclidean distance matrix, let H be the corresponding* 0–1 *adjacency matrix, and let*

$$\mathcal{F} := \left\{Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C : H \circ \mathcal{K}(Y) = H \circ D\right\}.$$

Let $\alpha \subseteq 1{:}n$ be an arbitrary subset,

$$\mathcal{F}_\alpha := \left\{ Y \in \mathcal{S}^n_+ \cap \mathcal{S}^n_C : H[\alpha] \circ \mathcal{K}(Y[\alpha]) = H[\alpha] \circ D[\alpha] \right\},$$

and

$$\bar{\mathcal{F}}_\alpha := \left\{ Y \in \mathcal{S}^{|\alpha|}_+ \cap \mathcal{S}^{|\alpha|}_C : H[\alpha] \circ \mathcal{K}(Y) = H[\alpha] \circ D[\alpha] \right\}.$$

Assume, without loss of generality, that $\alpha = 1{:}k$. Let $\bar{U} \in \mathbb{R}^{|\alpha| \times (t+1)}$ with full column rank satisfy $e \in \operatorname{range}(\bar{U})$ and

$$\operatorname{face}(\bar{\mathcal{F}}_\alpha) \trianglelefteq \left(\bar{U} \mathcal{S}^{t+1}_+ \bar{U}^T \right) \cap \mathcal{S}^{|\alpha|}_C,$$

and let

$$U := \begin{array}{c c} & \begin{array}{cc} \scriptstyle t+1 & \scriptstyle n-|\alpha| \end{array} \\ \begin{array}{c} \scriptstyle |\alpha| \\ \scriptstyle n-|\alpha| \end{array} & \left[\begin{array}{cc} \bar{U} & 0 \\ 0 & I \end{array} \right] \end{array} \in \mathbb{R}^{n \times (n - |\alpha| + t + 1)}.$$

If $\bar{Y} \in \bar{\mathcal{F}}_\alpha$ and there exists some clique $\beta \subseteq \alpha$ such that $\operatorname{embdim}(D[\beta]) = t$, then:

1. $\bar{\mathcal{F}}_\alpha = \left\{ Y \in \left(\bar{U} \mathcal{S}^{t+1}_+ \bar{U}^T \right) \cap \mathcal{S}^{|\alpha|}_C : \mathcal{K}(Y[\beta]) = D[\beta] \right\} = \left\{ \bar{Y} \right\}$*;*
2. $\mathcal{F}_\alpha = \left\{ Y \in \left(U \mathcal{S}^{n-|\alpha|+t+1}_+ U^T \right) \cap \mathcal{S}^n_C : \mathcal{K}(Y[\beta]) = D[\beta] \right\}$.

Proof. Suppose that $\bar{Y} \in \bar{\mathcal{F}}_\alpha$ and there exists some clique $\beta \subseteq \alpha$ such that $\operatorname{embdim}(D[\beta]) = t$. Item 1 then follows directly from Theorem 4.14. Since $\bar{U}$ has full column rank, from Theorem 4.4, we have

$$\operatorname{face}(\mathcal{F}_\alpha) \trianglelefteq \left(U \mathcal{S}^{n-|\alpha|+t+1}_+ U^T \right) \cap \mathcal{S}^n_C,$$

implying that

$$\mathcal{F}_\alpha \subseteq \left\{ Y \in \left(U \mathcal{S}^{n-|\alpha|+t+1}_+ U^T \right) \cap \mathcal{S}^n_C : \mathcal{K}(Y[\beta]) = D[\beta] \right\}.$$

Conversely, let $Y = UZU^T$, for some $Z \in \mathcal{S}^{n-|\alpha|+t+1}_+$, such that $Y \in \mathcal{S}^n_C$ and $\mathcal{K}(Y[\beta]) = D[\beta]$. Then

$$Y[\alpha] = \bar{U} \Sigma \bar{U}^T, \quad \text{for some } \Sigma \in \mathcal{S}^{t+1}_+.$$

We now show that $Y[\alpha]$ projected onto the centred matrices belongs to $\bar{\mathcal{F}}_\alpha$. Let $X := JY[\alpha]J$. To show that $X \in \bar{\mathcal{F}}_\alpha$, we first note that $\mathcal{K}(X) = \mathcal{K}(Y[\alpha])$, so we have $\mathcal{K}(X[\beta]) = \mathcal{K}(Y[\beta]) = D[\beta]$. We will now show that $\operatorname{range}(X) \subseteq \operatorname{range}(\bar{U})$. Observe that $X = J \bar{U} \Sigma \bar{U}^T J$, and that

$$J\bar{U} = \left(I - \frac{1}{|\alpha|} ee^T \right) \bar{U} = \bar{U} - \frac{1}{|\alpha|} e(e^T \bar{U}).$$

Since $e \in \text{range}(\bar{U})$, we have that $\text{range}(X) \subseteq \text{range}(J\bar{U}) \subseteq \text{range}(\bar{U})$. Therefore, $X \in \left(\bar{U}\mathcal{S}_+^{t+1}\bar{U}^T\right) \cap \mathcal{S}_C^{|\alpha|}$. Thus, $X = JY[\alpha]J \in \bar{\mathcal{F}}_\alpha$, so $JY[\alpha]J = \bar{Y}$ and $\mathcal{K}(Y[\alpha]) = \mathcal{K}(\bar{Y})$. Moreover, since $\bar{Y} \in \bar{\mathcal{F}}_\alpha$,

$$H[\alpha] \circ \mathcal{K}(\bar{Y}) = H[\alpha] \circ D[\alpha],$$

so we also have

$$H[\alpha] \circ \mathcal{K}(Y[\alpha]) = H[\alpha] \circ D[\alpha].$$

Since $Y \in \mathcal{S}_+^n \cap \mathcal{S}_C^n$, we conclude that $Y \in \mathcal{F}_\alpha$. Therefore,

$$\mathcal{F}_\alpha = \left\{ Y \in (UV)\mathcal{S}_+^{n-|\alpha|+t}(UV)^T : \mathcal{K}(Y[\beta]) = D[\beta] \right\}. \qquad \square$$

4.11 A semidefinite facial reduction algorithm for Euclidean distance matrix completion

We give a formal description of our semidefinite facial reduction algorithm for the Euclidean distance matrix completion problem in Algorithm 1. We would like to give a note here about our terminology. Suppose C_i and C_j are cliques in the graph. It may not be the case that $C_i \cup C_j$ is a clique in the graph. However, by our method for combining cliques, we can always complete the missing distances in $C_i \cup C_j$. Therefore, after computing the face for $C_i \cup C_j$ with the proper dimension, we will refer to $C_i \cup C_j$ as a *clique*; although it is not a clique in the original graph, it is a clique in the graph created by adding all the missing edges in the induced graph of $C_i \cup C_j$. Because of our focus on cliques in the graph, we call our facial reduction algorithm `EDMSDPclique`.

The output of Algorithm 1 is a family of cliques $\{C_i\}_{i \in \mathcal{C}}$ and matrices $\left\{\bar{U}_i\right\}_{i \in \mathcal{C}}$ such that

$$\text{face}\left\{ Y \in \mathcal{S}_+^{|C_i|} \cap \mathcal{S}_C^{|C_i|} : H[C_i] \circ \mathcal{K}(Y[C_i]) = H[C_i] \circ D[C_i] \right\} \trianglelefteq \bar{U}_i \mathcal{S}_+^{r+1} \bar{U}_i^T.$$

If the output is a single clique containing all the nodes in the graph, then we can use Theorem 4.14 to compute a completion of the partial Euclidean distance matrix. Otherwise, if not all the nodes are contained in a single clique, then the facial information can be used together with a semidefinite optimization solver, or can be used to just simply complete part of the partial Euclidean distance matrix.

4.12 Point representation

We have seen in Theorem 4.14 and Theorem 4.16 that we can easily obtain a point representation giving us the unique Euclidean distance matrix completion for a subgraph from the facial representation of that subgraph, provided the face has a small

```
Algorithm 1: EDMSDPclique – a facial reduction algorithm
input  : Partial $n \times n$ Euclidean distance matrix $D$ in $\mathbb{R}^r$ with
         corresponding 0–1 adjacency matrix $H$;
output: A family of cliques $\{C_i\}_{i \in \mathcal{C}}$ and matrices $\{\bar{U}_i\}_{i \in \mathcal{C}}$ such that

   $\operatorname{face}\left\{Y \in \mathcal{S}_+^{|C_i|} \cap \mathcal{S}_C^{|C_i|} : H[C_i] \circ \mathcal{K}(Y[C_i]) = H[C_i] \circ D[C_i]\right\} \trianglelefteq \bar{U}_i \mathcal{S}_+^{r+1} \bar{U}_i^T$;

$\mathcal{C} := \emptyset$;                          /* index set for a family of cliques */
for $i = 1$ to $n$ do
    Let $C_i$ be a clique in the graph of $D$ containing node $i$;
    if $\operatorname{embdim}(D[C_i]) = r$ then
        Compute $\bar{U}_i \in \mathbb{R}^{|C_i| \times (r+1)}$ to represent the face for clique $C_i$;
                                                            /* see Theorem 4.1 */
        $\mathcal{C} := \mathcal{C} \cup \{i\}$;
    end
end
repeat
    if $|C_i \cap C_j| \geq r+1$, for some $i, j \in \mathcal{C}$ then
        RigidCliqueUnion($C_i$,$C_j$);                       /* see Algorithm 2 */
    else if $|C_i \cap \mathcal{N}(j)| \geq r+1$, for some $i \in \mathcal{C}$ and node $j$ then
        RigidNodeAbsorption($C_i$,$j$);                     /* see Algorithm 3 */
    else if $|C_i \cap C_j| = r$, for some $i, j \in \mathcal{C}$ then
        NonRigidCliqueUnion($C_i$,$C_j$);                    /* see Algorithm 4 */
    else if $|C_i \cap \mathcal{N}(j)| = r$, for some $i \in \mathcal{C}$ and node $j$ then
        NonRigidNodeAbsorption($C_i$,$j$);                  /* see Algorithm 5 */
    end
until not possible to decrease $|\mathcal{C}|$ or increase $|C_i|$ for some $i \in \mathcal{C}$;
return $\{C_i\}_{i \in \mathcal{C}}$ and $\{\bar{U}_i\}_{i \in \mathcal{C}}$;
```

```
Algorithm 2: RigidCliqueUnion
input  : cliques $C_i$ and $C_j$ such that $|C_i \cap C_j| \geq r+1$;
Load $\bar{U}_i \in \mathbb{R}^{|C_i| \times (r+1)}$ and $\bar{U}_j \in \mathbb{R}^{|C_j| \times (r+1)}$ representing the faces
corresponding to $C_i$ and $C_j$, respectively;
Compute $\bar{U} \in \mathbb{R}^{|C_i \cup C_j| \times (r+1)}$ using one of the two formulas in equation (4.12),
where $\bar{U}_1 := \bar{U}_i$ and $\bar{U}_2 := \bar{U}_j$;
Update $C_i := C_i \cup C_j$;
Update $\bar{U}_i := \bar{U}$;
Update $\mathcal{C} := \mathcal{C} \setminus \{j\}$;
```

Algorithm 3: `RigidNodeAbsorption`

```
input : clique $C_i$ and node $j$ such that $|C_i \cap \mathcal{N}(j)| \geq r+1$;
Load $\bar{U}_i \in \mathbb{R}^{|C_i| \times (r+1)}$ representing the face corresponding to $C_i$;
if $C_i \cap \mathcal{N}(j)$ is not a clique in the graph then
    Use $\bar{U}_i$ to compute a point representation $P_i \in \mathbb{R}^{|C_i| \times r}$ of $C_i$;
                                                  /* see Theorem 4.14 */
    Use $P_i$ to compute the distances between the nodes in $C_i \cap \mathcal{N}(j)$;
end
Use the distances between the nodes in $(C_i \cap \mathcal{N}(j)) \cup \{j\}$ to compute the
matrix $\bar{U}_j \in \mathbb{R}^{(|C_i \cap \mathcal{N}(j)|+1) \times (r+1)}$ representing the face corresponding to
$(C_i \cap \mathcal{N}(j)) \cup \{j\}$;                           /* see Theorem 4.1 */
Compute $\bar{U} \in \mathbb{R}^{(|C_i|+1) \times (r+1)}$ using one of the two formulas in equation (4.12),
where $\bar{U}_1 := \bar{U}_i$ and $\bar{U}_2 := \bar{U}_j$;
Update $C_i := C_i \cup \{j\}$;
Update $\bar{U}_i := \bar{U}$;
```

Algorithm 4: `NonRigidCliqueUnion`

```
input : cliques $C_i$ and $C_j$ such that $|C_i \cap C_j| = r$;
Load $\bar{U}_i \in \mathbb{R}^{|C_i| \times (r+1)}$ and $\bar{U}_j \in \mathbb{R}^{|C_j| \times (r+1)}$ representing the faces
corresponding to $C_i$ and $C_j$, respectively;
Using $\bar{U}_i$ and $\bar{U}_j$, find the two point representations of $C_i \cup C_j$;
                                                  /* see Theorem 4.15 */
if exactly one of these two point representations is feasible then
    Use the feasible point representation to compute $\bar{U} \in \mathbb{R}^{|C_i \cup C_j| \times (r+1)}$
    representing the face corresponding to $C_i \cup C_j$;  /* see Theorem 4.1 */
    Update $C_i := C_i \cup C_j$;
    Update $\bar{U}_i := \bar{U}$;
    Update $\mathcal{C} := \mathcal{C} \setminus \{j\}$;
end
```

Algorithm 5: `NonRigidNodeAbsorption`

```
input  : cliques $C_i$ and node $j$ such that $|C_i \cap \mathcal{N}(j)| = r$;
Load $\bar{U}_i \in \mathbb{R}^{|C_i| \times (r+1)}$ representing the face corresponding to $C_i$;
if $C_i \cap \mathcal{N}(j)$ not a clique in the graph then
    Use $\bar{U}_i$ to compute a point representation $P_i \in \mathbb{R}^{|C_i| \times r}$ of $C_i$;
                                                    /* see Theorem 4.14 */
    Use $P_i$ to compute the distances between the nodes in $C_i \cap \mathcal{N}(j)$;
end
Use the distances between the sensors in $(C_i \cap \mathcal{N}(j)) \cup \{j\}$ to compute the
matrix $\bar{U}_j \in \mathbb{R}^{(|C_i \cap \mathcal{N}(j)|+1) \times (r+1)}$ representing the face corresponding to
$(C_i \cap \mathcal{N}(j)) \cup \{j\}$;                          /* see Theorem 4.1 */
Using $\bar{U}_i$ and $\bar{U}_j$, find the two point representations of $C_i \cup \{j\}$;
                                                    /* see Theorem 4.15 */
if exactly one of these two point representations is feasible then
    Use the feasible point representation to compute $\bar{U} \in \mathbb{R}^{|C_i \cup C_j| \times (r+1)}$
    representing the face corresponding to $C_i \cup \{j\}$; /* see Theorem 4.1 */
    Update $C_i := C_i \cup \{j\}$;
    Update $\bar{U}_i := \bar{U}$;
end
```

enough dimension. On the other hand, if we have determined a point representation for a subgraph that gives us the unique completion of its Euclidean distance matrix, then we can treat that subgraph as a clique, and we may compute its facial representation using Theorem 4.1. Indeed, we may even choose $U_C := P_C$ in Theorem 4.1, where $P_C \in \mathbb{R}^{|C| \times t}$ satisfies $\mathcal{K}(P_C P_C^T) = D[C]$, and $t = \operatorname{embdim}(D[C])$. Note that in this case, we must have $\begin{bmatrix} P_C & e \end{bmatrix}$ full column rank, so P_C necessarily has full column rank. Moreover, we may even chose P_C centred, so that $P_C^T e = 0$.

Thus, when we can verify that our subgraph has a unique Euclidean distance matrix completion, we have that facial representations and point representations are interchangeable. This idea gives rise to an algorithm that works with explicit point representations instead of the implicit facial representations we have been discussing thus far. However, before we continue, we should stress that the facial representations give us more flexibility and have a wider scope of application over the point representation approach. Nevertheless, as we will see, the point representation also carries an advantage over the facial representation approach.

One more point we should make is that these two approaches are somewhat dual to each other. On the one hand, the goal of the facial representation approach is to *reduce* the dimension of the face by *intersecting* pairs of faces; on the other hand, the goal of the point representation approach is to *increase* the size of the set of points by rotating two sets of points together to align their overlapping points, and then taking the *union* of the two sets. We will see now the relationship between intersecting subspaces and aligning points by an orthogonal transformation.

4.12.1 Point rotation and subspace intersection

We now provide more insight into the connection between the facial representation and the point representation. In the following proposition we see the similarity between the point rotation approach and the previous approach of subspace intersection in Lemma 4.6.

Proposition 4.17 (Point Rotation)**.** *Let $\alpha_1, \alpha_2 \subseteq 1{:}n$, $\alpha := \alpha_1 \cup \alpha_2$, and $k := |\alpha_1 \cap \alpha_2|$. Let $\bar{P}_1 \in \mathbb{R}^{|\alpha_1| \times r}$ and $\bar{P}_2 \in \mathbb{R}^{|\alpha_2| \times r}$ satisfy $P_1''^T e = P_2''^T e = 0$, where P_1 and P_2 are partitioned as follows:*

$$\bar{P}_1 =: \begin{array}{cc} & r \\ |\alpha_1|-k & \left[\begin{array}{c} P_1' \\ P_1'' \end{array}\right] \\ k & \end{array}; \quad \bar{P}_2 =: \begin{array}{cc} & r \\ k & \left[\begin{array}{c} P_2'' \\ P_2' \end{array}\right] \\ |\alpha_2|-k & \end{array}.$$

Let $\bar{P} \in \mathbb{R}^{|\alpha| \times r}$ be defined by either of the two following expressions:

$$\bar{P} := \begin{array}{cc} & r \\ |\alpha_1|-k & \left[\begin{array}{c} P_1'(P_1'')^\dagger P_2'' \\ \bar{P}_2 \end{array}\right] \\ |\alpha_2| & \end{array}; \tag{4.27a}$$

$$\bar{P} := \begin{array}{cc} & r \\ |\alpha_1| & \left[\begin{array}{c} \bar{P}_1 \\ P_2'(P_2'')^\dagger P_1'' \end{array}\right] \\ |\alpha_2|-k & \end{array}. \tag{4.27b}$$

Let $Q_1 := (P_1'')^\dagger P_2'' \in \mathbb{R}^{r \times r}$ and $Q_2 := (P_2'')^\dagger P_1'' \in \mathbb{R}^{r \times r}$.

If $\begin{bmatrix} P_1'' & e \end{bmatrix}$ and $\begin{bmatrix} P_2'' & e \end{bmatrix}$ have full column rank and P_1'' and P_2'' both determine the same Euclidean distance matrix, then

1. *$|\alpha_1 \cap \alpha_2| \geq r + 1$;*

2. *$P_1'' P_1''^T = P_2'' P_2''^T$ and $\operatorname{range}(P_1'') = \operatorname{range}(P_2'')$;*

3. *Q_1 and Q_2 are othogonal;*

4. *$P_1'' Q_1 = P_2''$ and $P_2'' Q_2 = P_1''$.*

Proof. Since $\begin{bmatrix} P_1'' & e \end{bmatrix}$ and $\begin{bmatrix} P_2'' & e \end{bmatrix}$ have full column rank, we have $|\alpha_1 \cap \alpha_2| \geq r + 1$, and both P_1'' and P_2'' have full column rank. Since P_1'' and P_2'' determine the same Euclidean distance matrix, we have that $\mathcal{K}(P_1'' P_1''^T) = \mathcal{K}(P_2'' P_2''^T)$, so $\mathcal{K}^\dagger \mathcal{K}(P_1'' P_1''^T) = \mathcal{K}^\dagger \mathcal{K}(P_2'' P_2''^T)$. However, since $\mathcal{K}^\dagger \mathcal{K}$ projects onto the centred matrices, and since $P_1''^T e = P_2''^T e = 0$, we have that $P_1'' P_1''^T = P_2'' P_2''^T$. Now, since P_1'' has full column rank, we have that $\operatorname{range}(P_1''^T) = \mathbb{R}^r$. Thus, if $y = P_1'' x \in \operatorname{range}(P_1'')$, then there exists some $z \in \mathbb{R}^k$ such that $x = P_1''^T z$, so we have

$$y = P_1'' P_1''^T z = P_2'' P_2''^T z \in \operatorname{range}(P_2''),$$

so $\operatorname{range}(P_1'') \subseteq \operatorname{range}(P_2'')$. Similarly, we have $\operatorname{range}(P_2'') \subseteq \operatorname{range}(P_1'')$. Next, we have

$$\begin{aligned} Q_1 Q_1^T &= (P_1'')^\dagger P_2'' P_2''^T (P_1'')^{\dagger T} \\ &= (P_1'')^\dagger P_1'' P_1''^T (P_1'')^{\dagger T} \\ &= I, \end{aligned}$$

since P_1'' having full column rank implies that $(P_1'')^\dagger P_1'' = I$. Thus, Q_1 is orthogonal. Similarly, we also have that Q_2 is orthogonal. Finally, we see that

$$P_1'' Q_1 = P_1'' (P_1'')^\dagger P_2'' = P_2'',$$

since $P_1''(P_1'')^\dagger$ is the projection onto $\operatorname{range}(P_1'') = \operatorname{range}(P_2'')$; similarly, we have $P_2'' Q_2 = P_1''$. □

4.13 A point representation algorithm for Euclidean distance matrix completion

The main differences between our point representation algorithm and our facial representation algorithm for Euclidean distance matrix completion is how we perform two operations. First, instead of choosing any U_C with full column rank such that $\operatorname{range}(U_C) = \operatorname{range}(\mathcal{K}^\dagger(D[C]))$ in Theorem 4.1, we compute a point representation P_C of $D[C]$ as follows:

- compute the compact eigenvalue decomposition $\mathcal{K}^\dagger(D[C]) = U_C \Lambda_C U_C^T$, where $U_C \in \mathbb{R}^{|C| \times r}$ has orthonormal columns and $\Lambda_C \in \mathcal{S}^r_{++}$ is diagonal;
- let $P_C := U_C \Lambda_C^{1/2}$.

The second difference is that we solve the Procrustes problem (3.3) when performing any of the clique union or node absorption steps. In the end, we return a family of cliques $\{C_i\}_{i \in \mathcal{C}}$ and matrices $\{\bar{P}_i\}_{i \in \mathcal{C}}$ such that

$$H[C_i] \circ \mathcal{K}(P_i P_i^T) = H[C_i] \circ D[C_i], \qquad \text{for all } i \in \mathcal{C}.$$

This family of point representations can then be used to compute a face containing the feasible set of the semidefinite relaxation of the Euclidean distance matrix completion problem, which can then be used to reduce the size of the problem before solving it using a semidefinite optimization solver.

Chapter 5

Numerical Results

We have conducted our numerical experiments based on the major applications of wireless sensor network localization and molecular conformation. The primary difference between these two applications is that the sensor network localization problem includes anchors, whereas the molecular conformation problem is anchor-free. In addition, the sensor network localization tests were randomly generated and primarily in two dimensions, while the molecular conformation tests were based on real three dimensional molecular data.

5.1 Sensor network localization

We begin with discussing the numerical tests we conducted on the sensor network localization problem.

5.1.1 Random problems

We follow Biswas and Ye (2004), Biswas et al. (2006), Tseng (2007), Wang et al. (2008), Pong and Tseng (2010), Kim et al. (2009a,b) by generating our random problems using the following procedure. Let $R > 0$ be a chosen radio range, and let $G = (N, E)$ be the graph with $N = \{1, \ldots, n\}$ and

$$E = \left\{ij : i \neq j, \|x_i - x_j\|_2 \leq R\right\},$$

where $\{x_1, \ldots, x_n\} \subseteq [0,1]^r$ are distributed independently and uniformly at random.

This procedure will produce sensor network localization instances with the following properties. For $R > 0$ small enough, the expected value of the *degree* of any node $i \in N$ is

$$\mathbf{E}[\deg(i)] = (n-1)\mathrm{vol}(RB^r) = (n-1)R^r\mathrm{vol}(B^r),$$

where B^r is the unit ball in $\mathbb{R}^r$. The expected value of the *average degree* of G is

$$\begin{aligned}\mathbf{E}\left[\frac{1}{n}\sum_{i=1}^{n}\deg(i)\right] &= \frac{1}{n}\sum_{i=1}^{n}\mathbf{E}[\deg(i)]\\ &= (n-1)\operatorname{vol}(B^r)R^r,\end{aligned}$$

and the expected number of edges in the graph is

$$\begin{aligned}\mathbf{E}[|E|] &= \mathbf{E}\left[\frac{1}{2}\sum_{i=1}^{n}\deg(i)\right]\\ &= \frac{1}{2}\sum_{i=1}^{n}\mathbf{E}[\deg(i)]\\ &= \frac{n(n-1)}{2}\operatorname{vol}(B^r)R^r.\end{aligned}$$

Therefore, the expected *density* of the graph is

$$\begin{aligned}\mathbf{E}\left[\frac{|E|}{\binom{n}{2}}\right] &= \frac{\mathbf{E}[|E|]}{\frac{n(n-1)}{2}}\\ &= \operatorname{vol}(B^r)R^r.\end{aligned}$$

5.1.2 The SNLSDPclique facial reduction algorithm

We now provide an explicit description of our SNLSDPclique facial reduction algorithm, in Algorithm 6. Although this algorithm is very similar to the EDMSDPclique Algorithm 1, for clarity we have included all the details together rather than simply point out the differences. We have implemented the SNLSDPclique algorithm in Matlab and have made our source code available on our website, released under a GNU General Public License.

5.1.3 Numerical tests

Our tests are on problems with sensors and anchors randomly placed in the region $[0,1]^r$ by means of a uniform random distribution. We vary the number of sensors from 2000 to 10000 in steps of 2000, and the radio range R from .07 to .04 in steps of $-.01$. We also include tests on very large problems with 20000 to 100000 sensors. Our tests were done using the 32-bit version of Matlab R2009b on a laptop running Windows XP, with a 2.16 GHz Intel Core 2 Duo processor and with 2 GB of RAM. The source code used for running these tests is SNLSDPclique, version 0.2, and is available from the author's website. The code used for running the point representation method has not yet been made available publicly.

Tables 5.1, 5.3, and 5.5 contain the results of our tests on *noiseless* problems. These tables contain the following information.

Algorithm 6: `SNLSDPclique` – a facial reduction algorithm

```
input  : Partial $n \times n$ Euclidean distance matrix $D$ and anchors $A \in \mathbb{R}^{m \times r}$;
output: $X \in \mathbb{R}^{|C_i| \times r}$, where $C_i$ is the largest final subset that contains the
        anchors;
Let $\mathcal{C} := \{1, \ldots, n+1\}$;
Let $\{C_i\}_{i \in \mathcal{C}}$ be a family of cliques satisfying $i \in C_i$ for all $i = 1, \ldots, n$;
                                   /* For example, $C_i := \{i\}$, for $i = 1, \ldots, n$.  */
Let $C_{n+1} := \{n - m + 1, \ldots, n\}$;   /* $C_{n+1}$ is the clique of anchors */
GrowCliques ;                                        /* see Algorithm 7 */
ComputeFaces ;                                       /* see Algorithm 8 */
repeat
    if $|C_i \cap C_j| \geq r + 1$, for some $i, j \in \mathcal{C}$ then
        RigidCliqueUnion($C_i$,$C_j$);                 /* see Algorithm 9 */
    else if $|C_i \cap \mathcal{N}(j)| \geq r + 1$, for some $i \in \mathcal{C}$ and node $j$ then
        RigidNodeAbsorption($C_i$,$j$);               /* see Algorithm 10 */
    else if $|C_i \cap C_j| = r$, for some $i, j \in \mathcal{C}$ then
        NonRigidCliqueUnion($C_i$,$C_j$);             /* see Algorithm 11 */
    else if $|C_i \cap \mathcal{N}(j)| = r$, for some $i \in \mathcal{C}$ and node $j$ then
        NonRigidNodeAbsorption($C_i$,$j$);            /* see Algorithm 12 */
    end
until not possible to decrease $|\mathcal{C}|$ or increase $|C_i|$ for some $i \in \mathcal{C}$;
Let $C_i$ be the largest subset that contains the anchors;
if $C_i$ contains some sensors then
    Compute a point representation $P \in \mathbb{R}^{|C_i| \times r}$ for $C_i$;
                                                  /* see Theorem 4.14 */
    Compute positions of sensors $X \in \mathbb{R}^{(|C_i| - m) \times r}$ in $C_i$ by rotating $P$ to
    align with anchor positions $A \in \mathbb{R}^{m \times r}$          /* see Theorem 3.1 */
    return $X$;
else
    return $X := \emptyset$;
end
```

Algorithm 7: `GrowCliques`

```
Choose MaxCliqueSize $> r + 1$;
                             /* For example, MaxCliqueSize $:= 3(r + 1)$ */
for $i \in \mathcal{C}$ do
    while ($|C_i| <$ MaxCliqueSize) and ($\exists$ a node $j$ adjacent to all nodes
    in $C_i$) do
        $C_i := C_i \cup \{j\}$;
    end
end
```

Algorithm 8: `ComputeFaces`

```
for $i \in \mathcal{C}$ do
    Compute $\bar{U}_i \in \mathbb{R}^{|C_i| \times (r+1)}$ to represent the face for clique $C_i$;
                                                        /* see Theorem 4.1 */
    /* Alternatively, wait to compute $\bar{U}_i$ when first needed.  This
       can be more efficient since $\bar{U}_i$ may not be needed for every
       clique.                                                             */
end
```

Algorithm 9: `RigidCliqueUnion`

```
input  : cliques $C_i$ and $C_j$ such that $|C_i \cap C_j| \geq r + 1$;
Load $\bar{U}_i \in \mathbb{R}^{|C_i| \times (r+1)}$ and $\bar{U}_j \in \mathbb{R}^{|C_j| \times (r+1)}$ representing the faces
corresponding to $C_i$ and $C_j$, respectively;
Compute $\bar{U} \in \mathbb{R}^{|C_i \cup C_j| \times (r+1)}$ using one of the two formulas in equation (4.12),
where $\bar{U}_1 := \bar{U}_i$ and $\bar{U}_2 := \bar{U}_j$;
Update $C_i := C_i \cup C_j$;
Update $\bar{U}_i := \bar{U}$;
Update $\mathcal{C} := \mathcal{C} \setminus \{j\}$;
```

Algorithm 10: `RigidNodeAbsorption`

```
input  : clique $C_i$ and node $j$ such that $|C_i \cap \mathcal{N}(j)| \geq r + 1$;
Load $\bar{U}_i \in \mathbb{R}^{|C_i| \times (r+1)}$ representing the face corresponding to $C_i$;
if $C_i \cap \mathcal{N}(j)$ is not a clique in the graph then
    Use $\bar{U}_i$ to compute a point representation $P_i \in \mathbb{R}^{|C_i| \times r}$ of $C_i$;
                                                        /* see Theorem 4.14 */
    Use $P_i$ to compute the distances between the sensors in $C_i \cap \mathcal{N}(j)$;
end
Use the distances between the sensors in $(C_i \cap \mathcal{N}(j)) \cup \{j\}$ to compute the
matrix $\bar{U}_j \in \mathbb{R}^{(|C_i \cap \mathcal{N}(j)|+1) \times (r+1)}$ representing the face corresponding to
$(C_i \cap \mathcal{N}(j)) \cup \{j\}$;                          /* see Theorem 4.1 */
Compute $\bar{U} \in \mathbb{R}^{(|C_i|+1) \times (r+1)}$ using one of the two formulas in equation (4.12),
where $\bar{U}_1 := \bar{U}_i$ and $\bar{U}_2 := \bar{U}_j$;
Update $C_i := C_i \cup \{j\}$;
Update $\bar{U}_i := \bar{U}$;
```

Algorithm 11: `NonRigidCliqueUnion`

```
input : cliques $C_i$ and $C_j$ such that $|C_i \cap C_j| = r$;
Load $\bar{U}_i \in \mathbb{R}^{|C_i| \times (r+1)}$ and $\bar{U}_j \in \mathbb{R}^{|C_j| \times (r+1)}$ representing the faces corresponding to $C_i$ and $C_j$, respectively;
Using $\bar{U}_i$ and $\bar{U}_j$, find the two point representations of $C_i \cup C_j$;
                                                        /* see Theorem 4.15 */
if exactly one of these two point representations is feasible then
    Use the feasible point representation to compute $\bar{U} \in \mathbb{R}^{|C_i \cup C_j| \times (r+1)}$ representing the face corresponding to $C_i \cup C_j$; /* see Theorem 4.1 */
    Update $C_i := C_i \cup C_j$;
    Update $\bar{U}_i := \bar{U}$;
    Update $\mathcal{C} := \mathcal{C} \setminus \{j\}$;
end
```

Algorithm 12: `NonRigidNodeAbsorption`

```
input : cliques $C_i$ and node $j$ such that $|C_i \cap \mathcal{N}(j)| = r$;
Load $\bar{U}_i \in \mathbb{R}^{|C_i| \times (r+1)}$ representing the face corresponding to $C_i$;
if $C_i \cap \mathcal{N}(j)$ not a clique in the graph then
    Use $\bar{U}_i$ to compute a point representation $P_i \in \mathbb{R}^{|C_i| \times r}$ of $C_i$;
                                                        /* see Theorem 4.14 */
    Use $P_i$ to compute the distances between the sensors in $C_i \cap \mathcal{N}(j)$;
end
Use the distances between the sensors in $(C_i \cap \mathcal{N}(j)) \cup \{j\}$ to compute the matrix $\bar{U}_j \in \mathbb{R}^{(|C_i \cap \mathcal{N}(j)|+1) \times (r+1)}$ representing the face corresponding to $(C_i \cap \mathcal{N}(j)) \cup \{j\}$;   /* see Theorem 4.1 */
Using $\bar{U}_i$ and $\bar{U}_j$, find the two point representations of $C_i \cup \{j\}$;
                                                        /* see Theorem 4.15 */
if exactly one of these two point representations is feasible then
    Use the feasible point representation to compute $\bar{U} \in \mathbb{R}^{|C_i \cup C_j| \times (r+1)}$ representing the face corresponding to $C_i \cup \{j\}$; /* see Theorem 4.1 */
    Update $C_i := C_i \cup \{j\}$;
    Update $\bar{U}_i := \bar{U}$;
end
```

1. **# sensors, r, # anchors, and R:**

   We use $m = (\#anchors)$, $n = (\#sensors) + (\#anchors)$, and r to generate ten random instances of $p_1, \ldots, p_n \in \mathbb{R}^r$; the last m random points are taken to be the anchors. For each of these ten instances, and for each value of the radio range $R > 0$, we generate the the $n \times n$ partial Euclidean distance matrix D_p according to

   $$(D_p)_{ij} = \begin{cases} \|p_i - p_j\|^2, & \text{if } \|p_i - p_j\| < R, \text{ or both } p_i \text{ and } p_j \text{ are anchors} \\ \text{unspecified}, & \text{otherwise.} \end{cases}$$

2. **# Successful Instances:**

   An instance was called *successful* if at least some, if not all, of the sensors could be positioned. If, by the end of the algorithm, the largest clique containing the anchors did not contain any sensors, then none of the sensor positions could be determined, making such an instance unsuccessful.

3. **Average Degree:**

   We have found that the average degree of the nodes of a graph is a good indicator of the percentage of sensors that can be positioned. In the results reported, we give the average of the average degree over all ten instances.

4. **# Sensors Positioned:**

   We give the average number of sensors that could be positioned over all ten instances. Note that below we indicate that the error measurements are computed only over the sensors that could be positioned.

5. **CPU Time:**

   Indicates the average running time of `SNLSDPclique` over all ten instances. This time does not include the time to generate the random problems, but it does include all aspects of the Algorithm 6, including the time for `GrowCliques` and `ComputeFaces` at the beginning of the algorithm.

6. **Max Error:**

   The maximum distance between the positions of the sensors found and the true positions of those sensors. This is defined as

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

7. **RMSD:**

   The *root mean square deviation* of the positions of the sensors found and the true positions of those sensors. This is defined as

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

Table 5.1: Results of Algorithm 6 on *noiseless* problems, using step `RigidCliqueUnion`. The values for Average Degree, # Sensors Positioned, and CPU Time are averaged over ten random instances. The values for Max Error and RMSD values are averaged over the successful instances.

#		#		# Succ.	Avg.	# Sensors	CPU	Max	
sensors	r	anchors	R	Instances	Deg.	Positioned	Time	Error	RMSD
2000	2	4	.07	9/10	14.5	1632.3	1 s	6e-13	2e-13
2000	2	4	.06	5/10	10.7	720.0	1 s	1e-12	4e-13
2000	2	4	.05	0/10	7.5	0.0	1 s	-	-
2000	2	4	.04	0/10	4.9	0.0	1 s	-	-
4000	2	4	.07	10/10	29.0	3904.1	2 s	2e-13	6e-14
4000	2	4	.06	10/10	21.5	3922.3	2 s	6e-13	2e-13
4000	2	4	.05	10/10	15.1	3836.2	2 s	4e-13	2e-13
4000	2	4	.04	1/10	9.7	237.8	2 s	1e-13	4e-14
6000	2	4	.07	10/10	43.5	5966.9	4 s	3e-13	8e-14
6000	2	4	.06	10/10	32.3	5964.4	4 s	2e-13	7e-14
6000	2	4	.05	10/10	22.6	5894.8	3 s	3e-13	1e-13
6000	2	4	.04	10/10	14.6	5776.9	3 s	7e-13	2e-13
8000	2	4	.07	10/10	58.1	7969.8	6 s	3e-13	8e-14
8000	2	4	.06	10/10	43.0	7980.9	6 s	2e-13	8e-14
8000	2	4	.05	10/10	30.1	7953.1	5 s	6e-13	2e-13
8000	2	4	.04	10/10	19.5	7891.0	5 s	6e-13	2e-13
10000	2	4	.07	10/10	72.6	9974.6	9 s	3e-13	7e-14
10000	2	4	.06	10/10	53.8	9969.1	8 s	9e-13	1e-13
10000	2	4	.05	10/10	37.7	9925.4	7 s	5e-13	2e-13
10000	2	4	.04	10/10	24.3	9907.2	7 s	3e-13	1e-13
20000	2	4	.030	10/10	27.6	19853.3	17 s	7e-13	2e-13
40000	2	4	.020	10/10	24.7	39725.2	50 s	2e-12	6e-13
60000	2	4	.015	10/10	21.0	59461.1	1 m 52 s	1e-11	8e-13
80000	2	4	.013	10/10	21.0	79314.1	3 m 24 s	4e-12	1e-12
100000	2	4	.011	10/10	18.8	99174.4	5 m 42 s	2e-10	9e-11

We note that for each set of ten random instances, the Max Error and RMSD values reported are the average Max Error and average RMSD values over the successful instances only; this is due to the fact that an unsuccessful instance will have no computed sensor positions to compare with the true sensor positions.

We have three sets of tests on noiseless problems.

1. In Table 5.1 we report the results of using only the `RigidCliqueUnion` step to solve our random problems. Table 5.2 reports the results of the same tests, but using the point representation method, for which we observe an improved accuracy.

2. In Table 5.3 we report the results of increasing the level of our algorithm to use both the `RigidCliqueUnion` and `RigidNodeAbsorb` steps to solve the random problems. We see that the number of sensors localized has increased

Table 5.2: Results of the point representation method on *noiseless* problems, using step `RigidCliqueUnion`. The values for Average Degree, # Sensors Positioned, and CPU Time are averaged over ten random instances. The values for Max Error and RMSD values are averaged over the successful instances.

# sensors	r	# anchors	R	# Succ. Instances	Avg. Deg.	# Sensors Positioned	CPU Time	Max Error	RMSD
2000	2	4	.07	9/10	14.5	1632.3	1 s	1e-15	5e-16
2000	2	4	.06	5/10	10.7	720.0	1 s	9e-16	3e-16
2000	2	4	.05	0/10	7.5	0.0	1 s	-	-
2000	2	4	.04	0/10	4.9	0.0	1 s	-	-
4000	2	4	.07	10/10	29.0	3904.1	2 s	2e-15	5e-16
4000	2	4	.06	10/10	21.5	3922.3	2 s	1e-15	5e-16
4000	2	4	.05	10/10	15.1	3836.2	2 s	1e-15	5e-16
4000	2	4	.04	1/10	9.7	237.8	2 s	1e-15	6e-16
6000	2	4	.07	10/10	43.5	5966.9	3 s	1e-15	4e-16
6000	2	4	.06	10/10	32.3	5964.4	3 s	1e-15	5e-16
6000	2	4	.05	10/10	22.6	5894.8	3 s	2e-15	8e-16
6000	2	4	.04	10/10	14.6	5776.9	3 s	2e-15	6e-16
8000	2	4	.07	10/10	58.1	7969.8	5 s	1e-15	5e-16
8000	2	4	.06	10/10	43.0	7980.9	5 s	2e-15	5e-16
8000	2	4	.05	10/10	30.1	7953.1	4 s	1e-15	5e-16
8000	2	4	.04	10/10	19.5	7891.0	4 s	1e-15	5e-16
10000	2	4	.07	10/10	72.6	9974.6	7 s	3e-15	9e-16
10000	2	4	.06	10/10	53.8	9969.1	6 s	2e-15	7e-16
10000	2	4	.05	10/10	37.7	9925.4	6 s	2e-15	6e-16
10000	2	4	.04	10/10	24.3	9907.2	5 s	3e-15	1e-15
20000	2	4	.030	10/10	27.6	19853.3	14 s	2e-15	8e-16
40000	2	4	.020	10/10	24.7	39725.2	41 s	3e-15	9e-16
60000	2	4	.015	10/10	21.0	59461.1	1 m 20 s	3e-15	9e-16
80000	2	4	.013	10/10	21.0	79314.1	2 m 34 s	3e-15	9e-16
100000	2	4	.011	10/10	18.8	99174.4	4 m 6 s	4e-15	1e-15

Table 5.3: Results of Algorithm 6 on *noiseless* problems, using steps RigidCliqueUnion and RigidNodeAbsorb. The values for Average Degree, # Sensors Positioned, and CPU Time are averaged over ten random instances. The values for Max Error and RMSD values are averaged over the successful instances.

# sensors	r	# anchors	R	# Succ. Instances	Avg. Deg.	# Sensors Positioned	CPU Time	Max Error	RMSD
2000	2	4	.07	10/10	14.5	2000.0	1 s	6e-13	2e-13
2000	2	4	.06	10/10	10.7	1999.9	1 s	8e-13	3e-13
2000	2	4	.05	10/10	7.5	1996.7	1 s	9e-13	2e-13
2000	2	4	.04	9/10	4.9	1273.8	3 s	2e-11	4e-12
4000	2	4	.07	10/10	29.0	4000.0	2 s	2e-13	6e-14
4000	2	4	.06	10/10	21.5	4000.0	2 s	6e-13	2e-13
4000	2	4	.05	10/10	15.1	3999.9	2 s	6e-13	3e-13
4000	2	4	.04	10/10	9.7	3998.2	2 s	1e-12	5e-13
6000	2	4	.07	10/10	43.5	6000.0	4 s	3e-13	8e-14
6000	2	4	.06	10/10	32.3	6000.0	4 s	2e-13	7e-14
6000	2	4	.05	10/10	22.6	6000.0	3 s	3e-13	1e-13
6000	2	4	.04	10/10	14.6	5999.4	3 s	8e-13	3e-13
8000	2	4	.07	10/10	58.1	8000.0	6 s	3e-13	7e-14
8000	2	4	.06	10/10	43.0	8000.0	5 s	2e-13	8e-14
8000	2	4	.05	10/10	30.1	8000.0	5 s	6e-13	2e-13
8000	2	4	.04	10/10	19.5	8000.0	4 s	7e-13	2e-13
10000	2	4	.07	10/10	72.6	10000.0	9 s	3e-13	7e-14
10000	2	4	.06	10/10	53.8	10000.0	8 s	3e-13	1e-13
10000	2	4	.05	10/10	37.7	10000.0	7 s	5e-13	2e-13
10000	2	4	.04	10/10	24.3	10000.0	6 s	3e-13	1e-13
20000	2	4	.030	10/10	27.6	20000.0	17 s	7e-13	2e-13
40000	2	4	.020	10/10	24.7	40000.0	51 s	2e-12	6e-13
60000	2	4	.015	10/10	21.0	60000.0	1 m 53 s	2e-12	7e-13
80000	2	4	.013	10/10	21.0	80000.0	3 m 21 s	4e-12	1e-12
100000	2	4	.011	10/10	18.8	100000.0	5 m 46 s	2e-10	9e-11

and that there has been a small, almost insignificant, increase in the CPU time. Table 5.4 reports the results of the same tests, but using the point representation method – again we observe better accuracy.

3. In Table 5.5 we report the results of increasing the level of our algorithm to use steps RigidCliqueUnion, RigidNodeAbsorb, and NonRigidCliqueUnion to solve the random problems, further increasing the class of problems that we can complete. Tests using all these three steps with the point representation method are expected in a forth-coming paper, but we expect the results to be similar to the results obtained using only the rigid steps.

These tests show that our algorithm performs very well, even on very large problems. We obtain RMSD values on the order of machine precision, regardless of the problem size; see Table 5.4. Thus, we were able to exactly solve most of these large semidefinite optimization problems – we did not need a semidefinite optimization

Table 5.4: Results of the point representation method on *noiseless* problems, using steps `RigidCliqueUnion` and `RigidNodeAbsorb`. The values for Average Degree, # Sensors Positioned, and CPU Time are averaged over ten random instances. The values for Max Error and RMSD values are averaged over the successful instances.

#		#		# Succ.	Avg.	# Sensors	CPU	Max	
sensors	r	anchors	R	Instances	Deg.	Positioned	Time	Error	RMSD
2000	2	4	.07	10/10	14.5	2000.0	1 s	1e-15	5e-16
2000	2	4	.06	10/10	10.7	1999.9	1 s	2e-15	6e-16
2000	2	4	.05	10/10	7.5	1996.7	1 s	3e-15	7e-16
2000	2	4	.04	9/10	4.9	1274.4	2 s	3e-15	7e-16
4000	2	4	.07	10/10	29.0	4000.0	2 s	2e-15	6e-16
4000	2	4	.06	10/10	21.5	4000.0	2 s	1e-15	6e-16
4000	2	4	.05	10/10	15.1	3999.9	2 s	2e-15	5e-16
4000	2	4	.04	10/10	9.7	3998.2	2 s	2e-15	8e-16
6000	2	4	.07	10/10	43.5	6000.0	3 s	1e-15	5e-16
6000	2	4	.06	10/10	32.3	6000.0	3 s	1e-15	5e-16
6000	2	4	.05	10/10	22.6	6000.0	3 s	2e-15	8e-16
6000	2	4	.04	10/10	14.6	5999.4	3 s	2e-15	6e-16
8000	2	4	.07	10/10	58.1	8000.0	5 s	1e-15	5e-16
8000	2	4	.06	10/10	43.0	8000.0	4 s	2e-15	5e-16
8000	2	4	.05	10/10	30.1	8000.0	4 s	2e-15	5e-16
8000	2	4	.04	10/10	19.5	8000.0	4 s	1e-15	5e-16
10000	2	4	.07	10/10	72.6	10000.0	7 s	3e-15	9e-16
10000	2	4	.06	10/10	53.8	10000.0	6 s	2e-15	7e-16
10000	2	4	.05	10/10	37.7	10000.0	6 s	2e-15	6e-16
10000	2	4	.04	10/10	24.3	10000.0	5 s	3e-15	1e-15
20000	2	4	.030	10/10	27.6	20000.0	14 s	2e-15	8e-16
40000	2	4	.020	10/10	24.7	40000.0	42 s	3e-15	8e-16
60000	2	4	.015	10/10	21.0	60000.0	1 m 27 s	3e-15	9e-16
80000	2	4	.013	10/10	21.0	80000.0	2 m 28 s	3e-15	1e-15
100000	2	4	.011	10/10	18.8	100000.0	3 m 55 s	4e-15	1e-15

Table 5.5: Results of Algorithm 6 on *noiseless* problems, using steps `RigidCliqueUnion`, `RigidNodeAbsorb`, and `NonRigidCliqueUnion`. The values for Average Degree, # Sensors Positioned, and CPU Time are averaged over ten random instances. The values for Max Error and RMSD values are averaged over the successful instances. The results of the tests with more than 6000 sensors remain the same as in Table 5.3.

#		#		# Succ.	Avg.	# Sensors	CPU	Max	
sensors	r	anchors	R	Instances	Deg.	Positioned	Time	Error	RMSD
2000	2	4	.07	10/10	14.5	2000.0	1 s	6e-13	2e-13
2000	2	4	.06	10/10	10.7	1999.9	1 s	8e-13	3e-13
2000	2	4	.05	10/10	7.5	1997.9	1 s	9e-13	2e-13
2000	2	4	.04	10/10	4.9	1590.8	5 s	2e-11	7e-12
4000	2	4	.07	10/10	29.0	4000.0	2 s	2e-13	6e-14
4000	2	4	.06	10/10	21.5	4000.0	2 s	6e-13	2e-13
4000	2	4	.05	10/10	15.1	3999.9	2 s	6e-13	3e-13
4000	2	4	.04	10/10	9.7	3998.2	3 s	1e-12	5e-13
6000	2	4	.07	10/10	43.5	6000.0	4 s	3e-13	8e-14
6000	2	4	.06	10/10	32.3	6000.0	4 s	2e-13	7e-14
6000	2	4	.05	10/10	22.6	6000.0	3 s	3e-13	1e-13
6000	2	4	.04	10/10	14.6	5999.4	3 s	8e-13	3e-13

solver because the facial reduction allowed us to determine that the feasible set was a singleton and could then be easily computed by solving a simple linear equation; see Theorem 4.14.

Testing a version of our algorithm that uses all four steps is still ongoing. From the above results, we can see that our facial reduction technique works very well for solving many instances of the sensor network localization problem. We are confident that the results of our ongoing tests will continue to show that we are able to solve an even larger class of sensor network localization problems.

5.1.4 Noisy data and higher dimensional problems

We used the *multiplicative noise model* noise model for our tests on noisy problems:

$$d_{ij} = \|p_i - p_j\|(1 + \sigma\varepsilon_{ij}), \quad \text{for all } ij \in E,$$

where $\sigma \geq 0$ represents the *noise factor* and, for all $ij \in E$, the random variable ε_{ij} is normally distributed with zero mean and standard deviation one. That is, $\{\varepsilon_{ij}\}_{ij\in E}$ are uncorrelated, have zero mean and the same variance. Here we are modelling the situation that the amount of additive noise corrupting a distance measurement between two sensors is directly proportional to the distance between the sensors.

This multiplicative noise model is the one most commonly considered in sensor network localization; see, for example, Biswas and Ye (2004), Biswas et al. (2006), Tseng (2007), Wang et al. (2008), Pong and Tseng (2010), Kim et al. (2009a,b). For large values of σ, it is possible that $1+\sigma\varepsilon$ is negative. Therefore, the alternate multiplicative noise model

$$d_{ij} = \|p_i - p_j\| |1+\sigma\varepsilon_{ij}|, \quad \text{for all } ij \in E,$$

is sometimes used. Note, however, that in both multiplicative noise models, we have

$$d_{ij}^2 = \|p_i - p_j\|^2 (1+\sigma\varepsilon_{ij})^2, \quad \text{for all } ij \in E. \tag{5.1}$$

Therefore, when generating our noisy problems, we have used equation (5.1).

The associated least squares problem for determining the *maximum likelihood* positions for $p_1, \ldots, p_n \in \mathbb{R}^r$ is

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{ij\in E} v_{ij}^2 \\ \text{subject to} & \|p_i - p_j\|^2(1+v_{ij})^2 = d_{ij}^2, \quad \text{for all } ij \in E \\ & \sum_{i=1}^n p_i = 0 \\ & p_1, \ldots, p_n \in \mathbb{R}^r. \end{array}$$

Let H be the 0–1 adjacency matrix associated with the n-by-n partial Euclidean distance matrix $D := (d_{ij}^2)$. Letting $V := (v_{ij}) \in \mathbb{R}^{n\times n}$ and $V_H := H \circ V$, we can use the $\mathcal{K}$ map defined in equation (2.5), to rewrite this problem as

$$\begin{array}{ll} \text{minimize} & \|V_H\|_F^2 \\ \text{subject to} & \mathcal{K}(PP^T) \circ (H + 2V_H + V_H \circ V_H) = H \circ D \\ & P^T e = 0 \\ & P \in \mathbb{R}^{n\times r}. \end{array}$$

Removing the rank constraint, we obtain the (nonlinear) semidefinite relaxation

$$\begin{array}{ll} \text{minimize} & \|V_H\|_F^2 \\ \text{subject to} & \mathcal{K}(Y) \circ (H + 2V_H + V_H \circ V_H) = H \circ D \\ & Y \in \mathcal{S}_+^n \cap \mathcal{S}_C. \end{array} \tag{5.2}$$

However, we should note that we have not used any refinement technique in our numerical tests on noisy problems. We are currently investigating refinement techniques, and will look at local search methods applied to the nonlinear least squares problems mentioned here. Our main point is that it is important to match the noise model with the appropriate least squares problem when handling noisy problems.

Instead, our approach here to handling noisy problems was to simply compute the nearest rank r matrix to the matrix $\mathcal{K}^\dagger(D[C])$, for each clique C, using the well-know Eckert-Young result, Theorem 3.5. In other words, to represent our face, we only use the eigenvectors corresponding to the r largest (in absolute value) eigenvalues.

In our future work, we plan to compare the following two approaches. Let D be an n-by-n Euclidean distance matrix corrupted by noise (hence D may not even be a true Euclidean distance matrix since $\mathcal{K}^\dagger(D)$ may have negative eigenvalues).

1. We compute the eigenvalue decomposition $\mathcal{K}^\dagger(D) = U\Lambda U^T$, and let $P := U_r\Lambda_r^{1/2} \in \mathbb{R}^{n\times r}$. This matrix P minimizes $\left\|PP^T - \mathcal{K}^\dagger(D)\right\|_F$ over all $P \in \mathbb{R}^{n\times r}$, and satisfies

$$\left\|PP^T - \mathcal{K}^\dagger(D)\right\|_F = \sqrt{\sum_{i=r+1}^{n} \lambda_i^2(\mathcal{K}^\dagger(D))}.$$

   Since we assume that $\operatorname{diag}(D) = 0$, we have that $\mathcal{K}\mathcal{K}^\dagger(D) = D$. Therefore,

$$\begin{aligned}\left\|\mathcal{K}(PP^T) - D\right\|_F &= \left\|\mathcal{K}(PP^T - \mathcal{K}^\dagger(D))\right\|_F \\ &\leq \left\|\mathcal{K}\right\|_F \cdot \left\|PP^T - \mathcal{K}^\dagger(D)\right\|_F \\ &= 2\sqrt{n}\sqrt{\sum_{i=r+1}^{n} \lambda_i^2(\mathcal{K}^\dagger(D))}.\end{aligned}$$

2. We can compute better Euclidean distance matrix approximations of D by increasing the rank of the approximation PP^T of $\mathcal{K}^\dagger(D)$. That is, we let $P := U_k\Lambda_k^{1/2} \in \mathbb{R}^{n\times k}$, for some $k > r$; see, for example, (Alfakih et al., 1999, Lemma 2). Our facial reduction technique lends well to this approach. There is no problem for us to compute the intersection of different faces that occupy different dimensions. This investigation is ongoing.

The results of our noisy and higher dimensional tests are as follows.

1. In Table 5.6 we report the result of using the rigid steps `RigidCliqueUnion` and `RigidNodeAbsorb` on problems in $\mathbb{R}^2$ and $\mathbb{R}^3$ with and without noise.

2. Table 5.7 runs the same tests, but with the point representation method using the rigid steps.

5.1.5 Comparison with existing algorithms

We now compare our running times and accuracy to the existing SDP-based algorithms. Currently, the Sparse FSDP (SFSDP) method of Kim et al. (2008, 2009a,b) and the Log-barrier Penalty Coordinate Gradient Descent (LPCGD) method of Pong and Tseng (2010) are the most efficient SDP-based methods available. Before these methods came out, the ESDP method from Wang et al. (2008) was the most efficient SDP-based algorithm for solving the sensor network localization problem.

First we note that most of the tests run in Kim et al. (2009b) and Pong and Tseng (2010) have 10% of the nodes in the network as anchors and 90% as sensors. The reason for having so many anchors is that SDP-based methods typically require

Table 5.6: Results of Algorithm 6 for problems with noise and $r = 2, 3$, using `RigidCliqueUnion` and `RigidNodeAbsorb`. The values for Average Degree, # Sensors Positioned, CPU Time, Max Error and RMSD are averaged over ten random instances.

	#		#		Avg.	# Sensors	CPU	Max	
σ	sensors	r	anchors	R	Deg.	Positioned	Time	Error	RMSD
0	2000	2	4	.08	18.8	2000.0	1 s	1e-13	3e-14
1e-6	2000	2	4	.08	18.8	2000.0	1 s	2e-04	4e-05
1e-4	2000	2	4	.08	18.8	2000.0	1 s	2e-02	4e-03
1e-2	2000	2	4	.08	18.8	2000.0	1 s	2e+01	3e+00
0	6000	2	4	.06	32.3	6000.0	4 s	2e-13	7e-14
1e-6	6000	2	4	.06	32.3	6000.0	4 s	8e-04	3e-04
1e-4	6000	2	4	.06	32.3	6000.0	4 s	9e-02	3e-02
1e-2	6000	2	4	.06	32.3	6000.0	4 s	2e+01	3e+00
0	10000	2	4	.04	24.3	10000.0	6 s	3e-13	1e-13
1e-6	10000	2	4	.04	24.3	10000.0	6 s	5e-04	2e-04
1e-4	10000	2	4	.04	24.3	10000.0	6 s	5e-02	2e-02
1e-2	10000	2	4	.04	24.3	10000.0	7 s	4e+02	1e+02
0	2000	3	5	.20	26.6	2000.0	1 s	3e-13	8e-14
1e-6	2000	3	5	.20	26.6	2000.0	1 s	7e-04	2e-04
1e-4	2000	3	5	.20	26.6	2000.0	1 s	8e-02	2e-02
1e-2	2000	3	5	.20	26.6	2000.0	1 s	2e+03	4e+02
0	6000	3	5	.15	35.6	6000.0	5 s	3e-13	6e-14
1e-6	6000	3	5	.15	35.6	6000.0	5 s	1e-03	2e-04
1e-4	6000	3	5	.15	35.6	6000.0	5 s	1e-01	2e-02
1e-2	6000	3	5	.15	35.6	6000.0	6 s	9e+01	9e+00
0	10000	3	5	.10	18.7	10000.0	9 s	3e-12	2e-13
1e-6	10000	3	5	.10	18.7	10000.0	10 s	4e-02	2e-03
1e-4	10000	3	5	.10	18.7	10000.0	10 s	2e+00	8e-02
1e-2	10000	3	5	.10	18.7	10000.0	10 s	4e+02	1e+01

Table 5.7: Results of the point representation method for problems with noise and $r = 2, 3$, using `RigidCliqueUnion` and `RigidNodeAbsorb`. The values for Average Degree, # Sensors Positioned, CPU Time, Max Error and RMSD are averaged over ten random instances.

σ	# sensors	r	# anchors	R	Avg. Deg.	# Sensors Positioned	CPU Time	Max Error	RMSD
0	2000	2	4	.08	18.8	2000.0	1 s	1e-15	5e-16
1e-6	2000	2	4	.08	18.8	2000.0	1 s	3e-06	1e-06
1e-4	2000	2	4	.08	18.8	2000.0	1 s	3e-04	1e-04
1e-2	2000	2	4	.08	18.8	2000.0	1 s	3e-01	7e-02
0	6000	2	4	.06	32.3	6000.0	3 s	1e-15	5e-16
1e-6	6000	2	4	.06	32.3	6000.0	3 s	3e-06	1e-06
1e-4	6000	2	4	.06	32.3	6000.0	3 s	3e-04	1e-04
1e-2	6000	2	4	.06	32.3	6000.0	3 s	7e-01	2e-01
0	10000	2	4	.04	24.3	10000.0	5 s	3e-15	1e-15
1e-6	10000	2	4	.04	24.3	10000.0	5 s	4e-06	1e-06
1e-4	10000	2	4	.04	24.3	10000.0	5 s	4e-04	1e-04
1e-2	10000	2	4	.04	24.3	10000.0	5 s	6e-01	2e-01
0	2000	3	5	.20	26.6	2000.0	1 s	4e-15	2e-15
1e-6	2000	3	5	.20	26.6	2000.0	1 s	8e-06	3e-06
1e-4	2000	3	5	.20	26.6	2000.0	1 s	8e-04	3e-04
1e-2	2000	3	5	.20	26.6	2000.0	1 s	7e-01	2e-01
0	6000	3	5	.15	35.6	6000.0	4 s	4e-15	1e-15
1e-6	6000	3	5	.15	35.6	6000.0	4 s	7e-06	2e-06
1e-4	6000	3	5	.15	35.6	6000.0	4 s	2e-02	2e-03
1e-2	6000	3	5	.15	35.6	6000.0	4 s	6e-01	1e-01
0	10000	3	5	.10	18.7	10000.0	7 s	5e-15	2e-15
1e-6	10000	3	5	.10	18.7	10000.0	7 s	1e-05	4e-06
1e-4	10000	3	5	.10	18.7	10000.0	7 s	6e-02	3e-03
1e-2	10000	3	5	.10	18.7	10000.0	7 s	1e+00	3e-01

the sensors to be in the convex hull of the anchors in order to return good solutions – if the anchors are chosen randomly, this requirement can be met by having many anchors. However, we found that our `SNLSDPclique` algorithm is capable of returning very good solutions even with only a few randomly placed anchor nodes. Indeed, we ran the SFSDP code on random instances like those in our tests and found that sensors far away from the convex hull of the anchors were very poorly localized. Therefore, a side-by-side comparison is difficult since they require many anchors whereas we need only a few anchors.

Another major difference between the SFSDP and LPCGD methods and our `SNLSDPclique` algorithm is that they require very sparse problems in order to run efficiently (for example, edge-sparsification heuristics are used in Kim et al. (2009b) and Pong and Tseng (2010) to speed-up their computations), whereas we are able to handle problems with many distance constraints quite efficiently. In fact, we can see from Table 5.4 that having too few distance constraints can be a problem for us since we may not be able to localize all the sensors in the network. Again, this fact makes a side-by-side comparison of our algorithm with the SFSDP and LPCGD methods difficult.

From the results in Pong and Tseng (2010), we see that the best variant of the LPCGD method requires about twice as much CPU time as the times reported in Table 5.4 for problems with up to $10,000$ nodes; moreover, the LPCGD method can only attain RMSD values on the order of 1e-3 for noiseless problems. However, we see in Table 5.4 that our `SNLSDPclique` algorithm attains RMSD values on the order of 1e-16 for noiseless problems, regardless of problem size. For noisy problems with a 1% noise factor, the LPCGD method again attains RMSD values on the order of 1e-3 (see Pong and Tseng (2010)); however, from Table 5.7 we see that our `SNLSDPclique` algorithm can only attain RMSD values on the order of 1e-1 for problems with a 1% noise factor.

From the latest results in Kim et al. (2009b), we see that the SFSDP method is able to attain RMSD values on the order of 1e-3 for two-dimensional problems with noise factors from 1% to 20%; however, the computation times are roughly six minutes for a problem with 6,000 nodes with four anchors at the corners of $[0,1]^2$, three minutes for a problem with 9,000 sensors and 1,000 randomly placed anchors, and nine minutes for a problem with 18,000 sensors and 2,000 randomly placed anchors. The CPU times we obtain for the `SNLSDPclique` algorithm in Table 5.4 are orders of magnitude smaller than the running times for the SFSDP method, although the RMSD values we obtain for noisy problems in Table 5.7 are much larger than those reported in Kim et al. (2009b) for the SFSDP method, unless the noise factor is very small, say less than 1e-6.

In conclusion, we find that our `SNLSDPclique` algorithm clearly outperforms the existing methods in terms of CPU time and accuracy on problems with very low noise. However, our method currently does not do anything special to handle noisy problems (for example, no refinement technique was used in our tests), and at this time we find that it is not competitive with the existing methods for accurately solving problems with medium to high noise. We feel confident that combining our facial reduction algorithm together with techniques for handling noise will produce a code which will be highly competitive in terms of both CPU time and solution accuracy.

5.2 Molecular conformation tests

We now present the results of testing our algorithm on molecular conformation problems. We follow the tests in Biswas et al. (2008). The main purpose here is not to claim we can solve molecular conformation problems well; our purpose is to investigate how our code runs on problems that have a different structure, that are not randomly generated, that come from a real-life source, and that have no anchors.

Molecular data was obtained from the Protein Data Bank Berman et al. (2002). Atomic positions $p_1, \ldots, p_n \in \mathbb{R}^3$ were extracted from the PDB files; lines of the PDB files beginning with `ATOM` or `HETATM` contain the relative coordinates of the atoms in $\mathbb{R}^3$, measured in ångströms, where $1\text{Å} = 10^{-10}\text{m}$. Using these atomic positions, we computed the interatomic distances less than some range $R > 0$; that is, we generate the the $n \times n$ partial Euclidean distance matrix D_p according to

$$(D_p)_{ij} = \begin{cases} \|p_i - p_j\|^2, & \text{if } \|p_i - p_j\| < R \\ \text{unspecified}, & \text{otherwise.} \end{cases}$$

For the noisy tests, we again use the multiplicative noise model (5.1). As mentioned before, NMR can be used to measure interatomic distances in the range of 5–6Å; longer ranges of up to 8Å are possible, but come at a cost of structural modification of the molecule Yuen et al. (2010). Therefore, we have run our tests using $R = 5$Å, 6Å, 7Å, 8Å.

We used the same computing environment as in our sensor network localization tests. Due to the improved accuracy observed for the point representation method in the sensor network localization tests, we have only tested the point representation method here on these molecular conformation instances.

The tables are labelled in a similar manner as in the sensor network localization tests. The only exception is how the Max Error and RMSD values are computed. Since we do not have absolute positions, we measure how well the computed solution fits the true solution by solving a Procrustes problem. That is, we let $(Q, c) \in \mathbb{R}^{r \times r} \times \mathbb{R}^r$ be an optimal solution to the problem

$$\min \left\{ \sum_{i \text{ positioned}} \left\| (Qp_i + c) - p_i^{\text{true}} \right\|^2 : (Q, c) \in \mathbb{R}^{r \times r} \times \mathbb{R}^r,\ Q^T Q = I \right\}.$$

We then compute Max Error and RMSD as follows:

$$\text{Max Error} := \max_{i \text{ positioned}} \|(Qp_i + c) - p_i^{\text{true}}\|_2;$$

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

The results of our tests are as follows.

1. Table 5.8 gives the results of the point representation algorithm using the rigid steps on *noiseless* molecular conformation problems, for various measurement ranges. As we can see, the accuracy is not as good as in the sensor network localization tests. Possible reasons for this are: these molecular conformation problems are not randomly generated and have a much different structure than our previous tests; there are no anchors.

2. Table 5.9 gives the results of the point representation algorithm using the rigid steps on *noisy* molecular conformation problems with range fixed at $R = 7\text{Å}$ for all problems. We chose $R = 7\text{Å}$ here since that ensures all atoms will be positioned, as can be seen from Table 5.8. From these noisy tests we see that some molecules are easier to solve than others. As an example, Figure 5.1 provides graphical representations of the solutions we obtained for the 1RGS protein molecule for different levels of noise. We can see from these images that our algorithm has difficulty solving problems with higher levels of noise.

In conclusion, we see that while the CPU times remain impressively fast for these molecular problems, the accuracy we obtain is not as good as the accuracy obtained in the sensor network localization tests. Comparing our results with the results in Biswas et al. (2008), we find that while our code takes only seconds to solve these problems, the method from Biswas et al. (2008) requires a couple of minutes to more than twenty minutes. However, the RMSD values obtained by Biswas et al. (2008) for problems with 10% noise was on the order of 1e+0; the RMSD values we obtained were on the order of 1e+1 for problems with 10% noise.

Table 5.8: Results of the point representation method on *noiseless* molecular conformation problems, using `RigidCliqueUnion` and `RigidNodeAbsorb`. The values for Average Degree, # Atoms Positioned, CPU Time, Max Error and RMSD are averaged over ten random instances. We have labelled with an asterisk the instances where not all atoms were positioned.

Molecule name	# atoms	R	Avg. Deg.	# Atoms Positioned	CPU Time	Max Error	RMSD
1PTQ	404	8Å	35.2	404.0	0.3 s	4e-14	2e-14
1PTQ	404	7Å	25.9	404.0	0.2 s	3e-14	1e-14
1PTQ	404	6Å	17.8	404.0	0.2 s	4e-14	2e-14
1PTQ	404	5Å	11.0	404.0	0.2 s	3e-13	3e-14
1LFB	641	8Å	35.1	641.0	0.3 s	7e-014	3e-14
1LFB	641	7Å	25.9	641.0	0.3 s	5e-014	3e-14
1LFB	641	6Å	17.8	641.0	0.3 s	6e-014	2e-14
1LFB	641	5Å	10.9	641.0	0.3 s	1e-013	3e-14
1F39	1653	8Å	40.3	1653.0	0.8 s	4e-13	9e-14
1F39	1653	7Å	28.6	1653.0	0.8 s	2e-13	1e-13
1F39	1653	6Å	19.1	1653.0	0.8 s	3e-13	1e-13
1F39	1653	5Å	11.6	1653.0	1.1 s	2e-13	1e-13
1RGS	2059	8Å	39.6	2059.0	1.2 s	2e-13	2e-13
1RGS	2059	7Å	28.6	2059.0	1.1 s	3e-13	2e-13
1RGS	2059	6Å	19.3	2059.0	1.1 s	4e-13	2e-13
1RGS	2059	5Å	11.5	*2054.0	1.2 s	2e-12	3e-13
1HQQ	4116	8Å	44.1	4116.0	2.6 s	3e-13	2e-13
1HQQ	4116	7Å	31.0	4116.0	2.2 s	2e-13	8e-14
1HQQ	4116	6Å	20.4	4116.0	2.2 s	2e-13	8e-14
1HQQ	4116	5Å	12.3	*4114.0	3.1 s	3e-12	3e-13
1I7W	9360	8Å	44.3	9360.0	10 s	1e-12	6e-13
1I7W	9360	7Å	31.4	9360.0	9 s	1e-12	6e-13
1I7W	9360	6Å	21.0	*9359.0	10 s	2e-12	7e-13
1I7W	9360	5Å	12.6	*9350.0	14 s	2e-12	7e-13

Table 5.9: Results of the point representation method on molecular conformation problems with $R = 7$Å, using `RigidCliqueUnion` and `RigidNodeAbsorb`. The values for Average Degree, # Atoms Positioned, CPU Time, Max Error and RMSD are averaged over ten random instances.

	Molecule	#		Avg.	# Atoms	CPU	Max	
σ	name	atoms	R	Deg.	Positioned	Time	Error	RMSD
1e-6	1PTQ	404	7Å	25.9	404.0	0.2 s	1e-04	4e-05
1e-5	1PTQ	404	7Å	25.9	404.0	0.2 s	1e-03	4e-04
1e-4	1PTQ	404	7Å	25.9	404.0	0.2 s	1e-02	4e-03
1e-3	1PTQ	404	7Å	25.9	404.0	0.2 s	1e+01	3e+00
1e-2	1PTQ	404	7Å	25.9	404.0	0.2 s	1e+01	4e+00
1e-1	1PTQ	404	7Å	25.9	404.0	0.2 s	1e+01	4e+00
1e-6	1LFB	641	7Å	25.9	641.0	0.3 s	9e-05	3e-05
1e-5	1LFB	641	7Å	25.9	641.0	0.3 s	9e-04	3e-04
1e-4	1LFB	641	7Å	25.9	641.0	0.3 s	9e-03	3e-03
1e-3	1LFB	641	7Å	25.9	641.0	0.3 s	8e-02	3e-02
1e-2	1LFB	641	7Å	25.9	641.0	0.3 s	2e+00	7e-01
1e-1	1LFB	641	7Å	25.9	641.0	0.3 s	2e+01	8e+00
1e-6	1F39	1653	7Å	28.6	1653.0	0.8 s	3e-04	8e-05
1e-5	1F39	1653	7Å	28.6	1653.0	0.8 s	3e-03	8e-04
1e-4	1F39	1653	7Å	28.6	1653.0	0.8 s	2e+00	1e-01
1e-3	1F39	1653	7Å	28.6	1653.0	0.8 s	1e+01	4e+00
1e-2	1F39	1653	7Å	28.6	1653.0	0.8 s	2e+01	7e+00
1e-1	1F39	1653	7Å	28.6	1653.0	0.8 s	4e+01	1e+01
1e-6	1RGS	2059	7Å	28.6	2059.0	1.1 s	4e-04	1e-04
1e-5	1RGS	2059	7Å	28.6	2059.0	1.1 s	4e-03	1e-03
1e-4	1RGS	2059	7Å	28.6	2059.0	1.1 s	4e-02	1e-02
1e-3	1RGS	2059	7Å	28.6	2059.0	1.1 s	3e+01	1e+01
1e-2	1RGS	2059	7Å	28.6	2059.0	1.1 s	3e+01	1e+01
1e-1	1RGS	2059	7Å	28.6	2059.0	1.1 s	3e+01	2e+01
1e-6	1HQQ	4116	7Å	31.0	4116.0	2.2 s	4e-04	1e-04
1e-5	1HQQ	4116	7Å	31.0	4116.0	2.2 s	1e+00	3e-01
1e-4	1HQQ	4116	7Å	31.0	4116.0	2.2 s	2e+01	4e+00
1e-3	1HQQ	4116	7Å	31.0	4116.0	2.2 s	3e+01	6e+00
1e-2	1HQQ	4116	7Å	31.0	4116.0	2.2 s	4e+01	1e+01
1e-1	1HQQ	4116	7Å	31.0	4116.0	2.2 s	5e+01	2e+01
1e-6	1I7W	9360	7Å	31.4	9360.0	12 s	2e-03	4e-04
1e-5	1I7W	9360	7Å	31.4	9360.0	12 s	2e-02	4e-03
1e-4	1I7W	9360	7Å	31.4	9360.0	12 s	1e+01	2e+00
1e-3	1I7W	9360	7Å	31.4	9360.0	12 s	4e+01	1e+01
1e-2	1I7W	9360	7Å	31.4	9360.0	12 s	6e+01	2e+01
1e-1	1I7W	9360	7Å	31.4	9360.0	12 s	9e+01	4e+01

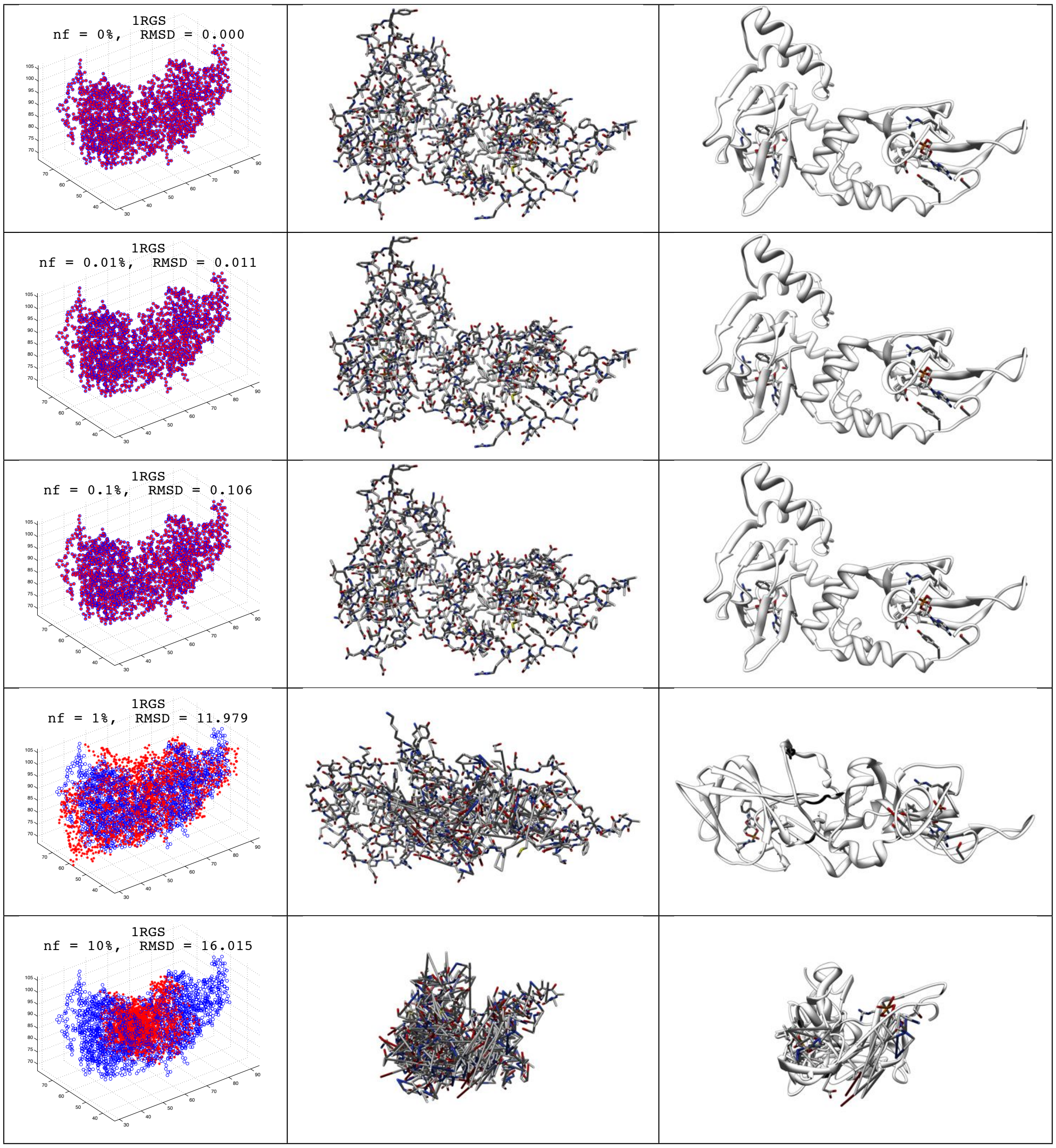


Figure 5.1: Three dimensional plots of the solutions obtained for the 1RGS protein (2059 atoms) with $R = 7$Å and for different levels of noise. The plots in the first column indicate the true positions (blue circles) and the computed positions (red asterisks) of the atoms. The atomic bonds are indicated in the second column with the computed positions of the atoms. The last column contains the corresponding ribbon plots, used for picturing the structure of the molecule.
Molecular graphics images in the second and third columns were produced using the UCSF Chimera package from the Resource for Biocomputing, Visualization, and Informatics at the University of California, San Francisco (supported by NIH P41 RR-01081); see Pettersen et al. (2004) for more information.

Chapter 6

Conclusions and Future Work

We have developmed a theory of semidefinite facial reduction for the Euclidean distance matrix completion problem. This theory is based on our novel result showing a close connection between cliques in the graph of the partial Euclidean distance matrix and faces of the semidefinite cone containing the feasible set of the semidefinite relaxation. This theory allows us to dramatically reduce the number of variables and constraints required to represent this semidefinite feasible set. Under certain circumstances, we are even able to use this facial reduction technique to determine that there is a single feasible solution, and that this solution can be computed without the need for a semidefinite optimization solver. Furthermore, we have used this theory to develop a highly efficient algorithm capable of solving many very large-scale Euclidean distance matrix completion problems exactly. For problems with a low level of noise, our `SNLSDPclique` Algorithm 6 clearly outperforms existing algorithms in terms of CPU time and accuracy.

Our facial reduction results also have wide applicability beyond the development of our `EDMSDPclique` Algorithm 1 and specialized `SNLSDPclique` Algorithm 6. Indeed, we have already seen our results on disjoint subset facial reduction Theorem 4.5 and distance constraint reduction Theorem 4.16 applied with great success to reduce the size of semidefinite optimization problems in the machine learning area of manifold unfolding; see our paper Alipanahi et al. (2010). Our theory can therefore be used to aid in the solution of much larger semidefinite relaxations of Euclidean distance matrix problems than was previously possible. We hope this contribution proves to be very useful in the applied areas mentioned in Chapter 1, and many others.

6.1 Future work

The results from this thesis have opened up many possibilities for future work.

1. *Noisy problems*: We would like to investigate techniques for handling noisy problems, such as the refinement techniques discussed in the literature, and

combine this with our facial reduction theory to be able to solve very large problems in less time and with more accuracy. We have already mentioned some ideas in this direction, such as the working with the nonlinear semidefinite optimization problem (5.2), or allowing the rank of the approximate Gram matrix of a noisy Euclidean distance matrix to increase. This latter approach would then give us higher dimensional faces. In the algorithms presented here, we strictly maintained the dimension of the faces by ensuring that the $\bar{U}$ part of U in Theorem 4.4 always had $r+1$ columns; indeed, the intersection lemmas we give here (for example, Lemma 4.11) assume that $\bar{U}_1$ and $\bar{U}_2$ both have $r+1$ columns. With this more general approach, it will no longer be the case that all computed faces have $r+1$ columns for $\bar{U}$. Therefore, we must generalize the intersection lemmas to find expressions for intersecting two faces where $\bar{U}_1$ and $\bar{U}_2$ have a different number of columns. In the end, we hope to have computed a face containing our feasible set, but that is more flexible in terms of the noisy distances. Combining this higher facial dimension approach with an appropriate least-squares refinement technique holds promise for solving very large noisy problems accurately.

2. *Non-uniquely localizable problems*: To find a localization of a non-uniquely localizable problem in a low dimensional space Weinberger et al. (2004) and Biswas et al. (2006) have suggested the regularization technique of maximizing the trace of the Gram matrix in the semidefinite relaxation in order to encourage the points to spread out and "flatten" the graph. We plan to investigate the use of our facial reduction procedure together with such a regularization technique. In this case, our facial reduction would be a preprocessor to reduce the size of the problem before passing it to a semidefinite optimization solver. However, since sensor network localization of non-uniquely localizable problems is NP-hard, we do not expect to be able to solve every problem efficiently with this regularization technique; however, we do expect to get improved results for a wide class of problems. It will also be interesting to compare this regularization technique to the nuclear norm minimization heuristic for rank minimization.

3. *Chordal extensions*: There is a close connection between the results we have developed here and the results of Bakonyi and Johnson (1995) on Euclidean distance matrix completion problems with chordal graphs. Our technique focusses on cliques in the graph of the partial Euclidean distance matrix. It is well known that determining the maximal cliques in a chordal graph is very inexpensive. This connection may open up some interesting results in the area of Euclidean distance matrix completion.

4. *Molecular conformation and multi-dimensional scaling*: We would like to return to the problem of molecular conformation to see if our facial reduction approach with regularization and noise-handling techniques can improve our numerical results. In addition, the area of multi-dimensional scaling holds

much promise for our facial reduction technique, as seen in the successful application of our results to the problem of manifold unfolding.

5. *Extending facial reduction to other matrix completion problems*: It seems possible to extend our results to the problem of correlation matrix completion. A correlation matrix is a positive semidefinite matrix with ones on the diagonal. Other matrix completion problems may also benefit from our results here.

We look forward to investigating these and other questions arising from this study on facial reduction for the semidefinite relaxation of the Euclidean distance matrix completion problem.

References

Akyildiz, I., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422. 5

Al-Homidan, S. and Wolkowicz, H. (2005). Approximate and exact completion problems for Euclidean distance matrices using semidefinite programming. *Linear Algebra Appl.*, 406:109–141. 4, 25, 26, 27, 29, 31

Alfakih, A. Y. (2000). Graph rigidity via Euclidean distance matrices. *Linear Algebra Appl.*, 310(1-3):149–165. 4

Alfakih, A. Y. (2001). On rigidity and realizability of weighted graphs. *Linear Algebra Appl.*, 325(1-3):57–70. 4

Alfakih, A. Y. (2003). On the uniqueness of Euclidean distance matrix completions. *Linear Algebra Appl.*, 370:1–14. 3, 39

Alfakih, A. Y. (2005). On the uniqueness of Euclidean distance matrix completions: the case of points in general position. *Linear Algebra Appl.*, 397:265–277. 3, 39

Alfakih, A. Y. (2006a). On the nullspace, the rangespace and the characteristic polynomial of Euclidean distance matrices. *Linear Algebra Appl.*, 416(2-3):348–354. 4

Alfakih, A. Y. (2006b). A remark on the faces of the cone of Euclidean distance matrices. *Linear Algebra Appl.*, 414(1):266–270. 4

Alfakih, A. Y. (2007). On dimensional rigidity of bar-and-joint frameworks. *Discrete Appl. Math.*, 155(10):1244–1253. 4

Alfakih, A. Y., Anjos, M. F., Piccialli, V., and Wolkowicz, H. (2009). Euclidean distance matrices, semidefinite programming, and sensor network localization. Technical report, University of Waterloo. 4, 40

Alfakih, A. Y., Khandani, A., and Wolkowicz, H. (1999). Solving Euclidean distance matrix completion problems via semidefinite programming. *Comput. Optim. Appl.*, 12(1-3):13–30. 4, 31, 110

Alfakih, A. Y. and Wolkowicz, H. (2002). Two theorems on Euclidean distance matrices and Gale transform. *Linear Algebra Appl.*, 340:149–154. 4

Alipanahi, B., Ghodsi, A., and Krislock, N. (2010). Manifold unfolding by Euclidean distance matrix completion. Submitted to the 2010 AAAI Conference on Artificial Intelligence. 119

Alizadeh, F. (1995). Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5(1):13–51. 2

Ames, B. P. and Vavasis, S. A. (2009). Nuclear norm minimization for the planted clique and biclique problems. Technical report, University of Waterloo. 46

An, L. T. H. and Tao, P. D. (2003). Large-scale molecular optimization from distance matrices by a D.C. optimization approach. *SIAM Journal on Optimization*, 14(1):77–114. 6

Andersen, E. D. and Andersen, K. D. (1995). Presolving in linear programming. *Mathematical Programming*, 71(2):221–245. 63

Aspnes, J., Eren, T., Goldenberg, D., Morse, A., Whiteley, W., Yang, Y., Anderson, B., and Belhumeur, P. (2006). A theory of network localization. *IEEE Transactions on Mobile Computing*, 5(12):1663–1678. 5, 41, 56

Aspnes, J., Goldenberg, D., and Yang, Y. (2004). On the computational complexity of sensor network localization. *Lecture Notes in Computer Science*, 3121:32–44. 5, 41, 56

Bădoiu, M., Demaine, E. D., Hajiaghayi, M., and Indyk, P. (2006). Low-dimensional embedding with extra information. *Discrete Comput. Geom.*, 36(4):609–632. 41

Bakonyi, M. and Johnson, C. R. (1995). The Euclidean distance matrix completion problem. *SIAM Journal on Matrix Analysis and Applications*, 16(2):646–654. 3, 37, 38, 120

Barker, G. and Carlson, D. (1975). Cones of diagonally dominant matrices. *Pacific J. of Math.*, 57:15–32. 4, 21

Barker, G. P. (1973). The lattice of faces of a finite dimensional cone. *Linear Algebra and Appl.*, 7:71–82. 4, 21

Barker, G. P. (1978). Faces and duality in convex cones. *Linear and Multilinear Algebra*, 6(3):161–169. 4

Barvinok, A. (1995). Problems of distance geometry and convex properties of quadratic maps. *Discrete and Computational Geometry*, 13(1):189–202. 4

Beck, A., Stoica, P., and Li, J. (2008a). Exact and approximate solutions of source localization problems. *Signal Processing, IEEE Transactions on*, 56(5):1770–1778. 5

Beck, A., Teboulle, M., and Chikishev, Z. (2008b). Iterative minimization schemes for solving the single source localization problem. *SIAM Journal on Optimization*, 19(3):1397–1416. 5

Belk, M. (2007). Realizability of graphs in three dimensions. *Discrete Comput. Geom.*, 37(2):139–162. 4

Belk, M. and Connelly, R. (2007). Realizability of graphs. *Discrete Comput. Geom.*, 37(2):125–137. 4

Ben-Israel, A. and Greville, T. N. E. (2003). *Generalized Inverses: Theory and Applications*. Wiley-Interscience, 2nd edition. 14

Berman, A. (1973). *Cones, matrices and mathematical programming*. Springer-Verlag, Berlin. Lecture Notes in Economics and Mathematical Systems, Vol. 79. 16, 34

Berman, A. and Ben-Israel, A. (1971). Linear inequalities, mathematical programming and matrix theory. *Mathematical Programming*, 1(1):291–300. 16

Berman, A. and Plemmons, R. J. (1979). *Nonnegative Matrices in the Mathematical Sciences*. Academic Press. 32

Berman, H. M., Battistuz, T., Bhat, T. N., Bluhm, W. F., Bourne, P. E., Burkhardt, K., Feng, Z., Gilliland, G. L., Iype, L., Jain, S., Fagan, P., Marvin, J., Padilla, D., Ravichandran, V., Schneider, B., Thanki, N., Weissig, H., Westbrook, J. D., and Zardecki, C. (2002). The Protein Data Bank. *Acta Crystallographica Section D*, 58(6):899–907. 61, 114

Biswas, P. (2007). *Semidefinite programming approaches to distance geometry problems*. PhD thesis, Stanford University. 5, 6, 54

Biswas, P., Liang, T.-C., Toh, K.-C., Wang, T.-C., and Ye, Y. (2006). Semidefinite programming approaches for sensor network localization with noisy distance measurements. *IEEE Transactions on Automation Science and Engineering*, 3:360–371. 5, 44, 54, 62, 98, 108, 120

Biswas, P., Toh, K.-C., and Ye, Y. (2008). A distributed SDP approach for large-scale noisy anchor-free graph realization with applications to molecular conformation. *SIAM Journal on Scientific Computing*, 30(3):1251–1277. 6, 54, 114, 115

Biswas, P. and Ye, Y. (2004). Semidefinite programming for ad hoc wireless sensor network localization. In *IPSN '04: Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 46–54, New York, NY, USA. ACM. 5, 54, 98, 108

Biswas, P. and Ye, Y. (2006). 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. 5, 54

Björck, Å. (1996). *Numerical Methods for Least Squares Problems.* SIAM, Philadelphia. 11, 57

Blumenthal, L. M. (1970). *Theory and applications of distance geometry.* Chelsea Pub. Co. 4

Borwein, J. M. and Lewis, A. S. (2000). *Convex Analysis and Nonlinear Optimization : Theory and Examples.* Springer-Verlag. 15

Borwein, J. M. and Wolkowicz, H. (1981a). Characterization of optimality for the abstract convex program with finite-dimensional range. *J. Austral. Math. Soc. Ser. A*, 30(4):390–411. 4, 21

Borwein, J. M. and Wolkowicz, H. (1981b). Facial reduction for a cone-convex programming problem. *J. Austral. Math. Soc. Ser. A*, 30(3):369–380. 4, 21

Borwein, J. M. and Wolkowicz, H. (1981c). Regularizing the abstract convex program. *J. Math. Anal. Appl.*, 83(2):495–530. 4, 21

Brearley, A. L., Mitra, G., and Williams, H. P. (1975). Analysis of mathematical programming problems prior to applying the simplex algorithm. *Mathematical Programming*, 8(1):54–83. 63

Bruck, J., Gao, J., and Jiang, A. (2009). Localization and routing in sensor networks by local angle information. *ACM Trans. Sen. Netw.*, 5(1):1–31. 5

Bulusu, N., Heidemann, J., and Estrin, D. (2000). GPS-less low-cost outdoor localization for very small devices. *Personal Communications, IEEE*, 7(5):28–34. 5

Candès, E. J. and Plan, Y. (2009). Matrix completion with noise. Submitted to Proceedings of the IEEE. 4, 46

Candès, E. J. and Recht, B. (2008). Exact matrix completion via convex optimization. Technical report, Caltech. 4, 45

Carter, M. W., Jin, H. H., Saunders, M. A., and Ye, Y. (2006). SpaseLoc: An adaptive subproblem algorithm for scalable wireless sensor network localization. *SIAM Journal on Optimization*, 17(4):1102–1128. 5

Cassioli, A. (2009). Solving the sensor network localization problem using an heuristic multistage approach. Technical report, Università degli Studi di Firenze. 5

Cheung, Y.-L., Schurr, S. P., and Wolkowicz, H. (2010). Numerical solution of ill-posed semidefinite programs. Technical report, University of Waterloo, Waterloo, Ontario. In progress. 21, 22, 64

Chua, C. and Tunçel, L. (2008). Invariance and efficiency of convex representations. *Mathematical Programming*, 111(1):113–140. 54

Connelly, R. (2005). Generic global rigidity. *Discrete Comput. Geom.*, 33(4):549–563. 4, 40

Costa, J. A., Patwari, N., and Hero, III, A. O. (2006). Distributed weighted-multidimensional scaling for node localization in sensor networks. *ACM Trans. Sen. Netw.*, 2(1):39–64. 5

Crippen, G. M. (1991). Chemical distance geometry: Current realization and future projection. *Journal of Mathematical Chemistry*, 6(1):307–324. 6

Crippen, G. M. and Havel, T. F. (1988). *Distance Geometry and Molecular Conformation*, volume 15 of *Chemometrics Series*. Research Studies Press Ltd., Chichester. 6

Critchley, F. (1988). On certain linear mappings between inner-product and squared distance matrices. *Linear Algebra Appl.*, 105:91–107. 3, 25, 32, 33, 35

Dattorro, J. (2008). *Convex Optimization & Euclidean Distance Geometry.* Meboo Publishing USA. 4, 33, 36

Ding, Y., Krislock, N., Qian, J., and Wolkowicz, H. (2008a). Sensor network localization, Euclidean distance matrix completions, and graph realization. In *MELT '08: Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, pages 129–134, New York, NY, USA. ACM. 54

Ding, Y., Krislock, N., Qian, J., and Wolkowicz, H. (2008b). Sensor network localization, Euclidean distance matrix completions, and graph realization. *Optimization and Engineering*, published online. 54, 60

Doherty, L., Pister, K. S. J., and El Ghaoui, L. (2001). Convex position estimation in wireless sensor networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1655–1663 vol.3. 4

Dong, Q. and Wu, Z. (2002). A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances. *Journal of Global Optimization*, 22(1):365–375. 6

Dong, Q. and Wu, Z. (2003). A geometric build-up algorithm for solving the molecular distance geometry problem with sparse distance data. *Journal of Global Optimization*, 26(3):321–333. 6

dos Santos Carvalho, R., Lavor, C., and Protti, F. (2008). Extending the geometric build-up algorithm for the molecular distance geometry problem. *Information Processing Letters*, 108(4):234 – 237. 6

Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218. 57

Emiris, I. Z. and Nikitopoulos, T. G. (2005). Molecular conformation search by distance matrix perturbations. *J. Math. Chem.*, 37(3):233–253. 6

Eren, T., Goldenberg, O., Whiteley, W., Yang, Y., Morse, A., Anderson, B., and Belhumeur, P. (2004). Rigidity, computation, and randomization in network localization. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2673–2684. 4

Farebrother, R. W. (1987). Three theorems with applications to Euclidean distance matrices. *Linear Algebra Appl.*, 95:11–16. 3

Fazel, M. (2002). *Matrix Rank Minimization with Applications.* PhD thesis, Stanford University. 4, 45

Fazel, M., Hindi, H., and Boyd, S. (2003). Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices. In *Proceedings of the American Control Conference*, pages 2156–2162. 4, 46

Fukuda, M., Kojima, M., Murota, K., and Nakata, K. (2001). Exploiting sparsity in semidefinite programming via matrix completion i: General framework. *SIAM Journal on Optimization*, 11(3):647–674. 5

Glunt, W., Hayden, T., and Raydan, M. (1993). Molecular conformations from distance matrices. *Journal of Computational Chemistry*, 14(1):114–120. 6

Glunt, W., Hayden, T. L., Hong, S., and Wells, J. (1990). An alternating projection algorithm for computing the nearest Euclidean distance matrix. *SIAM Journal on Matrix Analysis and Applications*, 11(4):589–600. 4

Goemans, M. X. and Williamson, D. P. (1995). Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145. 56

Goldfarb, D. and Scheinberg, K. (1998). Interior point trajectories in semidefinite programming. *SIAM Journal on Optimization*, 8(4):871–886. 57

Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations.* Johns Hopkins University Press, Baltimore, MD, 3rd edition. 11, 49

Göring, F., Helmberg, C., and Reiss, S. (2009). Graph realizations associated with minimizing the maximum eigenvalue of the laplacian. Technical report, Fakultät für Mathematik, Technischen Universität Chemnitz, Chemnitz, Germany. Preprint. 46

Göring, F., Helmberg, C., and Wappler, M. (2008). The rotational dimension of a graph. Technical report, Fakultät für Mathematik, Technischen Universität Chemnitz, Chemnitz, Germany. Preprint. 46

Gower, J. C. (1982). Euclidean distance geometry. *Math. Sci.*, 7(1):1–14. 3

Gower, J. C. (1984). Distance matrices and their Euclidean approximation. In *Data analysis and informatics, III (Versailles, 1983)*, pages 3–21. North-Holland, Amsterdam. 3

Gower, J. C. (1985). Properties of Euclidean and non-Euclidean distance matrices. *Linear Algebra Appl.*, 67:81–97. 3

Green, B. (1952). The orthogonal approximation of an oblique structure in factor analysis. *Psychometrika*, 17(4):429–440. 49

Grone, R., Johnson, C. R., Sá, E. M., and Wolkowicz, H. (1984). Positive definite completions of partial Hermitian matrices. *Linear Algebra and its Applications*, 58:109–124. 3, 38

Grooms, I. G., Lewis, R. M., and Trosset, M. W. (2009). Molecular embedding via a second order dissimilarity parameterized approach. *SIAM Journal on Scientific Computing*, 31(4):2733–2756. 6

Güler, O. and Ye, Y. (1993). Convergence behavior of interior-point algorithms. *Mathematical Programming*, 60(1):215–228. 58

Halická, M., de Klerk, E., and Roos, C. (2002). On the convergence of the central path in semidefinite optimization. *SIAM Journal on Optimization*, 12(4):1090–1099. 58

Havel, T., Kuntz, I., and Crippen, G. (1983). The theory and practice of distance geometry. *Bulletin of Mathematical Biology*, 45(5):665–720. 6

Havel, T. F. (2003). Metric matrix embedding in protein structure calculations, NMR spectra analysis, and relaxation theory. *Magnetic Resonance in Chemistry*, 41:S37–S50. 6

Hayden, T. L., Wells, J., Liu, W. M., and Tarazaga, P. (1991). The cone of distance matrices. *Linear Algebra Appl.*, 144:153–169. 4, 33

Hendrickson, B. (1990). *The Molecule Problem: Determining Conformation from Pairwise Distances*. PhD thesis, Cornell University. 6

Hendrickson, B. (1992). Conditions for unique graph realizations. *SIAM Journal on Computing*, 21(1):65–84. 4, 40

Hendrickson, B. (1995). The molecule problem: Exploiting structure in global optimization. *SIAM Journal on Optimization*, 5(4):835–857. 4

Higham, N. J. (1986). Computing the polar decomposition—with applications. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1160–1174. 49

Jackson, B. and Jordán, T. (2005). Connected rigidity matroids and unique realizations of graphs. *J. Combin. Theory Ser. B*, 94(1):1–29. 4

Jin, H. H. (2005). *Scalable Sensor Localization Algorithms for Wireless Sensor Networks*. PhD thesis, University of Toronto, Toronto, Ontario, Canada. (Joint research conducted at Stanford University). 5

Johnson, C. R., Kroschel, B., and Wolkowicz, H. (1998). An interior-point method for approximate positive semidefinite completions. *Comput. Optim. Appl.*, 9(2):175–190. 3

Johnson, C. R. and Tarazaga, P. (1995). Connections between the real positive semidefinite and distance matrix completion problems. *Linear Algebra Appl.*, 223/224:375–391. Special issue honoring Miroslav Fiedler and Vlastimil Pták. 3, 25

Karp, R. M. (1972). Reducibility among combinatorial problems. In *Complexity of computer computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972)*, pages 85–103. Plenum, New York. 40

Kim, S., Kojima, M., and Waki, H. (2009a). Exploiting sparsity in SDP relaxation for sensor network localization. *SIAM Journal on Optimization*, 20(1):192–215. 5, 98, 109, 110

Kim, S., Kojima, M., Waki, H., and Yamashita, M. (2008). User manual for SFSDP: a Sparse version of Full SemiDefinite Programming relaxation for sensor network localization problems. Technical report, Tokyo Institute of Technology. Revised July 2009. 110

Kim, S., Kojima, M., Waki, H., and Yamashita, M. (2009b). SFSDP: a Sparse version of Full SemiDefinite Programming relaxation for sensor network localization problems. Technical report, Tokyo Institute of Technology. 5, 98, 109, 110, 113

Krislock, N. (2003). Numerical solution of semidefinite least squares problems. Master's thesis, University of British Columbia. 16

Krislock, N. and Wolkowicz, H. (2010). Explicit sensor network localization using semidefinite representations and facial reductions. Technical report, University of Waterloo. 5

Kumar, P. S. and Madhavan, C. V. (1998). Minimal vertex separators of chordal graphs. *Discrete Appl. Math.*, 89:155–168. 38, 39

Laurent, M. (1998). A tour d'horizon on positive semidefinite and Euclidean distance matrix completion problems. In *Topics in Semidefinite and Interior-Point Methods*, pages 51–76. Amer. Math. Soc. 4

Laurent, M. (2001). Polynomial instances of the positive semidefinite and Euclidean distance matrix completion problems. *SIAM Journal on Matrix Analysis and Applications*, 22(3):874–894. 3, 41

Leung, N.-H. Z. and Toh, K.-C. (2008). An SDP-based divide-and-conquer algorithm for large scale noisy anchor-free graph realization. Technical report, Department of Mathematics, National University of Singapore. 6

Li, X.-Y. (2008). *Wireless Ad Hoc and Sensor Networks: Theory and Applications.* Cambridge University Press. 5

Megerian, S., Koushanfar, F., Potkonjak, M., and Srivastava, M. (2005). Worst and best-case coverage in sensor networks. *Mobile Computing, IEEE Transactions on*, 4(1):84–92. 5

Mészáros, C. and Suhl, U. H. (2003). Advanced preprocessing techniques for linear and quadratic programming. *OR Spectrum*, 25(4):575–595. 63

Moore, D., Leonard, J., Rus, D., and Teller, S. (2004). Robust distributed network localization with noisy range measurements. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 50–61, New York, NY, USA. ACM. 5

Moré, J. J. and Wu, Z. (1997). Global continuation for distance geometry problems. *SIAM Journal on Optimization*, 7(3):814–836. 6

Moré, J. J. and Wu, Z. (1999). Distance geometry optimization for protein structures. *Journal of Global Optimization*, 15(3):219–234. 6

Muramatsu, M. (2005). A unified class of directly solvable semidefinite programming problems. *Ann. Oper. Res.*, 133:85–97. 88

Nawaz, S. (2008). *Anchor Free Localization for Ad-hoc Wireless Sensor Networks.* PhD thesis, University of New South Wales. 5

Nesterov, Y. and Nemirovskii, A. (1994). *Interior-point polynomial algorithms in convex programming*, volume 13 of *SIAM Studies in Applied Mathematics.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA. 2

Nie, J. (2009). Sum of squares method for sensor network localization. *Computational Optimization and Applications*, 43(2):151–179. 5

Pataki, G. (2000). A simple derivation of a facial reduction algorithm and extended dual systems. Technical report, University of North Carolina at Chapel Hill. 4

Pettersen, E. F., Goddard, T. D., Huang, C. C., Couch, G. S., Greenblatt, D. M., Meng, E. C., and Ferrin, T. E. (2004). UCSF Chimera – A visualization system for exploratory research and analysis. *Journal of Computational Chemistry*, 25(13):1605–1612. 118

Poljak, S., Rendl, F., and Wolkowicz, H. (1995). A recipe for semidefinite relaxation for (0,1)-quadratic programming. *Journal of Global Optimization*, 7(1):51–73. 42

Pong, T. and Tseng, P. (2010). (Robust) edge-based semidefinite programming relaxation of sensor network localization. *Mathematical Programming*. Published online. 5, 98, 109, 110, 113

Ramana, M. V., Tunçel, L., and Wolkowicz, H. (1997). Strong duality for semidefinite programming. *SIAM Journal on Optimization*, 7(3):641–662. 4, 16, 21, 35, 64

Recht, B., Fazel, M., and Parrilo, P. A. (2008a). Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. To appear in SIAM Review. 4, 45

Recht, B., Xu, W., and Hassibi, B. (2008b). Necessary and sufficient conditions for success of the nuclear norm heuristic for rank minimization. 4, 45

Rockafellar, R. T. (1970). *Convex Analysis*. Princeton University Press. 15, 16

Savvides, A., Han, C.-C., and Strivastava, M. B. (2001). Dynamic fine-grained localization in ad-hoc networks of sensors. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179, New York, NY, USA. ACM. 5

Saxe, J. B. (1979). Embeddability of weighted graphs in k-space is strongly NP-hard. In *Proceedings of the 17th Allerton Conference on Communications, Control, and Computing*, pages 480–489. 4, 40, 41

Schneider, R. (1993). *Convex bodies: the Brunn-Minkowski theory*. Cambridge University Press. 4, 18

Schoenberg, I. J. (1935). Remarks to Maurice Fréchet's article "Sur la définition axiomatique d'une classe d'espace distanciés vectoriellement applicable sur l'espace de Hilbert". *Ann. of Math. (2)*, 36(3):724–732. 1, 3, 25, 32

Schönemann, P. (1966). A generalized solution of the orthogonal Procrustes problem. *Psychometrika*, 31(1):1–10. 49

So, A. M.-C. (2007). *A Semidefinite Programming Approach to the Graph Realization Problem: Theory, Applications and Extensions*. PhD thesis, Computer Science Department, Stanford University. 4, 40

So, A. M.-C. and Ye, Y. (2006). A semidefinite programming approach to tensegrity theory and realizability of graphs. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 766–775, New York, NY, USA. ACM. 4

So, A. M.-C. and Ye, Y. (2007). Theory of semidefinite programming for sensor network localization. *Math. Program.*, 109(2-3, Ser. B):367–384. 4, 56

Stoyanova, T., Kerasiotis, F., Prayati, A., and Papadopoulos, G. (2009). Evaluation of impact factors on RSS accuracy for localization and tracking applications in sensor networks. *Telecommunication Systems*, 42(3-4, Sp. Iss. SI):235–248. 5

Tarazaga, P. (2005). Faces of the cone of Euclidean distance matrices: characterizations, structure and induced geometry. *Linear Algebra Appl.*, 408:1–13. 4

Tarazaga, P., Hayden, T. L., and Wells, J. (1996). Circum-Euclidean distance matrices and faces. *Linear Algebra Appl.*, 232:77–96. 4

Todd, M. J. (2001). Semidefinite optimization. *Acta Numerica*, 10:515–560. 2

Tseng, P. (2007). Second-order cone programming relaxation of sensor network localization. *SIAM Journal on Optimization*, 18(1):156–185. 5, 98, 109

Tunçel, L. and Wolkowicz, H. (2008). Strong duality and minimal representations for cone optimization. Technical report, University of Waterloo. 35

Tunçel, L. (2001). On the Slater condition for the SDP relaxations of nonconvex sets. *Oper. Res. Lett.*, 29(4):181–186. 64

Vandenberghe, L. and Boyd, S. (1996). Semidefinite programming. *SIAM Review*, 38(1):49–95. 2

Vanderbei, R. J. and Yang, B. (1995). The simplest semidefinite programs are trivial. *Math. Oper. Res.*, 20(3):590–596. 88

Waki, H. and Muramatsu, M. (2009a). A facial reduction algorithm for finding sparse SOS representations. Technical Report CS-09-02, The University of Electro-Communications. 4

Waki, H. and Muramatsu, M. (2009b). Facial reduction algorithms for conic optimization problems. Technical Report CS-09-01, The University of Electro-Communications. 4

Wang, Z., Zheng, S., Ye, Y., and Boyd, S. (2008). Further relaxations of the semidefinite programming approach to sensor network localization. *SIAM Journal on Optimization*, 19(2):655–673. 5, 98, 109, 110

Weinberger, K. Q., Sha, F., and Saul, L. K. (2004). Learning a kernel matrix for nonlinear dimensionality reduction. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 106, New York, NY, USA. ACM. 44, 45, 62, 120

Wolkowicz, H. (1996). Explicit solutions for interval semidefinite linear programs. *Linear Algebra Appl.*, 236:95–104. 88

Wu, D. and Wu, Z. (2007). An updated geometric build-up algorithm for solving the molecular distance geometry problems with sparse distance data. *Journal of Global Optimization*, 37(4):661–673. 6

Wu, D., Wu, Z., and Yuan, Y. (2008). Rigid versus unique determination of protein structures with geometric buildup. *Optimization Letters*, 2(3):319–331. 6

Yang, Z., Liu, Y., and Li, X.-Y. (2009). Beyond trilateration: On the localizability of wireless ad-hoc networks. In *INFOCOM 2009, IEEE*, pages 2392–2400. 5

Yemini, Y. (1979). Some theoretical aspects of position-location problems. In *20th Annual Symposium on Foundations of Computer Science (San Juan, Puerto Rico, 1979)*, pages 1–8. IEEE, New York. 4, 40

Young, G. and Householder, A. (1938). Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3(1):19–22. 3, 25

Yuen, A. K. L., Lafon, O., Charpentier, T., Roy, M., Brunet, F., Berthault, P., Sakellariou, D., Robert, B., Rimsky, S., Pillon, F., Cintrat, J.-C., and Rousseau, B. (2010). Measurement of long-range interatomic distances by solid-state tritium-NMR spectroscopy. *Journal of the American Chemical Society*, 132(6):1734–1735. 60, 114

Zhao, Q., Karisch, S. E., Rendl, F., and Wolkowicz, H. (1998). Semidefinite programming relaxations for the quadratic assignment problem. *Journal of Combinatorial Optimization*, 2(1):71–109. 4

Index