A SEMIDEFINITE PROGRAMMING APPROACH TO THE GRAPH REALIZATION PROBLEM: THEORY, APPLICATIONS AND EXTENSIONS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Anthony Man–Cho So

June 2007

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Yinyu Ye) Principal Co-Advisor

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Rajeev Motwani) Principal Co-Advisor

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Vladlen Koltun)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Tim Roughgarden)

Approved for the University Committee on Graduate Studies.

Abstract

It is a trivial matter to see that given the coordinates of n points in $\mathbb{R}^k$, the distance between any two points can be computed efficiently. However, the inverse problem — given a subset of interpoint distances, find the coordinates of points (called a *realization*) in $\mathbb{R}^k$ (where $k \geq 1$ is fixed) that fit those distances — turns out to be anything but trivial. In fact, this problem has been shown to be NP–hard for any fixed $k \geq 1$. On the other hand, this problem arises from many applications, e.g., surveying, satellite ranging, sensor network localization and molecular conformation, just to name a few. Thus, many heuristics have been proposed. However, they either do not have any theoretical guarantees, or they work only for some very restricted classes of instances.

Recently, Biswas and Ye (2004) have proposed a semidefinite programming (SDP) based model for the problem and have reported its superb experimental performance. Our work is motivated by the desire to explain this phenomenon in a rigorous manner. We begin by showing that the SDP model can be used to find a realization in the required dimension if the input instance satisfies a certain uniqueness property. This uniqueness property has a straightforward geometric interpretation, and it allows us to identify a large class of efficiently realizable instances. Furthermore, it allows us to make some interesting connections with various notions in the rigidity theory of graphs.

Next, we consider a variant of the SDP model and discuss its connection with the theory of tensegrities in discrete geometry. In particular, we show how the theory of SDP can be used as an alternative proof technique for problems in tensegrity theory. As a consequence of this connection, we are able to obtain qualitatively improved and constructive proofs for some results in tensegrity theory.

Finally, we consider an extension of the SDP model and study the problem of finding a low–rank approximate solution to a system of linear matrix equations. We show that a simple randomized polynomial–time procedure produces a low–rank solution that has provably good approximation qualities. Our result provides a unified treatment of and generalizes several well–known results in the literature. In particular, it contains as special cases the Johnson–Lindenstrauss lemma on dimensionality reduction, results on low–distortion embeddings into low–dimensional Euclidean space, and approximation results on certain quadratic optimization problems.

Acknowledgments

On December 4, 1999, while I was an undergraduate student at Princeton University, I bought two books on optimization: one is George Dantzig's *Linear Programming and Extensions*, and the other is Yinyu Ye's *Interior Point Algorithms: Theory and Analysis*. Back then I did not know that Professor Dantzig was Professor Ye's doctoral advisor, nor would I have imagined that five years later I will have the great fortune of having Professor Ye as my doctoral advisor. I would like to express my heartfelt thanks to Professor Ye, for his guidance and constant encouragement and support, as well as the many enjoyable discussions we had over the last four years. I am greatly indebted to him.

I have benefitted tremendously from many great teachers at Stanford University. I am especially grateful to Professors Vladlen Koltun and Tim Roughgarden, who got me excited about various topics in theoretical computer science and generously gave their time to discuss research problems with me and to serve on all my committees (algorithm qual committee, dissertation reading committee, and orals committee). I thank Professor Leonidas Guibas for introducing me to the field of rigidity theory and motivating in part the research done in this thesis. I would also like to thank Professors Tze Leung Lai, Rajeev Motwani and Michael Saunders for serving on my orals committee.

I am grateful to Professor Shuzhong Zhang of the Chinese University of Hong Kong, Professors Sydney Chu and Wenan Zang of the University of Hong Kong, as well as Professor Mordecai Golin of the Hong Kong University of Science and Technology, for hosting me at their respective institutions during the summers of 2004–2006.

It is fair to say that this project would not be possible without the encouragement and support of many friends, and I would like to express my sincerest thanks to them all. My special thanks go to Joanne Chan, Michael Eichmair, Mike Feng, Eleana Fong, Samuel Ieong, Bob Liu, Alfred Mak, Damon Mosk–Aoyama, Ken Wu, Motohiro Yogo, Neil Yorke–Smith, Jiawei Zhang, Dan Zheng, and in particular, Tiffany Lin, to whom I am deeply indebted. I would also like to thank the brothers and sisters from three Christian fellowships that are close to my heart — Young Adults Fellowship at the Peninsula Bible Church, Aletheia Graduate Christian Fellowship (former Chinese Christian Fellowship at Stanford), and Elijah Christian Cell Group — for always praying for me and challenging me to be a better person.

Finally, I would like to express my deepest gratitude to my family, for their love and support over the years, and for instilling in me a passion for learning. This thesis is dedicated to them all.

Contents

List of Figures

Chapter 1

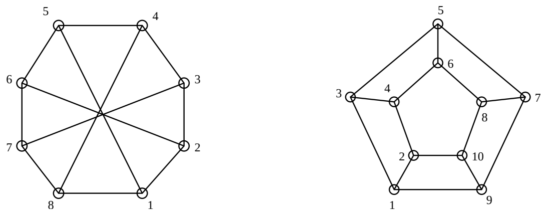
Introduction

Due to its fundamental nature and versatile modelling power, the *Graph Realization Problem* is one of the most well–studied problems in distance geometry and has received attention in many communities. In that problem, one is given a graph $G = (V, E)$ and a set of non–negative edge weights $\{d_{ij} : (i, j) \in E\}$, and the goal is to compute a *realization* of G in the Euclidean space $\mathbb{R}^k$ for a given dimension $k \geq 1$, i.e. to place the vertices of G in $\mathbb{R}^k$ such that the Euclidean distance between every pair of adjacent vertices v_i, v_j is equal to the prescribed weight d_{ij}. The Graph Realization Problem and its variants arise from applications in very diverse areas, the two most prominent of which being molecular conformation (see, e.g., Havel and Wüthrich (1985); Crippen and Havel (1988); Kaptein et al. (1988); Wüthrich (1989); Havel (2003)) and wireless sensor network localization (see, e.g., Doherty et al. (2001); Savvides et al. (2001); Shang et al. (2004); Aspnes et al. (2006); Biswas et al. (2006a)). In molecular conformation, one is interested in determining the spatial structure of a molecule from a set of geometric constraints. The structural information thus obtained can then be used to understand the properties and functions of the molecule. There are usually two distinct types of geometric constraints: (i) the *distance constraints*, which are bounds on the distances between certain pairs of atoms in the molecule; and (ii) the so–called *chirality constraints*, which are bounds on the oriented volumes spanned by selected sets of four atoms in the molecule. These constraints can be obtained both from the primary structure of the molecule and from Nuclear Magnetic Resonance (NMR) measurements (Havel and Wüthrich (1985); Havel (2003)). Observe that one can easily formulate a simplified version of the conformation problem as a Graph Realization Problem, where the vertices of G correspond to atoms, the edges correspond to chemical bonds, and the weights correspond to inter–atomic distances.

In wireless sensor network localization, one is interested in inferring the locations of sensor nodes in a sensor network from connectivity–imposed proximity constraints. The location information is crucial to many applications, such as habitat monitoring, target tracking and geographic routing. Clearly, one could solve the localization problem by equipping every sensor node with a Global Positioning System (GPS) device. However, this approach is prohibitively costly, both in terms of hardware and power requirements. Thus, in most practical settings, only a small subset of the sensor nodes are equipped with such a device. Again, the localization problem can be formulated as a Graph Realization Problem, where the vertices of G correspond to sensors, the edges correspond to communication links, and the weights correspond to distances between sensors. Furthermore, we can partition the vertices into two sets — one is the *anchors*, whose exact positions are known (via GPS, for example); and the other is the *sensors*, whose positions are unknown.

From the above discussion, we see that an algorithm that finds a realization of the vertices in the required dimension will have interesting biochemical and engineering consequences. Unfortunately, unless P=NP, there is no efficient algorithm for solving the Graph Realization Problem for any fixed $k \geq 1$ (Saxe (1979)). Nevertheless, many heuristics have been developed for the problem over the years, and various approaches have been taken to improve their efficiency. For instance, one can exploit additional knowledge from the application area (see, e.g., Crippen and Havel (1988); Hendrickson (1990, 1995); Havel (2003)), or consider relaxations of the original problem (see, e.g., Alfakih et al. (1999); Laurent (2000); Doherty et al. (2001); Aspnes et al. (2006)). However, these approaches have their limitations. Specifically, either they solve the original problem only for a very restricted family of instances, or it is not clear when the algorithm would solve the original problem. Thus, an interesting question arises, which partly motivates our work in this thesis: given a relaxation of the Graph Realization Problem, can one derive reasonably general conditions under which the relaxation is *exact*?

Our work begins by examining a semidefinite programming (SDP) relaxation proposed by Biswas and Ye (2004), who reported the model's superb experimental performance. It turns out that the SDP model captures a lot of geometric information of the problem, and that information can be extracted by SDP duality theory and properties of path–following interior–point algorithms for SDP. Specifically, in Chapter 3, we introduce the notion of *unique k–realizability* and show that the SDP relaxation is exact if and only if the input instance is uniquely k–realizable, where k is the given dimension. One of the attractive features of the notion of unique k–realizability is that it has a straightforward geometric interpretation. Informally, an instance is uniquely k–realizable if (i) it has a unique realization in $\mathbb{R}^k$, and (ii) it does not have any non–trivial realization in $\mathbb{R}^l$, where $l > k$. The above definition also suggests that the notion of unique k–realizability is closely related to the notion of rigidity in the study of frameworks (Graver et al. (1993)). However, the latter focuses mainly on the combinatorial aspect of the Graph Realization Problem, while the former takes into account both the combinatorial and geometric aspects. As a result, the notion of unique k–realizability is more suitable for the algorithmic treatment of the Graph Realization Problem. In Chapter 3 we also introduce the notion of *strong k–realizability*. Roughly speaking, an instance is strongly k–realizable if it is uniquely k–realizable and remains so under slight perturbations. We show that the SDP model will identify all strongly k–realizable sub–instances of a given instance. Thus, even though a given instance may not be uniquely k–realizable, the SDP model can identify portions of it that are uniquely k–realizable. Such a property can be useful in applications. For example, it can be used to detect portions of a network that do not have enough coverage (and hence are not uniquely k–realizable).



(a) V_8 (b) $C_5 \times C_2$

Figure 1.1: The Graphs V_8 and $C_5 \times C_2$

Although we have formulated the Graph Realization Problem as a feasibility problem, it is clear that one can also formulate various optimization versions of it. One particularly useful objective is to maximize the sum of the distances between certain pairs of non–adjacent vertices. Such an objective essentially stretches apart pairs of non–adjacent vertices, and is more likely to flatten a high–dimensional realization into a lower dimensional one. Indeed, such a device has been proven to be very useful for finding low–dimensional realizations in practice (see, e.g., Biswas et al. (2007)). Curiously, such a device can also be used to obtain a purely combinatorial characterization of the class of instances that have low–dimensional realizations. Specifically, in Belk and Connelly (2007), the authors introduced the notion of *k–realizability* of graphs and, among other things, gave a combinatorial characterization of the class of 3–realizable graphs. Informally, a graph G is k–realizable if, given any set of edge lengths, whenever G is realizable at all, then it can also be realized in $\mathbb{R}^k$. The main bottleneck in the characterization of 3–realizable graphs is to show that two graphs, V_8 and $C_5 \times C_2$, are 3–realizable (see Figure 1.1). In order to achieve that, Belk and Connelly first argued that there exists a realization $\mathbf{p}$ of $H \in \{V_8, C_5 \times C_2\}$ in some finite–dimensional Euclidean space such that the distance between a certain pair of non–adjacent vertices (i,j) is maximized. Then, using the Inverse Function Theorem, they showed that the realization $\mathbf{p}$ induces a *non–zero equilibrium force* on the graph H' obtained from H by adding the edge (i,j). Belk (2007) then showed how to use this equilibrium force to "fold" H into $\mathbb{R}^3$.

It turns out that the obstacle to obtaining an algorithm from the above arguments lies in the computation of $\mathbf{p}$ and the associated non–zero equilibrium force. Unfortunately, the proof by Belk and Connelly is not constructive, as it uses the Inverse Function Theorem. However, as we shall see in Chapter 4, the problem of computing the desired realization $\mathbf{p}$ can be formulated as an SDP. What is even more interesting is that the optimal dual multipliers of our SDP give rise to a non–zero equilibrium force. Thus, we are able to give a constructive proof of the Belk–Connelly result. In fact, our proof provides more information than the original proof, and it demonstrates the power of SDP duality theory in tackling distance geometry problems. We then show how to combine our result with the techniques developed in Belk (2007) to obtain an algorithm for realizing 3–realizable graphs.

Finally, in Chapter 5, we take a step back and consider the Graph Realization Problem in a broader context, namely that of rank–constrained semidefinite programming. Indeed, our results in Chapters 3 and 4 can be viewed as providing conditions that guarantee the feasibility and efficient solvability of certain system of linear matrix equations with rank constraints. However, for a general system of linear matrix equations, a low–rank solution may not exist, and even if it does, one may not be able to find it efficiently. Thus, we focus on the problem of finding a low–rank *approximate* solution instead. In the context of the Graph Realization Problem, this corresponds to finding a low–distortion embedding of the given distances into a low–dimensional Euclidean space, which is of course a well–studied problem in the theory of metric embeddings (see, e.g., Matoušek (1990, 2002); Dasgupta and Gupta (1999)). However, to the best of our knowledge, there are no approximation results for a general system of linear matrix equations. In Chapter 5, we show how a simple randomized polynomial–time procedure can be used to produce a low–rank solution that has provably good approximation qualities. Our result provides a unified treatment of and generalizes several well–known results in the literature. In particular, it contains as special cases the Johnson–Lindenstrauss lemma on dimensionality reduction, results on low–distortion embeddings into low–dimensional Euclidean space, and approximation results on certain quadratic optimization problems.

Chapter 2

Preliminaries

In this chapter we collect the main technical tools that will be used throughout this thesis. Readers who are familiar with these materials are encouraged to skip ahead to Chapter 3.

2.1 Notions of Complexity

As we shall see, the algorithms developed in this thesis involve both combinatorial manipulations and numerical computations. In order to measure the efficiency of such algorithms, we need to first specify the model of computation. In this section, we will recall the relevant definitions from complexity theory. The materials in this section are standard; see, e.g., Aho et al. (1974); Grötschel et al. (1993); Wright (1997). Our focus will be on the meaning of polynomial time computations in various models. For the definitions of various complexity classes, we refer the reader to Garey and Johnson (1979).

2.1.1 Problems and Their Encodings

By a *problem* we mean a general question to be answered, which possesses several parameters (or variables), whose values are left unspecified. A problem is described by giving a description of all its parameters and specifying what properties a *solution* is required to satisfy. An *instance* of a problem is obtained by specifying particular values for all the parameters.

More precisely, we assume that we have an *encoding scheme* which represents each instance of the problem as well as each solution as a string of 0's and 1's. In other words, a problem is a subset Π of $\{0,1\}^* \times \{0,1\}^*$, where $\{0,1\}^*$ denotes the set of all finite strings of 0's and 1's. Each string $\sigma \in \{0,1\}^*$ is called an *instance* or *input* of Π, and an $\tau \in \{0,1\}^*$ with $(\sigma, \tau) \in \Pi$ is called a corresponding *solution* or *output* of Π. The *encoding length* (or *input size*, or simply *size*) of an instance I is defined to be the number of symbols in the description of I obtained from the encoding scheme for Π.

It is conceivable that different encoding schemes may give rise to different encoding lengths. However, most of the standard encoding schemes are equivalent, in the sense that the resulting encoding lengths differ at most polynomially from one another. Thus, we shall not specify which particular encoding scheme is used, except that we insist that numbers occurring in an instance are encoded in binary (or in any fixed base other than 1). In particular, the encoding length (or size) of a rational number p/q with $p, q \in \mathbb{Z}$ co–prime and $q \geq 1$ is $1 + \lceil \log_2(|p|+1) \rceil + \lceil \log_2 q \rceil$ and is denoted by $\langle p/q \rangle$. If v is a rational vector or matrix, then its encoding length $\langle v \rangle$ is defined as the sum of the encoding lengths of its entries. If $v_1, \ldots, v_n$ is a sequence of vectors and matrices, then we write $\langle v_1, \ldots, v_n \rangle$ to denote the sum of the encoding lengths of $v_1, \ldots, v_n$.

2.1.2 Models of Computation

An algorithm is often identified with some computational model. Let us recall three popular models of computation in the literature, which we shall use in the sequel. An important distinction among these models is on how the running time of an arithmetic operation is defined.

The first is the *Turing machine* model, which works with binary strings representing abstract objects (e.g. a graph) or rational numbers. The running time of an elementary arithmetic operation (i.e. $+, -, \times, \div$ or comparison) between two rational numbers will depend on their encoding lengths.

The second is the *random access machine (RAM)* model, which operates on *entries* of binary strings stored in a memory. The elementary arithmetic operations listed above are assumed to take unit time. The *encoding length* of an instance of a problem in the RAM model is defined as the *number* of numbers occuring in the input. In particular, we do not count the encoding length of the input numbers. If the input has some non–numeric part (e.g. a graph), then we assume that it is encoded in binary, and each entry of this sequence is considered as a number. Consequently, each non–numeric step in the computation is considered as an arithmetic operation.

The third is the *real random access machine (real RAM)* model, which is similar to the

RAM model but works with real numbers and assumes that exact real arithmetic can be performed. In particular, the memory in a real RAM stores arbitrary real numbers, and the real arithmetic operations $+, -, \times, \div, \sqrt{\ }$ and comparison are assumed to take unit time. We remark that it does not seem reasonable to include the floor function $x \mapsto \lfloor x \rfloor$ in the real RAM model, as it allows any problem in PSPACE or #P to be solved in polynomial time (i.e. the number of arithmetic operations is bounded by a polynomial in the number of input numbers); see Schönhage (1979); Bertoni et al. (1985). One of the motivations for using the real RAM model is that it allows one to ignore issues related to numerical accuracies. In particular, it often simplifies the analysis of geometric algorithms (see, e.g., Boissonnat and Yvinec (1998)).

2.1.3 Polynomial and Strongly Polynomial Computations

For the Turing machine model, we may define the time and space complexity of an algorithm with respect to a given encoding scheme as follows. We say that an algorithm runs in *polynomial time* (resp. *space*) in the Turing machine model if the number of steps it takes to reach the end state from the beginning state (resp. if the maximum length of strings occurring during the execution of the algorithm) is bounded by a polynomial in the encoding length of the input.

In a similar fashion, we say that an algorithm runs in polynomial time in the RAM model if the number of elementary arithmetic operations performed by the algorithm is bounded by a polynomial in the number of input numbers. However, a natural question arises whether this definition has any connection with that in the Turing machine model. To answer this question, we first observe that the elementary arithmetic operations listed above can be executed on a Turing machine in polynomial time. Thus, if we can ensure that the sizes of the numbers occurring during the execution of the algorithm are bounded by a polynomial in the encoding length of the input, then it would imply that a polynomial number of arithmetic operations can be executed in polynomial time on a Turing machine.

The above observations lead us to the definition of a strongly polynomial time algorithm. We say that an algorithm runs in *strongly polynomial time* if (i) the sizes of the numbers occurring during the execution of the algorithm are bounded by a polynomial in the encoding length of the input, and (ii) the number of elementary arithmetic operations performed by the algorithm is bounded by a polynomial in the number of input numbers. In particular, a strongly polynomial time algorithm is a polynomial space algorithm in the Turing machine model and a polynomial time algorithm in the RAM model. We remark that the notion of a strongly polynomial time algorithm is of relevance only for problems that have numbers among its input data. Otherwise, the notions of strongly polynomial time and polynomial time coincide.

2.2 Linear Algebra and Its Algorithmic Aspects

As matrices (in particular, positive semidefinite matrices) play an important role in our investigations, let us first review the relevant definitions and properties. Then, we shall describe some fundamental matrix manipulation algorithms and summarize what is known about their time complexities. In the sequel, all matrices are assumed to have real entries. The materials in this section are again standard; see, e.g., Horn and Johnson (1985); Grötschel et al. (1993).

2.2.1 Eigenvalues, Eigenvectors and Symmetric Matrices

Let $A = [a_{ij}]_{1 \le i \le j \le n}$ be an $n \times n$ matrix. We say that $\lambda \in \mathbb{C}$ is an *eigenvalue* of A if there exists a non–zero vector $u \in \mathbb{C}^n$ such that $Au = \lambda u$. The vector u is then called an *eigenvector* of A associated with λ. The set of eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$ of an $n \times n$ matrix A is closely related to the trace and determinant of A. Indeed, we have the following relations:

$$\mathrm{tr}(A) = \sum_{i=1}^{n} \lambda_i, \quad \det(A) = \prod_{i=1}^{n} \lambda_i \tag{2.1}$$

In addition, the eigenvalues and eigenvectors of a *symmetric* A (i.e. $a_{ij} = a_{ji}$ for all $1 \le i \le j \le n$) are closely related to the structure of A. Specifically, we have the following theorem:

Theorem 2.2.1 (Spectral Theorem for Symmetric Matrices) *Let A be an $n \times n$ symmetric matrix. Then, the following hold:*

(1) The eigenvalues $\lambda_1, \ldots, \lambda_n$ of A are real.

(2) A has n orthonormal eigenvectors $v_1, \ldots, v_n$, where v_i is associated with λ_i for $1 \le i \le n$. Moreover, we have the decomposition $A = V\Lambda V^T$, where $V = [v_1, \ldots, v_n]$ and $\Lambda = diag(\lambda_1, \ldots, \lambda_n)$.

(3) The rank of A is equal to the number of non–zero eigenvalues of A.

Proof. See Theorem 4.1.5 of Horn and Johnson (1985) for (1) and (2). To establish (3), we first observe that $\text{rank}(A) = \text{rank}(U^T AU)$ for any orthogonal matrix U (i.e. U satisfies $U^T U = I$). The proof is then completed upon observing that $V^T AV = \Lambda$. ■

2.2.2 Positive (Semi)–Definite Matrices and Their Properties

An $n \times n$ matrix A is said to be *positive semidefinite (psd)* (resp. *positive definite (pd)*) if A is symmetric and if $x^T Ax \geq 0$ for all $x \in \mathbb{R}^n$ (resp. $x^T Ax > 0$ for all $x \in \mathbb{R}^n \backslash \{0\}$). We emphasize that symmetry is part of our definitions of psd and pd matrices. We write $A \succeq \mathbf{0}$ (resp. $A \succ \mathbf{0}$) as a shorthand for A is positive semidefinite (resp. positive definite).

There are many equivalent characterizations of psd matrices. Below we summarize those that will be used frequently in the sequel.

Theorem 2.2.2 *Let A be an $n \times n$ symmetric matrix. Then, the following are equivalent:*

(1) A is positive semidefinite.

(2) All the eigenvalues of A are non–negative.

(3) All the principal minors of A are non–negative.

(4) There exists a unique $n \times n$ positive semidefinite matrix $A^{1/2}$ such that $A = A^{1/2}A^{1/2}$.

(5) There exists an $k \times n$ matrix B, where $k = \text{rank}(A)$, such that $A = B^T B$.

Proof. See Horn and Johnson (1985), Theorem 7.2.1 for (1) $\Leftrightarrow$ (2); Corollary 7.1.5 for (1) $\Leftrightarrow$ (3); Theorem 7.2.6 for (1) $\Leftrightarrow$ (4). The equivalence of (1) and (5) follows from the spectral theorem for symmetric matrices (Theorem 2.2.1 above). ■

Similar characterizations can be obtained for pd matrices. In particular, we have the following theorem:

Theorem 2.2.3 *Let A be an $n \times n$ symmetric matrix. Then, the following are equivalent:*

(1) A is positive definite.

(2) A^{-1} is positive definite.

(3) A is positive semidefinite and non–singular.

(4) All the eigenvalues of A are positive.

(5) All the principal minors of A are positive.

(6) There exists a unique $n \times n$ positive definite matrix $A^{1/2}$ such that $A = A^{1/2}A^{1/2}$.

We note that many decomposition results for psd matrices (such as (4) and (5) of Theorem 2.2.2) follow from the spectral theorem (Theorem 2.2.1) and the fact that the eigenvalues of an psd matrix are non–negative. Below is another very useful decomposition result that can be obtained via such reasoning.

Theorem 2.2.4 *Let A be an $n \times n$ positive semidefinite matrix of rank k. Then, A can be expressed in the form $A = \sum_{i=1}^k w_i w_i^T$ for some non–zero vectors $w_1, \ldots, w_k \in \mathbb{R}^n$.*

Proof. Since A is symmetric, by Theorem 2.2.1, we have $A = V\Lambda V^T$. Moreover, since A is psd, by Theorem 2.2.2, we see that $\Lambda_{ii} = \lambda_i \geq 0$ for all $1 \leq i \leq n$. Hence, we may write:

$$A = \left(V\Lambda^{1/2}\right)\left(\Lambda^{1/2}V^T\right) = \sum_{i=1}^n \left(\lambda_i^{1/2} v_i\right)\left(\lambda_i^{1/2} v_i\right)^T$$

where $\Lambda^{1/2} = \operatorname{diag}\left(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_n}\right)$. Now, since A has rank k, by Theorem 2.2.1, the sum above has only k non–zero terms. This completes the proof. ■

It is useful to have a criterion for determining the positive semidefiniteness of a matrix from a block partitioning of the matrix. The following well–known result provides one such criterion.

Theorem 2.2.5 (Schur Complement) *Let $A = \begin{bmatrix} X & Y \\ Y^T & Z \end{bmatrix}$ be an $n \times n$ symmetric matrix, where X and Z are square. Suppose that $Z \succ \mathbf{0}$. Then, we have $A \succeq \mathbf{0}$ if and only if $X - YZ^{-1}Y^T \succeq \mathbf{0}$.*

Proof. (cf. p. 28 of Boyd et al. (1994)) By Theorem 2.2.1, there exists an orthogonal matrix U such that $U^TZU = \Lambda$, where $\Lambda \succ \mathbf{0}$ is a diagonal matrix. Clearly, we have $A \succeq \mathbf{0}$ if and only if

$$A' \equiv \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & U^T \end{bmatrix} \begin{bmatrix} X & Y \\ Y^T & Z \end{bmatrix} \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & U \end{bmatrix} = \begin{bmatrix} X & YU \\ U^TY^T & \Lambda \end{bmatrix} \succeq \mathbf{0}$$

Now, consider the quadratic form $f : \mathbb{R}^n \to \mathbb{R}$ defined by A', i.e.:

$$f(w) = w^TA'w = w_1^TXw_1 + 2w_1^TYUw_2 + w_2^T\Lambda w_2 \tag{2.2}$$

where $w = (w_1, w_2)$. Note that for each fixed w_1, the function $f(w_1, \cdot)$ is convex in w_2, and it attains its minimum at $w_2^*(w_1) = -\Lambda^{-1}U^TY^Tw_1$. Upon substituting this into (2.2) and using the relation $Z^{-1} = U\Lambda^{-1}U^T$, we see that the non–negativity of the quadratic form defined by A' is equivalent to the non–negativity of the quadratic form defined by $X - YZ^{-1}Y^T$. This completes the proof. ■

Sometimes it is more convenient to use the following decomposition of positive semidefinite matrices, especially for computational purposes:

Theorem 2.2.6 (LDL^T (or Cholesky) Decomposition) *Let A be an $n \times n$ positive semidefinite matrix. Then, we can decompose A as $A = LDL^T$, where D is an $n \times n$ diagonal matrix whose entries are non–negative, and L is an $n \times n$ unit lower–triangular matrix (i.e. $L_{ii} = 1$ for $1 \le i \le n$ and $L_{ij} = 0$ for $1 \le i < j \le n$). Moreover, the rank of A is equal to the number of non–zero elements on the diagonal of D.*

Proof. We prove the statement by induction on n. For $n = 1$, we have $L = [1]$ and $D = A$. Now, consider the $(n+1) \times (n+1)$ matrix A, given in the form:

$$A = \begin{bmatrix} a_{11} & v^T \\ v & A' \end{bmatrix}$$

Suppose first that $a_{11} > 0$. Then, by Theorem 2.2.5, the $n \times n$ matrix $A' - vv^T/a_{11}$ is positive semidefinite. By the inductive hypothesis, we have $A' - vv^T/a_{11} = L'D'(L')^T$ for some unit lower–triangular matrix L' and some diagonal matrix D' whose entries are non–negative. Then, it is straightforward to verify that:

$$A = \begin{bmatrix} 1 & \mathbf{0} \\ v/a_{11} & L' \end{bmatrix} \begin{bmatrix} a_{11} & \mathbf{0} \\ \mathbf{0} & D' \end{bmatrix} \begin{bmatrix} 1 & v^T/a_{11} \\ \mathbf{0} & (L')^T \end{bmatrix} \tag{2.3}$$

is the desired LDL^T decomposition of A. If $a_{11} = 0$, then we must have $v = \mathbf{0}$. Since A' is positive semidefinite, we have $A' = L'D'(L')^T$ by the inductive hypothesis. It follows that:

$$A = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & L' \end{bmatrix} \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & D' \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & (L')^T \end{bmatrix} \tag{2.4}$$

is the desired LDL^T decomposition of A.

Now, observe that $\text{rank}(A) = \text{rank}(D^{1/2}L^T)$. Since $\det(L^T) = \det(L) = 1$, we see that

L^T has rank n. Since D is a diagonal matrix, it follows that the rank of $D^{1/2}L^T$ is equal to the number of non–zero elements on the diagonal of D. This completes the proof. ■

We close this section with some useful facts concerning positive semidefinite matrices.

Theorem 2.2.7 *Let A be an $n \times n$ positive semidefinite matrix. Then, the following hold:*

(1) Let $M = \max_{1 \le i,j \le n} |a_{ij}|$. Then, there exists an $i \in \{1, \dots, n\}$ such that $a_{ii} = M$.

(2) Suppose that $a_{ii} = 0$ for some $1 \le i \le n$. Then, we have $a_{ij} = a_{ji} = 0$ for all $1 \le j \le n$.

(3) Let $x \in \mathbb{R}^n$ be arbitrary. Then, we have $x^T Ax = 0$ if and only if $Ax = \mathbf{0}$.

Proof. To prove (1), suppose to the contrary that $a_{ii} < M$ for all $1 \le i \le n$. Let $M = |a_{jk}|$ for some $1 \le j < k \le n$. Suppose that $a_{jk} = M > 0$. Then, we have $(e_j - e_k)^T A(e_j - e_k) = a_{jj} + a_{kk} - 2a_{jk} < 0$, where e_j is the j–th standard basis vector of $\mathbb{R}^n$. This contradicts the fact that A is positive semidefinite. The case where $a_{jk} = -M < 0$ can be handled similarly.

To prove (2), suppose to the contrary that $a_{ij} \neq 0$ for some $1 \le j \le n$. Then, for any $\alpha \in \mathbb{R}$, we have $(\alpha e_i + e_j)^T A(\alpha e_i + e_j) = \alpha^2 a_{ii} + 2\alpha a_{ij} + a_{jj} = 2\alpha a_{ij} + a_{jj}$. By choosing α appropriately, we can force the above expression to be negative, thus contradicting the positive semidefiniteness of A.

To prove (3), let $y \in \mathbb{R}^n$ be arbitrary, and consider the polynomial $p : \mathbb{R} \to \mathbb{R}$ given by $p(t) = (x + ty)^T A(x + ty)$. Suppose that $x^T Ax = 0$. Then, we have $p(t) \ge 0$ for all $t \in \mathbb{R}$ and $p(0) = 0$, whence $dp/dt|_{t=0} = 0$. It follows that $y^T Ax = 0$. Since y is arbitrary, we conclude that $Ax = \mathbf{0}$. ■

2.2.3 Matrix Manipulation Algorithms

As we saw in the last section, a positive semidefinite matrix A possesses a lot of structural information, and that information can be extracted by various decompositions of A. In this section, we consider the algorithmic aspects of some of the decomposition results covered in Section 2.2.2. In particular, we are interested in the following algorithmic questions:

Problem. *Given an $n \times n$ symmetric matrix A,*

(1) determine the rank of A;

(2) determine whether A is psd;

(3) if A is psd, compute its LDL^T decomposition.

Before we proceed, let us first review a frequently used algorithm in linear algebra, namely the Gaussian elimination method. We will then use it to develop algorithms for the above problems. Our treatment follows that of Grötschel et al. (1993).

Gaussian Elimination

Given an $m \times n$ matrix A, we would like to transform it into an $m \times n$ matrix of the form:

$$\begin{bmatrix} \Lambda & B \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{2.5}$$

where Λ is an $k \times k$ diagonal matrix and k is the rank of A, using only the following operations:

(1) adding a multiple of one row to another row

(2) permuting rows and columns

The Gaussian elimination is a method for achieving this and works as follows. Given the $m \times n$ matrix A, we first find matrices $A_0, A_1, \ldots, A_k$ such that:

$$A_j = \begin{bmatrix} X_j & Y_j \\ \mathbf{0} & Z_j \end{bmatrix} \qquad \text{for } 0 \leq j \leq k \tag{2.6}$$

where X_j is a non–singular upper triangular $j \times j$ matrix (in particular, the diagonal entries of X_j are non–zero). The matrices $A_0, A_1, \ldots, A_k$ are defined inductively as follows. Let $A_0 = A$ (whence X_0 is empty). For $j \geq 0$, the matrix A_{j+1} is obtained from A_j via the following *Forward Step*:

Forward Step. *Given the matrix A_j which has the form (2.6), choose a non–zero element of Z_j, called the* pivot element, *and permute the rows and columns of A_j so that this pivot element is $Z_j(1,1)$. Now, add multiples of the first row of Z_j to the other rows of Z_j in such a way that $Z_j(1,1)$ becomes the only non–zero element in the first column of Z_j. In* *other words, we perform the following updates:*

$$Z_j(p,q) \leftarrow Z_j(p,q) - \frac{Z_j(p,1)}{Z_j(1,1)} Z_j(1,q) \qquad \textit{for } 2 \leq p \leq m-j,\ 1 \leq q \leq n-j$$

We then define A_{j+1} to be the resulting matrix.

It is clear that the matrix produced by each Forward Step has the form (2.6). We apply the Forward Step until Z_j contains no non–zero element. Thus, the last matrix in the sequence, say A_k, has the form:

$$A_k = \begin{bmatrix} X_k & Y_k \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

where X_k is a non–singular upper triangular $k \times k$ matrix. Since the Forward Step is rank–preserving, we see that $\text{rank}(A) = k$.

To get A_k into the form (2.5), we need to go backwards. Specifically, starting from $E_k = A_k$, we find matrices $E_{k-1}, E_{k-2}, \ldots, E_0$ such that:

$$E_j = \begin{bmatrix} X_j & \mathbf{0} & Y_j \\ \mathbf{0} & \Lambda_j & Z_j \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \qquad \text{for } 0 \leq j \leq k \tag{2.7}$$

where X_j is a non–singular upper triangular $j \times j$ matrix and Λ_j is an $(k-j) \times (k-j)$ diagonal matrix. For $j \leq k$, the matrix E_{j-1} is obtained from E_j via the following *Backward Step*:

Backward Step. *Add multiples of the j–th row of E_j to the other rows of E_j so that $X_j(j,j)$ will be the only non–zero entry in the j–th column of E_j. In other words, we perform the following updates:*

$$X_j(p,q) \leftarrow X_j(p,q) - \frac{X_j(p,j)}{X_j(j,j)} X_j(j,q) \qquad \textit{for } 1 \leq p \leq j-1,\ 1 \leq q \leq j$$

$$Y_j(p,q) \leftarrow Y_j(p,q) - \frac{X_j(p,j)}{X_j(j,j)} Y_j(j,q) \qquad \textit{for } 1 \leq p \leq j-1,\ j+1 \leq q \leq n$$

We then define the matrix so obtained to be E_{j-1}.

Again, it is clear that the matrix produced by each Backward Step has the form (2.7). Moreover, the matrix E_0, which is obtained after k applications of the Backward Step, has the form (2.5). This completes the description of the Gaussian elimination method.

From the descriptions of the Forward Step and the Backward Step, it is not hard to see that the total number of elementary arithmetic operations (i.e. $+, -, \times, \div$ and comparison) used by the Gaussian elimination method is bounded by $O(\max\{m, n\}^3)$. Thus, in order to show that the method runs in polynomial time, it suffices to show that the numbers occurred during the execution of the method do not become too large. There is a subtlety here, however, which involves the issue of how the results of arithmetic operations are encoded. It can be shown (see, e.g., Grötschel et al. (1993)) that if each rational number in A is represented in its co–prime form and we bring the result of each arithmetic operation into this form, then the Gaussian elimination method runs in time polynomial in $\langle A \rangle$. However, such an implementation does not run in strongly polynomial time, as bringing a rational number into its co–prime form is not a strongly polynomial time procedure. On the other hand, by carefully choosing a representation scheme for the rationals, Edmonds (1967) was able to show that the Gaussian elimination method runs in strongly polynomial time. We summarize as follows:

Theorem 2.2.8 *(Grötschel et al. (1993)) For any rational $m \times n$ matrix A, the Gaussian elimination method (using co–prime representation of rationals) runs in time polynomial in $\langle A \rangle$. Furthermore, there is a representation scheme for the rationals so that the Gaussian elimination method runs in strongly polynomial time.*

We are now ready to address the algorithmic questions raised in the beginning of this section.

Determining the Rank of A

To determine the rank of A, it suffices to carry out the Forward Step until there is no non–zero element available for pivoting. The number of Forward Steps carried out is then the rank of A. Thus, we have the following:

Corollary 2.2.9 *There is a strongly polynomial time algorithm for determining the rank of a rational matrix A.*

Determining whether A is Positive Semidefinite

We now show how to use the Gaussian elimination method to determine whether a given matrix is psd. The idea, which is implicit in the proof of Theorem 2.2.6 and is outlined in

Grötschel et al. (1993), is to pivot only on the main diagonal. If the rank of the matrix is found and only positive pivots have been carried out, then the matrix is psd.

Specifically, let $A = [a_{ij}]_{1 \le i,j \le n}$ be an $n \times n$ symmetric matrix with rational entries. We first dispense with the easy cases. If $a_{11} < 0$, then A is not psd. If $a_{11} = 0$ but $a_{1i} \neq 0$ for some $2 \le i \le n$, then A is not psd, as the determinant of the principal submatrix $\begin{bmatrix} a_{11} & a_{1i} \\ a_{1i} & a_{ii} \end{bmatrix}$ is negative. If $a_{11} = 0$ and $a_{1i} = 0$ for all $2 \le i \le n$, then we can drop the first row and the first column and restrict our attention on the remaining $(n-1) \times (n-1)$ matrix.

If $a_{11} > 0$, then we consider the matrix $\tilde{A} = [\tilde{a}_{ij}]_{2 \le i,j \le n}$ obtained by pivoting on a_{11}, i.e.:

$$\tilde{a}_{ij} = a_{ij} - \frac{a_{i1}}{a_{11}} a_{1j} \qquad \text{for } 2 \le i,j \le n \tag{2.8}$$

We claim that A is psd if and only if $\tilde{A}$ is psd. This follows from Theorem 2.2.5. Indeed, let

$$A = \begin{bmatrix} a_{11} & v^T \\ v & A' \end{bmatrix}$$

Then, it is easy to verify that $\tilde{A} = A' - vv^T / a_{11}$. Thus, we have reduced the original $n \times n$ problem to an $(n-1) \times (n-1)$ problem. Moreover, similar to the Gaussian elimination method, the sizes of the numbers occurred during the execution of the above algorithm remain bounded by the size of A. Hence, we have the following:

Corollary 2.2.10 *There is a polynomial time algorithm for determining whether a symmetric rational matrix A is positive semidefinite.*

Computing the LDL^T Decomposition of a Positive Semidefinite A

We observe that the proof of Theorem 2.2.6 essentially gives an algorithm for computing the LDL^T decomposition of a positive semidefinite A, and the idea is very similar to that for determining whether a symmetric matrix A is positive semidefinite. Specifically, let $A = [a_{ij}]_{1 \le i,j \le n}$ be an $n \times n$ positive semidefinite matrix with rational entries. If $a_{11} = 0$, then we compute the LDL^T decomposition of the $(n-1) \times (n-1)$ matrix $A' = [a_{ij}]_{2 \le i,j \le n}$ and return the LDL^T decomposition of A via (2.4). If $a_{11} > 0$, then we compute the LDL^T decomposition of the $(n-1) \times (n-1)$ matrix $\tilde{A}$ defined by (2.8) and return the LDL^T decomposition of A via (2.3). As the sizes of the numbers occurred during the execution of the algorithm remain bounded by the size of A, we have the following:

Corollary 2.2.11 *There is a polynomial time algorithm for computing the LDL^T decomposition of a positive semidefinite rational matrix A.*

2.3 Semidefinite Programming (SDP)

In this section, we review some of the basic notions and results in the theory of semidefinite programming. In particular, we recall the development of SDP duality theory and give an overview of some complexity results on SDP. For additional background information we refer the readers to Helmberg (2000); Nemirovski (2005).

2.3.1 Basic Definitions

In semidefinite programming one is interested in optimizing a linear function of a symmetric matrix subject to linear constraints and a crucial additional constraint that the matrix be positive semidefinite. It can be viewed as an extension of the well–known linear programming problem (the vector of variables is replaced by a symmetric matrix, and the non–negativity constraints are replaced by a positive semidefinite constraint), and is a special case of the so–called *conic programming* problems (restricted to the cone of positive semidefinite matrices). Before we proceed, let us set up some notation. Let $\mathcal{S}^n$ be the space of $n \times n$ symmetric matrices. Note that $\mathcal{S}^n$ can be identified with the Euclidean space $\mathbb{R}^{n(n+1)/2}$. As such, there is a natural inner product, called the *Frobenius inner product* and denoted by $\bullet$, defined on $\mathcal{S}^n$. Specifically, for $A, B \in \mathcal{S}^n$, we have:

$$A \bullet B = \mathrm{tr}(A^T B) = \sum_{i,j=1}^{n} a_{ij} b_{ij}$$

We shall equip $\mathcal{S}^n$ with the partial order $\succeq$ (called the *Löwner partial order*), which is defined as follows: for any $A, B \in \mathcal{S}^n$, we have $A \succeq B$ if and only if $A - B$ is positive semidefinite. The set of $n \times n$ positive semidefinite matrices is denoted by $\mathcal{S}^n_+$. Note that $\mathcal{S}^n_+$ is a closed convex pointed cone (a cone is *pointed* if it does not contain any line that passes through the origin). Moreover, the cone $\mathcal{S}^n_+$ is *self–dual*, i.e.:

$$\left(\mathcal{S}^n_+\right)^* \equiv \left\{ Z \in \mathcal{S}^n : X \bullet Z \geq 0 \text{ for all } X \in \mathcal{S}^n_+ \right\} = \mathcal{S}^n_+ \tag{2.9}$$

For a proof, see, e.g., Lemma 1.2.6 of Helmberg (2000). Now, let $C, A_1, \ldots, A_m \in \mathcal{S}^n$ and $b \in \mathbb{R}^m$ be given. A semidefinite programming problem of the form:

$$\text{(PSDP)}: \quad \begin{array}{rl} \inf & C \bullet X \\ \text{subject to} & A_i \bullet X = b_i \quad \text{for } 1 \leq i \leq m \\ & X \in \mathcal{S}^n_+ \end{array}$$

is said to be in *standard form*. Note that there is no loss of generality in assuming that $C, A_1, \ldots, A_m$ are symmetric, since we have $H^T \bullet X = H \bullet X = \left(\frac{1}{2}(H + H^T)\right) \bullet X$ for any $n \times n$ matrix H. Now, in order to bring out the similarity between standard form linear programs (LP) and standard form semidefinite programs, let us introduce the continuous linear operator $\mathcal{A} : \mathcal{S}^n \to \mathbb{R}^m$ given by $\mathcal{A}X = (A_1 \bullet X, \ldots, A_m \bullet X)$. Then, the standard form SDP (PSDP) can be written as:

$$\begin{array}{rl} \inf & C \bullet X \\ \text{subject to} & \mathcal{A}X = b \\ & X \in \mathcal{S}^n_+ \end{array}$$

which is reminiscent of the standard form LP. Moreover, similar to the LP case, we can write the dual of (PSDP) as follows:

$$\text{(DSDP)}: \quad \begin{array}{rl} \sup & b^T y \\ \text{subject to} & \mathcal{A}^T y + Z = C \\ & y \in \mathbb{R}^m,\ Z \in \mathcal{S}^n_+ \end{array}$$

Here, $\mathcal{A}^T$ is the adjoint operator of $\mathcal{A}$ (i.e. the unique continuous linear operator satisfying $(\mathcal{A}X)^T y = X \bullet (\mathcal{A}^T y)$ for all $X \in \mathcal{S}^n$ and $y \in \mathbb{R}^m$) and is given by:

$$\mathcal{A}^T y = \sum_{i=1}^m y_i A_i$$

We shall refer to (PSDP) as the *primal* problem and to (DSDP) as the *dual* problem.

In order to decide about possible infeasibility and unboundedness of the problems (PSDP) and (DSDP), let us consider the following definitions. A matrix $X \in \mathcal{S}^n_+$ is called a *primal ray* if $\mathcal{A}X = \mathbf{0}$ and $C \bullet X < 0$. A vector $y \in \mathbb{R}^m$ is called a *dual ray* if $-\mathcal{A}^T y \succeq \mathbf{0}$ and $b^T y > 0$. We then have the following elementary result:

Proposition 2.3.1 *The existence of a dual ray implies the infeasibility of (PSDP). Similarly, the existence of a primal ray implies the infeasibility of (DSDP).*

Proof. Let $y \in \mathbb{R}^m$ be a dual ray, and suppose to the contrary that there exists a primal feasible X. Then, we have:

$$0 < b^T y = (\mathcal{A}X)^T y = -(X \bullet (-\mathcal{A}^T y)) \leq 0$$

where the last inequality follows from the self–duality of $\mathcal{S}_+^n$ (2.9). This is a contradiction. The proof for the case of a primal ray proceeds in a similar fashion. ■

In the sequel, we say that (PSDP) (resp. (DSDP)) is *strongly infeasible* if there is a dual (resp. primal) ray.

It turns out that a situation called *weak infeasibility* can also occur in semidefinite programming. Specifically, we say that *(PSDP) is weakly infeasible* if its feasible set is empty, and that for every $\epsilon > 0$, there exists an $X \in \mathcal{S}_+^n$ such that:

$$\max_{1 \leq i \leq m} |A_i \bullet X - b_i| \leq \epsilon$$

Similarly, we say that *(DSDP) is weakly infeasible* if its feasible set is empty, and that for every $\epsilon > 0$, there exists an $(y, Z) \in \mathbb{R}^m \times \mathcal{S}_+^n$ such that:

$$\left\| \sum_{i=1}^m y_i A_i + Z - C \right\| \leq \epsilon$$

Before we prove that (PSDP) and (DSDP) are indeed dual of each other, let us verify the following straightforward result:

Proposition 2.3.2 (SDP Weak Duality) *Let X and (y, Z) be feasible for (PSDP) and (DSDP), respectively. Then, we have $C \bullet X - b^T y = X \bullet Z \geq 0$.*

Proof. We have:

$$C \bullet X - b^T y = (\mathcal{A}^T y + Z) \bullet X - b^T y = (\mathcal{A}X)^T y + Z \bullet X - b^T y = X \bullet Z \geq 0$$

as desired. ■

2.3.2 Theorems of the Alternative and Duality Theory

In the development of the strong duality theory for LPs, the Farkas lemma plays an important role. Given the similarities between LPs and SDPs, it thus seems natural to try generalizing the Farkas lemma to the positive semidefinite cone $\mathcal{S}_+^n$ and use it to develop a strong duality theory for SDPs. It turns out that such a generalization is possible, but only at a cost of additional qualifications. Those additional qualifications are needed in order to ensure that a certain affine transformation of the closed cone $\mathcal{S}_+^n$ remains closed, so that a strict separating hyperplane theorem can be invoked to prove the desired theorem of the alternative. On the other hand, such qualifications are not needed for polyhedral cones (such as the non–negative orthant $\mathbb{R}_+^n$), as the affine image of a closed polyhedral cone is always closed. Consequently, the strong duality theorem for SDPs will be slightly weaker than that for LPs.

Before we state our theorem of the alternative, let us recall the following strict separating hyperplane theorem:

Theorem 2.3.3 *Let $C_1, C_2 \subset \mathbb{R}^n$ be two non–empty closed convex sets with $C_1 \cap C_2 = \emptyset$. If C_2 is bounded, then there exists an $s \in \mathbb{R}^n$ such that:*

$$\sup_{y \in C_1} s^T y < \min_{y \in C_2} s^T y$$

Proof. See Corollary 4.1.3 of Hiriart-Urruty and Lemaréchal (2001). ■

Theorem 2.3.4 (Theorem of the Alternative) *Let $A_1, \ldots, A_m \in \mathcal{S}^n$ and $b \in \mathbb{R}^m$. Suppose that there exists an $\tilde{y} \in \mathbb{R}^m$ such that $\mathcal{A}^T \tilde{y} \succ \mathbf{0}$. Then, either the system (i) $\mathcal{A}X = b$, $X \succeq \mathbf{0}$ has a solution, or the system (ii) $\mathcal{A}^T y \succeq \mathbf{0}$, $b^T y = -1$, $y \in \mathbb{R}^m$ has a solution, but not both.*

Proof. We first show that systems (i) and (ii) cannot hold simultaneously. Suppose to the contrary that there exists a pair $(X, y) \in \mathcal{S}_+^n \times \mathbb{R}^m$ such that X solves (i) and y solves (ii). Then, we have:

$$0 \le (\mathcal{A}^T y) \bullet X = (\mathcal{A}X)^T y = b^T y = -1$$

which is a contradiction.

Now, suppose that system (i) has no solution. Consider the convex cone $C = \{\mathcal{A}X : X \succeq \mathbf{0}\}$. By our hypothesis, we have $b \notin C$. We claim that C is a non–empty closed convex set. Indeed, we have $\mathbf{0} \in C$, and the convexity of C follows from that of $\mathcal{S}_+^n$. To show that C is closed, suppose that we have a sequence $b^i = \mathcal{A}X^i \in C$ for some $X^i \succeq \mathbf{0}$ such that $b^i \to \bar{b}$. We need to show that $\bar{b} \in C$. First, observe that the sequence $\{b^i\}$ is bounded, which in turn implies that $\{\tilde{y}^T b^i\}$ is bounded, where $\tilde{y} \in \mathbb{R}^m$ is such that $\mathcal{A}^T\tilde{y} \succ \mathbf{0}$. We now claim that the sequence $\{X^i\}$ is bounded. Indeed, observe that if $X^i \neq \mathbf{0}$, then we have:

$$\tilde{y}^T b^i = \tilde{y}^T \mathcal{A}X^i = (\mathcal{A}^T\tilde{y}) \bullet X^i = \sqrt{X^i \bullet X^i} \cdot \left((\mathcal{A}^T\tilde{y}) \bullet \frac{X^i}{\sqrt{X^i \bullet X^i}} \right) \geq \delta \cdot \sqrt{X^i \bullet X^i}$$

where $\delta = \inf\{(\mathcal{A}^T\tilde{y}) \bullet X : X \succeq \mathbf{0},\ X \bullet X = 1\} > 0$ by the compactness of the feasible region and the fact that $(\mathcal{A}^T\tilde{y}) \bullet X = \operatorname{tr}(X^{1/2}(\mathcal{A}^T\tilde{y})X^{1/2}) > 0$ for all $X \succeq \mathbf{0}$, $X \neq \mathbf{0}$ (because then we have $X^{1/2}(\mathcal{A}^T\tilde{y})X^{1/2} \succ \mathbf{0}$). Since the leftmost quantity is bounded, it follows that the sequence $\{X^i\}$ is bounded. In particular, by the Bolzano–Weierstrass theorem, the sequence $\{X^i\}$ has a convergent subsequence whose limit we shall denote by $\bar{X}$. Note that $\bar{X} \in \mathcal{S}_+^n$, since $\mathcal{S}_+^n$ is closed. It follows that $\bar{b} = \mathcal{A}\bar{X} \in C$, as desired.

In order to complete the proof, it remains to apply the strict separating hyperplane theorem (Theorem 2.3.3). Since $b \notin C$, there exists an $s \in \mathbb{R}^m$ such that $b^T s < \inf\{z^T s : z \in C\}$. Since $\mathbf{0} \in C$, we see that $b^T s = \alpha < 0$. Now, for any $X \in \mathcal{S}_+^n$, we have $(\mathcal{A}^T s) \bullet X = s^T \mathcal{A}X$. We claim that $s^T\mathcal{A}X \geq 0$ for all $X \in \mathcal{S}_+^n$. Suppose that this is not the case. Then, there exists an $X \in \mathcal{S}_+^n$ such that $0 > s^T\mathcal{A}X \geq \inf\{z^T s : z \in C\} > b^T s$, where the second inequality follows from the fact that $\mathcal{A}X \in C$. However, since $\mathcal{S}_+^n$ is a cone, we have $\gamma X \in \mathcal{S}_+^n$ for all $\gamma > 0$. This implies that $0 > \gamma s^T \mathcal{A}X > b^T s$ for all $\gamma > 0$, which is impossible. Thus, we have $s^T\mathcal{A}X \geq 0$ for all $X \in \mathcal{S}_+^n$, which, together with the self–duality of $\mathcal{S}_+^n$, implies that $\mathcal{A}^T s \succeq \mathbf{0}$. Now, set $y = -s/\alpha$. Then, we have $b^T y = -1$. Moreover, since $-\alpha > 0$, we have $\mathcal{A}^T y \succeq \mathbf{0}$. This completes the proof. ■

We remark that other theorems of the alternative can be derived from the one above, just like in the case of the Farkas lemma. Below we present one such variant. We refer the readers to, e.g., Alizadeh (1995); Lovász (2000) for some other possibilities.

Corollary 2.3.5 *Let $A_1, \ldots, A_m \in \mathcal{S}^n$ be linearly independent, and let $B \in \mathcal{S}^n$. Suppose that there exists an $\tilde{X} \succ \mathbf{0}$ such that $\mathcal{A}\tilde{X} = \mathbf{0}$. Then, either the system (i) $B \bullet X > 0$, $\mathcal{A}X = \mathbf{0}$, $X \succeq \mathbf{0}$ has a solution, or the system (ii) $\mathcal{A}^T y \succeq B$, $y \in \mathbb{R}^m$ has a solution, but not both.*

Proof. Suppose that $(X, y) \in \mathcal{S}_+^n \times \mathbb{R}^m$ is such that X solves (i) and y solves (ii). Then, we have:

$$0 = y^T \mathcal{A}X = (\mathcal{A}^T y) \bullet X \geq B \bullet X > 0$$

which is a contradiction.

Now, let us define the linear operator $\mathcal{H} : \mathcal{S}^n \to \mathbb{R}^{m+1}$ by $\mathcal{H}X = (\mathcal{A}X, B \bullet X)$. We claim that for any $S \succ \mathbf{0}$, there exists an $\tilde{y} \in \mathbb{R}^m$ such that $\mathcal{H}^T(\tilde{y}, -1) = \mathcal{A}^T\tilde{y} - B = S$. To see this, we first vectorize the equation $\mathcal{A}^T\tilde{y} - B = S$ as follows:

$$\mathrm{Mat}(\mathcal{A})\tilde{y} - \mathrm{vec}(B) = \mathrm{vec}(S)$$

Here, $\mathrm{vec}(B) \in \mathbb{R}^{n^2}$ is the column vector formed from the matrix $B \in \mathcal{S}^n$ by stacking its columns on top of each other, and $\mathrm{Mat}(\mathcal{A})$ is the $n^2 \times m$ matrix whose i–th column is $\mathrm{vec}(A_i)$, for $1 \leq i \leq m$. By our hypothesis, the columns of $\mathrm{Mat}(\mathcal{A})$ are linearly independent. It follows that the $m \times m$ matrix $(\mathrm{Mat}(\mathcal{A}))^T\mathrm{Mat}(\mathcal{A})$ is non–singular, and we have:

$$\tilde{y} = \left[(\mathrm{Mat}(\mathcal{A}))^T\mathrm{Mat}(\mathcal{A})\right]^{-1}(\mathrm{Mat}(\mathcal{A}))^T(\mathrm{vec}(S) + \mathrm{vec}(B))$$

as desired.

Lastly, observe that system (i) is equivalent to the system $\mathcal{H}X = (\mathbf{0}, 1)$, $X \succeq \mathbf{0}$. By Theorem 2.3.4, its alternative system is given by $\mathcal{H}^T(y, y_{m+1}) \succeq \mathbf{0}$, $(\mathbf{0}, 1)^T(y, y_{m+1}) = -1$, $y \in \mathbb{R}^m$, $y_{m+1} \in \mathbb{R}$, which is equivalent to $\mathcal{A}^T y - B \succeq \mathbf{0}$, $y \in \mathbb{R}^m$. This completes the proof. ■

We are now ready to prove the strong duality theorem for SDPs (cf. Theorem 2.2.5 of Helmberg (2000), Theorem 1.7.1 of Nemirovski (2005), Lemma 3 of Alizadeh et al. (1997)):

Theorem 2.3.6 (SDP Strong Duality) *Consider the following pair of SDPs:*

$$\begin{aligned} &\textit{(PSDP):} \quad v_p^* = \inf\{C \bullet X : \mathcal{A}X = b,\ X \in \mathcal{S}_+^n\} \\ &\textit{(DSDP):} \quad v_d^* = \sup\{b^T y : \mathcal{A}^T y + Z = C,\ y \in \mathbb{R}^m,\ Z \in \mathcal{S}_+^n\} \end{aligned}$$

Then, the following hold:

(1) If (DSDP) is strictly feasible (i.e. there exists an $(\tilde{y}, \tilde{Z}) \in \mathbb{R}^m \times \mathcal{S}_+^n$ such that $\mathcal{A}^T\tilde{y} + \tilde{Z} = C$, $\tilde{y} \in \mathbb{R}^m$ and $\tilde{Z} \succ \mathbf{0}$), then we have $v_p^* = v_d^*$. If in addition (DSDP) is bounded above, then the common optimal value is attained by some $X^* \in \{X \in \mathcal{S}_+^n : \mathcal{A}X = b\}$.

(2) If (PSDP) is strictly feasible (i.e. there exists an $\tilde{X} \succ \mathbf{0}$ such that $\mathcal{A}\tilde{X} = b$), then

we have $v_p^* = v_d^*$. If in addition (PSDP) is bounded below, then the common optimal value is attained by some $(y^*, Z^*) \in \{(y, Z) \in \mathbb{R}^m \times \mathcal{S}_+^n : \mathcal{A}^T y + Z = C\}$.

(3) Suppose that at least one of (PSDP) or (DSDP) is bounded and strictly feasible. Then, a primal–dual feasible pair $(X; y, Z)$ is a pair of optimal solutions to the respective problems if and only if either one of the following holds:

(a) **(Zero Duality Gap)** $C \bullet X = b^T y$

(b) **(Complementary Slackness I)** $X \bullet Z = 0$*, or equivalently,* $XZ = \mathbf{0}$

(c) **(Complementary Slackness II)** *There exists an $n \times n$ orthogonal matrix V (i.e. $V^T V = I$) such that (i) $X = V \Lambda V^T$ and $Z = V \Omega V^T$ for some $n \times n$ diagonal matrices Λ and Ω, and (ii) $\Lambda\Omega = \mathbf{0}$. In particular, we have rank(X) + rank$(Z) \le n$.*

(4) Suppose that both (PSDP) and (DSDP) are strictly feasible. Then, we have $v_p^* = v_d^*$. The common optimal value is attained by some primal feasible X^* and dual feasible (y^*, Z^*). Moreover, each of the conditions 3(a) and 3(b) is necessary and sufficient for optimality of a primal–dual feasible pair.

Proof. We begin with (1). By weak duality (Proposition 2.3.2), we have $v_p^* \ge b^T \tilde{y} > -\infty$. Now, suppose that $v_p^* > v_d^*$. Consider the linear operator $\mathcal{H} : \mathcal{S}^n \to \mathbb{R}^{m+1}$ defined by $\mathcal{H}X = (\mathcal{A}X, C \bullet X)$. Then, we have $\mathcal{H}^T(\tilde{y}, -1) = \tilde{Z} \succ \mathbf{0}$. Moreover, the system $\mathcal{H}X = (b, v_d^*)$, $X \succeq \mathbf{0}$ has no solution. Thus, by Theorem 2.3.4, there exists an $(y, y_{m+1}) \in \mathbb{R}^{m+1}$ such that $\mathcal{H}^T(y, y_{m+1}) = \mathcal{A}^T y + y_{m+1} C \succeq \mathbf{0}$ and $b^T y + v_d^* y_{m+1} = -1$. We consider the following cases:

Case 1. $y_{m+1} = 0$. Then, we have $\mathcal{A}^T y \succeq \mathbf{0}$ and $b^T y = -1$. It follows that $y_\alpha = \tilde{y} - \alpha y$ with $\alpha \ge 0$ is a dual feasible ray along which the dual objective value is strictly increasing. Thus, we have $v_d^* = \infty$, which is a contradiction.

Case 2. $y_{m+1} > 0$. Then, we have $\mathcal{A}^T(y/y_{m+1}) + C \succeq \mathbf{0}$ and $b^T(y/y_{m+1}) + v_d^* = -1$. Then, the dual feasible solution $-y/y_{m+1}$ yields an objective value better than v_d^*, which is absurd.

Case 3. $y_{m+1} < 0$. Then, we have $\mathcal{A}^T(-y/y_{m+1}) - C \succeq \mathbf{0}$ and $b^T(-y/y_{m+1}) - v_d^* = 1/y_{m+1}$. We claim that $v_d^* = \infty$. Suppose that this is not the case. Let $y' \in \mathbb{R}^m$ be such that $v_d^* - b^T y' \le -1/2y_{m+1}$ and $C - \mathcal{A}^T y' \succeq \mathbf{0}$. It then follows that:

$$\mathcal{A}^T(-y' - y/y_{m+1}) \succeq \mathbf{0}, \quad b^T(-y' - y/y_{m+1}) \le 1/y_{m+1} < 0$$

However, this implies that $y_\alpha = \tilde{y} + \alpha(y' + y/y_{m+1})$ with $\alpha \geq 0$ is a dual feasible ray along which the dual objective value is strictly increasing. This is a contradiction.

Hence, we conclude that $v_p^* = v_d^*$. To see that the common optimal value is attained by some primal feasible X^*, it suffices to observe that the set $\{\mathcal{H}X : X \succeq \mathbf{0}\}$ is closed; see the proof of Theorem 2.3.4. This completes the proof of (1).

One may establish (2) using the primal–dual symmetry. Alternatively, we can use Corollary 2.3.5. By weak duality (Proposition 2.3.2), we have $v_d^* \leq C \bullet \tilde{X} < \infty$. Now, suppose that $v_p^* > v_d^*$. Without loss of generality, we may assume that the matrices $A_1, \ldots, A_m$ are linearly independent, for otherwise we can eliminate the redundant primal constraints. Now, let

$$B = \begin{bmatrix} -C & \mathbf{0} \\ \mathbf{0} & v_p^* \end{bmatrix}, \qquad A_i' = \begin{bmatrix} A_i & \mathbf{0} \\ \mathbf{0} & -b_i \end{bmatrix} \quad \text{for } 1 \leq i \leq m$$

It is clear that $B, A_1', \ldots, A_m' \in \mathcal{S}^{n+1}$, and that $A_1', \ldots, A_m'$ are linearly independent. Moreover, the $(n+1) \times (n+1)$ matrix:

$$\bar{X} = \begin{bmatrix} \tilde{X} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \succ \mathbf{0}$$

satisfies $\mathcal{A}'\bar{X} = \mathbf{0}$, and that the system:

$$\mathcal{A}'^T y - B = \begin{bmatrix} -\sum_{i=1}^{m}(-y_i)A_i + C & \mathbf{0} \\ \mathbf{0} & b^T(-y) - v_p^* \end{bmatrix} \succeq \mathbf{0}, \quad y \in \mathbb{R}^m$$

has no solution. Hence, by Corollary 2.3.5, there exists an $X' \succeq \mathbf{0}$ of the form:

$$X' = \begin{bmatrix} X & * \\ * & \theta \end{bmatrix}$$

such that $B \bullet X' = -C \bullet X + \theta v_p^* > 0$ and $\mathcal{A}'X' = \mathbf{0}$. We consider two cases:

Case 1. $\theta = 0$. Then, we have $C \bullet X < 0$, and $X_\alpha = \tilde{X} + \alpha X$ with $\alpha \geq 0$ is a primal feasible ray along which the primal objective value is strictly decreasing. Thus, we have $v_p^* = -\infty$, which is a contradiction.

Case 2. $\theta > 0$. Then, we have $C \bullet (X/\theta) < v_p^*$ and $A_i \bullet (X/\theta) - b_i = 0$ for $1 \leq i \leq m$. It follows that the primal feasible solution X/θ yields an objective value better than v_p^*, which is not possible.

Hence, we again have $v_p^* = v_d^*$. Moreover, it can be shown that the set $\{(b^T y, \mathcal{A}^T y + Z) : y \in \mathbb{R}^m, Z \in \mathcal{S}_+^n\}$ is closed. Thus, the common optimal value is attained by some dual feasible (y^*, Z^*).

We now proceed to establish (3). We first show that if $X, Z \succeq \mathbf{0}$, then $X \bullet Z = 0$ implies that $XZ = \mathbf{0}$ (the converse is trivial). Indeed, by Theorem 2.2.2, we have:

$$0 = X \bullet Z = \mathrm{tr}(XZ) = \mathrm{tr}((X^{1/2}Z^{1/2})^T(X^{1/2}Z^{1/2})) \tag{2.10}$$

Since the matrix $(X^{1/2}Z^{1/2})^T(X^{1/2}Z^{1/2})$ is positive semidefinite, its diagonal entries must be non–negative. This, together with (2.10), implies that $(X^{1/2}Z^{1/2})^T(X^{1/2}Z^{1/2}) = \mathbf{0}$, which in turn yields $XZ = \mathbf{0}$ as desired.

Next, we show that the conditions (a), (b) and (c) are all equivalent. Indeed, recall that the duality gap satisfies $C \bullet X - b^T y = X \bullet Z$ (see Proposition 2.3.2). Thus, the duality gap is zero if and only if complementary slackness holds, i.e. (a) and (b) are equivalent. Now, suppose that (c) holds. Then, we have $XZ = (V\Lambda V^T)(V\Omega V^T) = V(\Lambda\Omega)V^T = \mathbf{0}$, i.e. (b) holds. Conversely, if (b) holds, then X and Z commute, and hence they share a common system of orthonormal eigenvectors $v_1, \ldots, v_n$. By Theorem 2.2.1, we can then write $X = V\Lambda V^T$ and $Z = V\Omega V^T$, where $V = [v_1, \ldots, v_n]$, and Λ and Ω are diagonal. Moreover, we have $\mathbf{0} = XZ = V(\Lambda\Omega)V^T$. This implies that $\Lambda\Omega = \mathbf{0}$, and hence (c) holds.

Now, let $(X; y, Z)$ be a primal–dual feasible pair. We express the duality gap as follows:

$$C \bullet X - b^T y = (C \bullet X - v_p^*) + (v_p^* - v_d^*) + (v_d^* - b^T y)$$

Note that all the parenthesized terms on the right–hand side are non–negative, and they all equal to zero if and only if $v_p^* = v_d^*$, and that $(X; y, Z)$ is a primal–dual optimal pair. It is then clear that (3) follows from (1) and (2). We remark that the above arguments hold without any assumptions of strict feasibility. In particular, if a primal–dual feasible pair has zero duality gap, then the pair is automatically primal–dual optimal. On the other hand, the converse is true only if $v_p^* = v_d^*$.

Finally, it is straightforward to see that (4) is a consequence of (3). ■

It is curious that despite the similarities between LPs and SDPs, the strong duality theorem for SDPs stated above is weaker than that for LPs. In particular, recall that in the case of LP, as long as either the primal or the dual problem is feasible and bounded, the primal and dual optimal values are equal, and the common optimal value can be attained both in the primal and in the dual. Thus, it is natural to ask whether the strong duality theorem for SDPs holds without the strict feasibility assumption. Unfortunately, the answer is negative in general. We shall see in Chapter 4 one such example that arises from a geometric context. For additional, more algebraically motivated examples, we refer the readers to Helmberg (2000).

2.3.3 Algorithms and Complexity Issues

One of the most fundamental algorithmic problems in semidefinite programming is the following:

Problem (Semidefinite Feasibility (SDF)). *Given rational matrices $A_0, A_1, \ldots, A_m \in \mathcal{S}^n$, determine if there exists an $y \in \mathbb{R}^m$ such that $A_0 - \sum_{i=1}^m y_i A_i \succeq \mathbf{0}$.*

It is not hard to see that (SDF) belongs to NP $\cap$ co–NP in the real RAM model, since one can test whether a matrix is positive semidefinite in polynomial time using, e.g., Gaussian elimination (see Section 2.2.3). However, it is not known whether (SDF) can be solved in polynomial time in the real RAM model. On the other hand, the exact complexity of (SDF) in the Turing machine model is not known. In particular, it is not known whether (SDF) belongs to NP. Part of the difficulty comes from the fact that there are "ill–conditioned" instances of (SDF) in which all the solutions are either irrational or do not have polynomially bounded sizes. For an illustration of the latter, consider the following example due to Ramana (1997):

$$\mathcal{F} = \left\{ y \in \mathbb{R}^m : (y_1 - 4) \succeq 0, \begin{bmatrix} 1 & y_i \\ y_i & y_{i+1} \end{bmatrix} \succeq \mathbf{0} \text{ for } 1 \le i \le m-1 \right\} \tag{2.11}$$

It is easy to see that $\mathcal{F} = \left\{ y \in \mathbb{R}^m : y_i \ge 2^{2^i} \text{ for } 1 \le i \le m \right\}$. In particular, any $y \in \mathcal{F}$ has $\log \|y\| \ge 2^m$, i.e. the encoding length of any feasible solution is exponential in the size of the input if binary encoding is used. Currently, the best bound on the complexity of semidefinite programming was obtained by Porkolab and Khachiyan (1997), who showed the following:

Theorem 2.3.7 *Suppose that the $n \times n$ symmetric matrices $A_0, A_1, \ldots, A_m$ in (SDF) are* *integral. Let l be the maximum bit length of the entries in $A_0, A_1, \ldots, A_m$. Define:*

$$\mathcal{F} = \left\{ y \in \mathbb{R}^m : A_0 - \sum_{i=1}^{m} y_i A_i \succeq \mathbf{0} \right\} \tag{2.12}$$

If $\mathcal{F} \neq \emptyset$, then there exists an $y \in \mathcal{F}$ such that $\log \|y\| \leq ln^{O(\min\{m,n^2\})}$. Moreover, (SDF) can be solved in $O(mn^4) + n^{O(\min\{m,n^2\})}$ arithmetic operations over $ln^{O(\min\{m,n^2\})}$–bit numbers. In particular, if m is fixed, then (SDF) can be solved in polynomial time in the Turing machine model.

In terms of complexity classes, Ramana (1997) showed that (SDF) is either in NP ∩ co–NP or outside NP ∪ co–NP. This is a consequence of an exact duality theory he developed for semidefinite programming.

On the other hand, there exist polynomial–time algorithms (modulo some technical conditions) that solve (SDF) (or more generally, (DSDP)) *approximately.* In order to make the notion of approximation precise, let $\mathcal{F}$ be as in (2.12), where the entries in $A_0, A_1, \ldots, A_m$ are rational. For any $\epsilon > 0$, set:

$$\begin{aligned} S(\mathcal{F}, \epsilon) &= \{ z \in \mathbb{R}^m : \exists y \in \mathcal{F} \text{ such that } \|z - y\| < \epsilon \} \\ S(\mathcal{F}, -\epsilon) &= \mathbb{R}^m \backslash S(\mathbb{R}^m \backslash \mathcal{F}, \epsilon) \end{aligned}$$

The *weak optimization problem* associated with $\mathcal{F}$ (which we shall denote by (W–DSDP)) is defined as follows: given a rational $b \in \mathbb{R}^m$ and a rational $\epsilon > 0$, either (i) find an $y \in \mathbb{Q}^n$ such that $y \in S(\mathcal{F}, \epsilon)$ and $b^T x \leq b^T y + \epsilon$ for all $x \in S(\mathcal{F}, -\epsilon)$, or (ii) assert that $S(\mathcal{F}, -\epsilon) = \emptyset$. A vector $y \in \mathbb{Q}^n$ that satisfies the conditions in (i) will be called *an ϵ–approximate solution* to (DSDP).

Now, suppose that we know a bound $R > 0$ such that either $\mathcal{F} = \emptyset$, or there exists an $z \in \mathcal{F}$ with $\|z\| \leq R$. Furthermore, let l be the maximum bit length of the entries in $A_1, \ldots, A_m$ and b. Then, the ellipsoid method (where the separation oracle is given by the algorithm for checking the positive semidefiniteness of a matrix) yields an algorithm for solving (W–DSDP) whose running time is polynomial in $n, m, L, \log R$ and $\log(1/\epsilon)$ (Corollary 4.2.7 of Grötschel et al. (1993)). Note that this does not automatically yield a polynomial time algorithm (i.e. one whose running time is polynomial in n, m, L and $\log(1/\epsilon)$) for (W–DSDP), as $\log R$ may not be polynomially bounded (see (2.11)).

Of course, the ellipsoid method is notoriously slow in practice. To remedy this situation, many interior–point algorithms have been developed, and they are efficient both in theory and in practice. One of the classes of algorithms for solving SDPs is the so–called *primal–dual path–following* methods, which, as the name suggests, tackle both the primal and dual problems simultaneously. Specifically, let $\mathcal{F}_p^\circ$ and $\mathcal{F}_d^\circ$ be the set of strictly feasible solutions to (PSDP) and (DSDP), respectively. For simplicity, we shall assume that (i) both $\mathcal{F}_p^\circ$ and $\mathcal{F}_d^\circ$ are non–empty, and (ii) the matrices $A_1, \ldots, A_m$ are linearly independent. Consider the *central path* defined by:

$$\mathcal{C} = \left\{ (X; y, Z) \in \mathcal{F}_p^\circ \times \mathcal{F}_d^\circ : XZ = \mu I,\ \mu > 0 \right\}$$

The following is well–known (see, e.g., Corollary 2.20 of Ye (1997)) and shows that the central path is well–defined:

Theorem 2.3.8 *Suppose that assumptions (i) and (ii) hold. Then, for any $\mu > 0$, there exists a unique $(X(\mu); y(\mu), Z(\mu)) \in \mathcal{F}_p^\circ \times \mathcal{F}_d^\circ$ such that $X(\mu)Z(\mu) = \mu I$. Furthermore, the limit $(X^*; y^*, Z^*) = \lim_{\mu \searrow 0}(X(\mu); y(\mu), Z(\mu))$ exists and X^* (resp. (y^*, Z^*)) is an optimal solution to (PSDP) (resp. (DSDP)).*

Another important property of the central path is that it converges to a pair of so–called *maximally complementary* solutions as $\mu \searrow 0$ (cf. de Klerk et al. (1997)):

Theorem 2.3.9 *Suppose that assumptions (i) and (ii) hold, and let $(X^*; y^*, Z^*)$ be as in Theorem 2.3.8. Then, the rank of X^* is maximal among all optimal solutions to (PSDP). Similarly, the rank of Z^* is maximal among all optimal solutions to (DSDP). The pair (X^*, Z^*) is said to be* maximally complementary.

Proof. Let $\tilde{X}^*$ (resp. $\tilde{Z}^*$) be an optimal solution to (PSDP) (resp. (DSDP)) of maximal rank. For any $\mu > 0$, observe that:

$$\begin{aligned} (\tilde{X}^* - X(\mu)) \bullet (\tilde{Z}^* - Z(\mu)) &= (\tilde{X}^* - X(\mu)) \bullet \left(\mathcal{A}^T(y(\mu) - \tilde{y}^*) \right) \\ &= (y(\mu) - \tilde{y}^*)^T (\mathcal{A}(\tilde{X}^* - X(\mu))) \\ &= 0 \end{aligned}$$

where the last equality follows from the fact that both $\tilde{X}^*$ and $X(\mu)$ are feasible for (PSDP).

By Theorem 2.3.6, we have $\tilde{X}^* \bullet \tilde{Z}^* = 0$. Thus, we obtain:

$$\tilde{Z}^* \bullet X(\mu) + \tilde{X}^* \bullet Z(\mu) = X(\mu) \bullet Z(\mu) = n\mu \tag{2.13}$$

Since $X(\mu), Z(\mu) \succ \mathbf{0}$, we may multiply both sides of (2.13) by $X(\mu)^{-1}Z(\mu)^{-1}$ and obtain:

$$\tilde{Z}^* \bullet Z(\mu)^{-1} + \tilde{X}^* \bullet X(\mu)^{-1} = n \tag{2.14}$$

Now, let $P(\mu) = X(\mu)^{-1/2}\tilde{X}^* X(\mu)^{-1/2}$ and $Q(\mu) = Z(\mu)^{-1/2}\tilde{Z}^* Z(\mu)^{-1/2}$. It is clear that $P(\mu)$ and $Q(\mu)$ are positive semidefinite. Since $\tilde{X}^* \bullet X(\mu)^{-1} = \mathrm{tr}(P(\mu))$ and $\tilde{Z}^* \bullet Z(\mu)^{-1} = \mathrm{tr}(Q(\mu))$, it follows from (2.14) that the matrices $nI - P(\mu)$ and $nI - Q(\mu)$ are positive semidefinite. This in turn implies that:

$$\begin{array}{rcll} nX(\mu) - \tilde{X}^* & = & X(\mu)^{1/2}(nI - P(\mu))X(\mu)^{1/2} & \succeq \mathbf{0} \\ nZ(\mu) - \tilde{Z}^* & = & Z(\mu)^{1/2}(nI - Q(\mu))Z(\mu)^{1/2} & \succeq \mathbf{0} \end{array}$$

or equivalently,

$$X(\mu) \succeq \frac{1}{n}\tilde{X}^* \quad \text{and} \quad Z(\mu) \succeq \frac{1}{n}\tilde{Z}^*$$

Therefore, for any $\mu > 0$, the rank of $X(\mu)$ is bounded below by that of $\tilde{X}^*$, and the rank of $Z(\mu)$ is bounded below by that of $\tilde{Z}^*$. The proof is then completed by taking $\mu \searrow 0$ and observing that $\mathcal{S}_+^n$ is closed. ■

The development of path–following algorithms are motivated by Theorem 2.3.8 and they aim at tracing the central path as the parameter μ is decreased to zero. As an illustration, let us consider the *short step* path–following algorithm (see, e.g., Nesterov and Todd (1998); Monteiro and Zanjácomo (1999)), which works roughly as follows. Suppose that at the beginning of some iteration $k \geq 0$, we are given a point $(X^k; y^k, Z^k) \in \mathcal{F}_p^\circ \times \mathcal{F}_d^\circ$ that lies in a (suitably defined) neighborhood $\mathcal{N}(\mathcal{C})$ of the central path. Let $\mu^k = X^k \bullet Z^k/n$ be the normalized duality gap. The algorithm then uses Newton's method to produce a point $(X^{k+1}; y^{k+1}, Z^{k+1})$ that is close to the central path and has a smaller normalized duality gap. In particular, as demonstrated in Nesterov and Todd (1998), the Newton directions can be chosen so that $(X^{k+1}; y^{k+1}, Z^{k+1}) \in \mathcal{N}(\mathcal{C}) \cap (\mathcal{F}_p^\circ \times \mathcal{F}_d^\circ)$, and that $\mu^{k+1} = X^{k+1} \bullet Z^{k+1}/n = (1 - \kappa/\sqrt{n})\mu^k$ for some constant $\kappa \in (0,1)$. Thus, we can repeat the process until some convergence criteria are met. In particular, the algorithm will produce a strictly feasible pair $(X^k; y^k, Z^k)$ that satisfies $X^k \bullet Z^k \leq \epsilon$ in $O(\sqrt{n} \cdot \log(X^0 \bullet Z^0/\epsilon))$ iterations.

Now, in each iteration, the algorithm needs to compute the Newton directions and determine the next iterate, and these can be implemented using $O(mn^3 + n^2m^2 + m^3)$ arithmetic operations (see Monteiro and Zanjácomo (1999) and the references therein). Hence, we may summarize the overall complexity of the short step method as follows:

Theorem 2.3.10 *Let X^0 and (y^0, Z^0) be strictly feasible for (PSDP) and (DSDP), respectively, and let $\epsilon > 0$ be given. Suppose that $(X^0; y^0, Z^0)$ lies in a sufficiently small neighborhood of $\mathcal{C}$. Then, by using the short step path–following algorithm, one can obtain a strictly feasible pair $(X^k; y^k, Z^k)$ that satisfies $X^k \bullet Z^k \le \epsilon$ in $O(\sqrt{n} \cdot \log(X^0 \bullet Z^0/\epsilon))$ iterations. Each iteration requires $O(mn^3 + n^2m^2 + m^3)$ arithmetic operations.*

We remark that some computational savings are possible if the matrices $A_1, \ldots, A_m$ are sparse. For instance, Benson et al. (2000) developed a dual scaling algorithm that takes advantage of the case where all the matrices $A_1, \ldots, A_m$ are positive semidefinite of rank–1. Their algorithm has the same iteration complexity as the short–step method, but the number of arithmetic operations needed in each iteration is smaller. We refer the readers to Benson et al. (2000) for further details.

Observe that the short–step path–following algorithm as stated above requires a strictly feasible starting pair $(X^0; y^0, Z^0)$. However, such a pair may not be readily available. To circumvent this difficulty, de Klerk et al. (1997, 1998) presented an initialization strategy called *self–dual embedding* where the original problems (PSDP) and (DSDP) are embedded in a larger SDP problem with a known strictly feasible starting point. Specifically, they introduced the following problem:

$$
\text{(SDE)}: \quad
\begin{array}{lll}
\text{minimize} & \theta\beta & \\
\text{subject to} & A_i \bullet X - \tau b_i + \theta \bar{b}_i = 0 & \text{for } 1 \le i \le m \\
 & -\sum_{i=1}^m y_i A_i + \tau C - \theta \bar{C} - Z = 0 & \\
 & b^T y - C \bullet X + \theta\alpha - \rho = 0 & \\
 & -\bar{b}^T y + \bar{C} \bullet X - \tau\alpha - \nu = -\beta & \\
 & X \in \mathcal{S}^n_+, \ y \in \mathbb{R}^m, \ Z \in \mathcal{S}^n_+ & \\
 & \theta \ge 0, \ \rho \ge 0, \ \tau \ge 0, \ \nu \ge 0 &
\end{array}
$$

where:

$$\bar{b}_i = b_i - \mathrm{tr}(A_i),\ \bar{C} = C - I,\ \alpha = 1 + \mathrm{tr}(C),\ \beta = n + 2$$

The above embedding problem has many nice properties. First, it is straightforward to verify that $X^0 = Z^0 = I$, $y^0 = 0$ and $\theta^0 = \rho^0 = \tau^0 = \nu^0 = 1$ constitutes a strictly feasible solution to (SDE). Consequently, we may use any path–following algorithm to solve (SDE). Secondly, the embedding problem is self–dual and hence the duality gap is $2\theta\beta$. Moreover, since (SDE) has a strictly feasible solution, it follows from Theorem 2.3.6 that the duality gap is zero, and $\theta^* = 0$ is an optimal solution. Thirdly, it can be verified that $\theta\beta = X \bullet Z + \rho\tau + \theta\nu$, whence an optimal solution to (SDE) satisfies the following complementarity conditions:

$$XZ = \mathbf{0},\ \rho\tau = \theta\nu = 0 \tag{2.15}$$

Now, by Theorem 2.3.9, there exists a pair of maximally complementary solutions to (SDE). It turns out that such a pair can be used to obtain information about the original problems (PSDP) and (DSDP). Specifically, consider the following possibilities for (PSDP) and (DSDP):

(I) An optimal pair $(X^*; y^*, Z^*)$ for (PSDP) and (DSDP) is obtained with zero duality gap.

(II) A primal and/or dual ray is obtained, i.e. strong infeasibility is detected.

(III) A certificate is obtained that no optimal pair with zero duality gap exists, and that neither (PSDP) nor (DSDP) has a ray.

The following result shows that the above three possibilities can be distinguished once a pair of maximally complementary solutions to (SDE) is given:

Theorem 2.3.11 *(de Klerk et al. (1997)) Let $(X^*, \rho^*, \theta^*; y^*, Z^*, \tau^*, \nu^*)$ be a pair of maximally complementary solutions to (SDE) (where the pairing is given by (2.15)). Then, the following statements are valid:*

(1) if $\tau^* > 0$, then case (I) holds;

(2) if $\tau^* = 0$ and $\rho^* > 0$, then case (II) holds;

(3) if $\tau^* = \rho^* = 0$, then case (III) holds.

Proof. Recall that we have $\theta^* = 0$. Now, suppose that $\tau^* > 0$. Then, by complementarity, we have $\rho^* = 0$. Moreover, it is straightforward to verify that X^*/τ^* and $(y^*/\tau^*, Z^*/\tau^*)$ are maximally complementary and are optimal for (PSDP) and (DSDP), respectively. Thus, case (I) holds.

If $\tau^* = 0$, then we have $\tau = 0$ in any optimal solution to (SDE). This implies that we cannot have a pair of optimal solutions to (PSDP) and (DSDP) with zero duality gap, for the existence of such a pair would imply the existence of an optimal solution to (SDE) with $\tau = 1$. Note that $\tau^* = 0$ implies that $A_i \bullet X = 0$ for all $1 \leq i \leq m$ and $-\sum_{i=1}^m y_i A_i \succeq \mathbf{0}$. Now, suppose in addition that $\rho^* > 0$. Then, we have $b^T y^* - C \bullet X^* > 0$, i.e. we have either $b^T y^* > 0$ or $C \bullet X^* < 0$, or both. In other words, we have a primal and/or dual ray, whence case (II) holds.

Conversely, suppose that there exists a primal and/or dual ray. We need to show that any pair of maximally complementary solutions must have $\tau^* = 0$ and $\rho^* > 0$. We shall consider the case of a primal ray. The proof for the case of a dual ray will proceed similarly. Given a primal ray $\tilde{X}$, we set $X^* = \kappa \tilde{X}$, where $\kappa > 0$ is some constant to be determined. Furthermore, we set $\tau^* = y^* = \theta^* = 0$. Then, the constraints give:

$$\begin{aligned} Z^* &= \mathbf{0} \\ \rho^* &= -\kappa(C \bullet \tilde{X}) > 0 \\ \nu^* &= \kappa(C \bullet \tilde{X} - \operatorname{tr}(\tilde{X})) + n + 2 \end{aligned}$$

It is clear that Z^* and ρ^* are feasible for (SDE). Moreover, we can make ν^* feasible (i.e. $\nu^* \geq 0$) by choosing $\kappa > 0$ to be sufficiently small. Since $\theta^* = 0$, the choice $(X^*, \rho^*, \theta^*; y^*, Z^*, \tau^*, \nu^*)$ is optimal for (SDE).

Finally, if a pair of maximally complementary solutions is such that $\tau^* = \rho^* = 0$, then we have $\tau = \rho = 0$ in every optimal solution to (SDE). In particular, cases (I) and (II) cannot occur. This completes the proof. ■

We remark that case (III) can only occur if one or both of (PSDP) and (DSDP) are not strictly feasible. Specifically, case (III) covers the following situations:

- there exists a pair of optimal solutions to (PSDP) and (DSDP) with a positive duality gap;
- either (PSDP) or (DSDP) (or both) are weakly infeasible;

- the duality gap between the optimal value of (PSDP) and that of (DSDP) is zero, but the common optimal value is not attained in the primal or in the dual, or both.

For further properties of (SDE) we refer the readers to the articles by de Klerk et al. (1997, 1998).

Chapter 3

The Graph Realization Problem

3.1 Problem Statement

Since we are mainly interested in inferring information about a point set in some Euclidean space from the interpoint Euclidean distances, we shall use $\|\cdot\|$ to denote the Euclidean norm (i.e. $\|x\| = \left(\sum_{i=1}^{k} x_i^2\right)^{1/2}$ for $x \in \mathbb{R}^k$). Also, all graphs below are assumed to be *simple*, i.e. there are no loops or multiple edges. The focus of this chapter is the *Graph Realization Problem*, which is defined as follows:

Graph Realization Problem (GRP). *Given a graph $G = (V, E)$ with n vertices and m edges, a vector $\mathbf{d} = (d_{ij}^2)_{(i,j)\in E} \in \mathbb{Q}_+^m$ and an integer $k \geq 1$, decide if there exist vectors $p_1, \ldots, p_n \in \mathbb{R}^k$ such that $d_{ij}^2 = \|p_i - p_j\|^2$ for all $(i,j) \in E$ (the configuration $\mathbf{p} = (p_1, \ldots, p_n) \in \mathbb{R}^{kn}$ is then called a* realization *of $(G, \mathbf{d})$ in $\mathbb{R}^k$).*

Naturally, we are also interested in *finding* an actual realization of a Yes instance of (GRP). However, as one readily observes, a Yes instance of (GRP) may admit more than one non–congruent realizations (we say that two realizations $\mathbf{p} = (p_1, \ldots, p_n)$ and $\mathbf{q} = (q_1, \ldots, q_n)$ are *congruent* to each other (and denoted by $\mathbf{p} \equiv \mathbf{q}$) if $\|p_i - p_j\| = \|q_i - q_j\|$ for all $1 \leq i < j \leq n$). For instance, Figure 3.1 shows two non–congruent realizations of a unit square in $\mathbb{R}^2$. Of course, we could then formulate the problem as simply finding *a* realization (i.e. any one would do), or as finding a realization that satisfies certain criteria. However, the above observation also motivates the following related question:

Unique Realization Problem (URP). *Given a graph $G = (V, E)$ with n vertices and m edges, a vector $\mathbf{d} = (d_{ij}^2)_{(i,j)\in E} \in \mathbb{Q}_+^m$, an integer $k \geq 1$ and a realization $\mathbf{p}$ of $(G, \mathbf{d})$ in $\mathbb{R}^k$,*

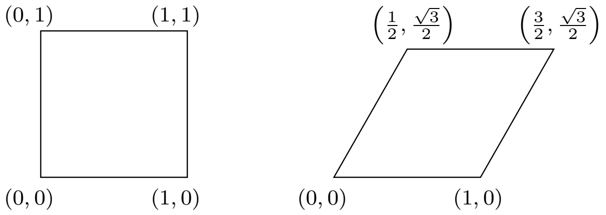

(0, 1)
(1, 1)
(0, 0)
(1, 0)
$\left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$
$\left(\frac{3}{2}, \frac{\sqrt{3}}{2}\right)$
(0, 0)
(1, 0)

Figure 3.1: Two Non–Congruent Realizations of a Unit Square in $\mathbb{R}^2$

decide if there exists another realization of $(G, \mathbf{d})$ *in* $\mathbb{R}^k$ *that is not congruent to* $\mathbf{p}$.

Before we proceed to tackle the above problems, let us first summarize what is known about their complexities.

3.2 Complexity Issues

It turns out that both (GRP) and (URP) are NP–hard, even for $k \geq 1$ fixed (Saxe (1979)). In more recent developments, Aspnes et al. (2004) and Bădoiu et al. (2006) independently showed that (GRP) remains NP–hard even when the input graph is a *unit disk graph.* More precisely, they showed that the following decision problem is NP–hard:

Unit Disk Graph Realization Problem (UDGRP). *Given a graph* $G = (V, E)$ *with* n *vertices and* m *edges, a vector* $\mathbf{d} = (d_{ij}^2)_{(i,j) \in E} \in \mathbb{Q}_+^m$, *an integer* $k \geq 2$ *and a distance bound* $r^2 \in \mathbb{Q}_+$, *decide if there exist vectors* $p_1, \ldots, p_n \in \mathbb{R}^k$ *such that* $d_{ij}^2 = \|p_i - p_j\|^2$ *for all* $(i,j) \in E$, *and that* $\|p_i - p_j\|^2 > r^2$ *for all* $(i,j) \not\in E$.

(For the case where $k = 1$, the problem (UDGRP) can be solved in polynomial time (Bădoiu et al. (2006))). Consequently, unless P=NP, there is no efficient algorithm that, given an instance of (GRP) (resp. (UDGRP)), finds a realization of that instance in $\mathbb{R}^k$ for any fixed $k \geq 1$ (resp. $k \geq 2$).

Now, suppose that we know a given instance of (UDGRP) has a *unique* realization in $\mathbb{R}^2$, say. Then, it seems reasonable to expect that the realization is easy to find. However, Aspnes et al. (2004) showed that this is not the case. Specifically, they showed that unless RP=NP, there does not exist an efficient randomized algorithm that, given an instance of (UDGRP) that has a unique realization in $\mathbb{R}^2$, finds that realization.

In contrast, the problem (URP) has quite a different characteristic, at least for the cases where $k = 1, 2$. Although (URP) is NP–hard, the NP–hardness proof by Saxe relies on special combinations of edge lengths, which result in specific algebraic relations among the coordinates of the vertices. Curiously, by using tools from rigidity theory (see, e.g., Graver et al. (1993) for an introduction to the subject), one can show that if the coordinates of the vertices are unrelated, then the problem becomes tractable for the cases where $k = 1, 2$. Let us now briefly review this approach. To begin, let $G = (V, E)$ be a graph with n vertices, and let $\mathbf{p} = (p_1, \ldots, p_n) \in \mathbb{R}^{kn}$ be a *configuration* of G in $\mathbb{R}^k$, i.e. vertex $i \in V$ is assigned the coordinates $p_i \in \mathbb{R}^k$ for $1 \le i \le n$. We say that $\mathbf{p}$ is *generic* if there does not exist a non–zero polynomial $h : \mathbb{R}^{kn} \to \mathbb{R}$ with integer coefficients such that $h(p_1, \ldots, p_n) = 0$. A *framework* in $\mathbb{R}^k$, which is denoted by $G(\mathbf{p})$, is a graph G together with its configuration $\mathbf{p}$. The usage of the term "framework" is motivated by the desire to view the graph as a physical structure, where the vertices represent joints and edges represent bars. We say that two frameworks $G(\mathbf{p})$ and $G(\mathbf{q})$ in $\mathbb{R}^k$ are *equivalent* (denoted by $G(\mathbf{p}) \equiv G(\mathbf{q})$) if $\|p_i - p_j\| = \|q_i - q_j\|$ for all $(i, j) \in E$. A framework $G(\mathbf{p})$ in $\mathbb{R}^k$ is called *rigid* if there exists an $\epsilon > 0$ such that for any other configuration $\mathbf{q}$ in $\mathbb{R}^k$ with $\|\mathbf{p} - \mathbf{q}\| < \epsilon$ and $G(\mathbf{p}) \equiv G(\mathbf{q})$, we have $\mathbf{p} \equiv \mathbf{q}$. A framework $G(\mathbf{p})$ in $\mathbb{R}^k$ is called *globally rigid* if $G(\mathbf{p}) \equiv G(\mathbf{q})$ implies that $\mathbf{p} \equiv \mathbf{q}$.

Before we proceed, let us make some comments on the definitions. First, a configuration of G is just an *arbitrary* assignment of coordinates to the vertices of G, while a realization of $(G, \mathbf{d})$ is an assignment of coordinates to the vertices of G such that the distances between adjacent vertices match those given by $\mathbf{d}$. In particular, a framework $G(\mathbf{p})$ is globally rigid if and only if any *realization* of $(G, \mathbf{d})$, where $\mathbf{d} = (\|p_i - p_j\|^2)_{(i,j)\in E}$, is congruent to $\mathbf{p}$. Secondly, it is easy to see that a globally rigid framework is rigid. However, the converse is not true (see, e.g., Connelly (2005)).

With the above definitions, it is natural to formulate the generic version of (URP) as the problem of deciding whether the framework $G(\mathbf{p})$ is globally rigid in $\mathbb{R}^k$ given a graph G, an integer $k \ge 1$ and a generic configuration $\mathbf{p}$ of G in $\mathbb{R}^k$. Note that a priori the problem has both combinatorial and geometric aspects, since the answer may depend both on the graph G and on the configuration $\mathbf{p}$. In addition, it is not clear how to efficiently check whether a configuration is generic or not. This suggests that one way to tackle the problem is to first try to separate the combinatorial and geometric aspects. In particular, we could formulate the following problem:

Generic Global Rigidity Problem (GGRP). *Given a graph $G = (V, E)$ and an integer $k \ge 1$, decide whether G is* generically globally rigid *in $\mathbb{R}^k$, i.e. whether the framework $G(\mathbf{p})$*

is globally rigid for all *generic configurations* $\mathbf{p}$ *of* G *in* $\mathbb{R}^k$.

At first sight the problem (GGRP) may seem ambitious. However, it is in fact quite natural. Indeed, suppose that $\mathbf{p}$ is a generic configuration of G in $\mathbb{R}^k$. Then, it is known (see, e.g., Graver et al. (1993)) that the rigidity of the framework $G(\mathbf{p})$ in $\mathbb{R}^k$ depends only on G, i.e. rigidity is a generic property. In particular, if we restrict ourselves to generic configurations, then rigidity is a purely combinatorial concept. Thus, it is natural to hope that the same holds for global rigidity, and this leads us to the problem (GGRP). Unfortunately, it is not known whether global rigidity is a generic property, except for the cases where $k = 1, 2$ (see, e.g., Connelly (2005)). For $k = 1$, it is not hard to see that a graph G is generically globally rigid if and only if it is vertex 2–connected. (Recall that a graph is vertex l–connected if it takes the removal of at least l vertices of G to break the graph into more than one component.) As a corollary, we see that for $k = 1$, the problem (GGRP) can be solved in linear time by testing the biconnectivity of G (see, e.g., Aho et al. (1974)). For $k = 2$, the following result is known:

Theorem 3.2.1 *Let* $G = (V, E)$ *be a graph with at least* 3 *vertices. Then,* G *is generically globally rigid in* $\mathbb{R}^2$ *if and only if*

(1) G *is vertex* 3*–connected, and*

(2) G *is* redundantly rigid *in* $\mathbb{R}^2$*, i.e. for any generic configuration* $\mathbf{p}$ *of* G *in* $\mathbb{R}^2$*, the framework* $G_e(\mathbf{p})$ *is rigid in* $\mathbb{R}^2$ *for any* $e \in E$*, where* $G_e = (V, E\backslash\{e\})$.

Note that if G is redundantly rigid, then the framework $G(\mathbf{p})$ is rigid for any generic configuration $\mathbf{p}$. The necessity of the two conditions is first proven in Hendrickson (1992). The sufficiency, however, is more difficult and is only established recently in Jackson and Jordán (2005). The upshot of Theorem 3.2.1 is that it provides the basis for an efficient algorithm that decides the generic global rigidity of a graph in $\mathbb{R}^2$. Indeed, the triconnectivity of a graph can be tested in linear time (Hopcroft and Tarjan (1973)). Moreover, Hendrickson (1992) showed that the two–dimensional redundant rigidity of a graph can be tested in quadratic time. Thus, we conclude that for $k = 2$, the problem (GGRP) can be solved in quadratic time.

Despite the apparent tractability of (GGRP) for the cases where $k = 1, 2$, several interesting issues remain. First, the case where $k = 3$ is most relevant for various applications, and yet there is still no algorithm for determining whether a graph is generically globally rigid in $\mathbb{R}^3$, despite decades of research efforts. Secondly, recall that our original goal is to solve the Graph Realization Problem. However, in view of the hardness result by Aspnes et al. (2004) that we mentioned earlier, even if we know that a graph is generically globally rigid, it is not clear whether the corresponding Graph Realization Problem is any easier. (Aspnes et al. (2006) claimed that the problem of finding the realization of a globally rigid weighted graph in $\mathbb{R}^2$ is NP–hard. However, their reduction involves the use of transcendental numbers. Thus, it is not clear whether the reduction can be computed in polynomial time in the Turing machine model.) Such a disconnection can be partly attributed to the fact the notions of rigidity developed in the literature focus solely on the combinatorial aspect of the problem. Based on these observations, it is then natural to ask the following questions:

(1) Can we develop other more computationally–friendly notions of rigidity?

(2) Can we use the notions of rigidity developed in (1) to tackle (GRP)? In particular, do those notions exploit both the combinatorial and geometric aspects of (GRP) and aid the design of algorithms for it?

(3) Since (GRP) is NP–hard in general, can we identify families of instances for which the realization problem can be efficiently solved?

Due to the wide applicability of (GRP), a lot of efforts has been devoted to address question (3). However, to the best of our knowledge, our work (So and Ye (2007)) was the first to address questions (1) and (2). In the process, we also identified a large class of instances for which the realization problem can be efficiently solved (up to any desired degree of accuracy). Before we give the details of our approach in Section 3.4, let us review some of the works in the literature that address question (3) above.

3.3 Some Special Cases of the Graph Realization Problem

3.3.1 Complete Graph

Let us first consider the case where $G = K_n$, the complete graph on n vertices. Then, the problem (GRP) becomes a question of isometric embeddability of a finite (semi–)metric into an Eucliden space, which is well–studied in analysis (see, e.g., Wells and Williams (1975)). (Recall that a semi–metric is a metric, except that it does not necessarily satisfy the triangle inequality.) This problem was first solved by Menger (1931), who gave a necessary and sufficient condition involving the signs of certain determinants. An alternative solution was later proposed by Schoenberg (1935), and independently by Young and Householder (1938), who gave a necessary and sufficient condition involving the positive semidefiniteness of a certain quadratic form. We shall present the Schoenberg–Young–Householder solution, not only because it allows us to find the realization efficiently, but it also motivates the use of semidefinite programming for tackling the realization problem in many recent works (see, e.g., Alfakih and Wolkowicz (1998); Alfakih et al. (1999); Biswas et al. (2006a); So and Ye (2007); Ding et al. (2006)). We remark that the Schoenberg–Young–Householder result also forms the basis for the classical approach to multidimensional scaling, where various algorithms were developed for constructing a configuration of points in $\mathbb{R}^k$ (where k is part of the input) such that the induced distance matrix matches or approximates the given (complete) distance matrix (see, e.g., Torgerson (1952); Gower (1966)).

Theorem 3.3.1 *(Schoenberg (1935); Young and Householder (1938)) Let G be the complete graph on $n+1$ vertices, where $n \geq 1$. Then, the set of weights $\mathbf{d} = (d_{ij}^2)_{0 \leq i,j \leq n}$ admits a realization $(p_0, p_1, \ldots, p_n)$ in $\mathbb{R}^k$ but not in $\mathbb{R}^{k-1}$ (i.e. $d_{ij}^2 = \|p_i - p_j\|^2$ for $0 \leq i,j \leq n$) if and only if the $n \times n$ symmetric matrix $A = [a_{ij}]_{1 \leq i,j \leq n}$ defined by:*

$$a_{ij} = \frac{1}{2}\left(d_{0i}^2 + d_{0j}^2 - d_{ij}^2\right) \tag{3.1}$$

is positive semidefinite and has rank k.

Proof. Suppose that there exist $n+1$ points $p_0, p_1, \ldots, p_n \in \mathbb{R}^k$ such that $d_{ij}^2 = \|p_i - p_j\|^2$ for $0 \leq i,j \leq n$, and that $\mathrm{span}(\{p_0, p_1, \ldots, p_n\}) = \mathbb{R}^k$. Without loss of generality, we may assume that $p_0 = \mathbf{0}$. Upon writing $p_i - p_j = (p_i - p_0) + (p_0 - p_j)$, we see that:

$$\|p_i - p_j\|^2 = ((p_i - p_0) + (p_0 - p_j))^T((p_i - p_0) + (p_0 - p_j)) = \|p_i - p_0\|^2 + \|p_j - p_0\|^2 + 2p_i^T p_j$$

or equivalently,

$$p_i^T p_j = \frac{1}{2}\left(\|p_i - p_0\|^2 + \|p_j - p_0\|^2 - \|p_i - p_j\|^2\right) = \frac{1}{2}\left(d_{0i}^2 + d_{0j}^2 - d_{ij}^2\right) = a_{ij}$$

It follows that for any $x \in \mathbb{R}^n$, we have:

$$x^T A x = \sum_{i,j=1}^{n} a_{ij} x_i x_j = \left\| \sum_{i=1}^{n} p_i x_i \right\|^2 \geq 0$$

i.e. A is positive semidefinite. Moreover, if we let V to be the $k \times n$ matrix whose i–th column is the vector p_i (where $1 \leq i \leq n$), then we see that $A = V^T V$. Since $\mathrm{span}(\{p_0, p_1, \ldots, p_n\}) = \mathbb{R}^k$, we have $\mathrm{rank}(A) = k$.

Conversely, let $v_0, v_1, \ldots, v_n$ be the vertices of G, and suppose that A is positive semidefinite and has rank $k \leq n$. Then, by Theorem 2.2.1, we may write $A = U \Lambda U^T$ for some $n \times n$ orthogonal matrix U and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, where $\lambda_1 \geq \cdots \geq \lambda_k > \lambda_{k+1} = \cdots = \lambda_n \geq 0$ are the eigenvalues of A. Now, let $e_1, \ldots, e_n$ be the standard basis vectors of $\mathbb{R}^n$, and set $p_i = P_k \Lambda^{1/2} U^T e_i \in \mathbb{R}^k$ for $1 \leq i \leq n$, where P_k is the orthogonal projection of $\mathbb{R}^n$ onto the first k coordinates. It is easy to see that:

$$d_{0i}^2 = a_{ii} = e_i^T A e_i = e_i^T U \Lambda U^T e_i = \|\Lambda^{1/2} U^T e_i\|^2 = \|p_i\|^2 \qquad \text{for } 1 \leq i \leq n$$

and

$$d_{ij}^2 = a_{ii}^2 + a_{jj}^2 - 2a_{ij} = (e_i - e_j)^T A (e_i - e_j) = \|p_i - p_j\|^2 \qquad \text{for } 1 \leq i, j \leq n$$

Thus, the map that sends v_0 to $\mathbf{0}$ and v_i to p_i for $1 \leq i \leq n$ defines the desired realization, and the proof is completed. ■

Theorem 3.3.1 immediately yields a polynomial time algorithm for solving (GRP) in the case where $G = K_n$. Indeed, it suffices to use the algorithms described in Section 2.2.3 to check the rank and the positive semidefiniteness of the matrix A. To obtain a realization $\mathbf{p} = (p_0, p_1, \ldots, p_n)$ of $(G, \mathbf{d})$ in $\mathbb{R}^k$, we first compute the LDL^T decomposition of A (see Section 2.2.3). By using suitable permutation matrices and invoking Theorem 2.2.6, we may then rewrite A as $A = V D' V^T$, where V is some $n \times n$ matrix, and $D' = \mathrm{diag}(d_1', \ldots, d_n')$ with $d_1', \ldots, d_k' > 0$ and $d_{k+1}' = \cdots = d_n' = 0$. If we assume that square roots can be computed, then we can set $p_0 = \mathbf{0} \in \mathbb{R}^k$ and $p_i = P_k (D')^{1/2} V^T e_i \in \mathbb{R}^k$ for $1 \leq i \leq n$. The correctness of this procedure follows immediately from the proof of Theorem 3.3.1. Note that since $G = K_n$, the realization thus obtained is unique up to congruence. Finally, it is clear that the above procedure can be executed in polynomial time in the real RAM model.

We summarize as follows:

Corollary 3.3.2 *The problem (GRP) with $G = K_n$ can be solved in polynomial time. Moreover, a realization can be found in polynomial time in the real RAM model.*

It is often useful to express the conditions for isometric embeddability in terms of the matrix $A' = [d_{ij}^2]_{0 \le i,j \le n}$, whose entries do not favor any particular vertex. (In contrast, the vertex v_0 plays a special role in the definition of the matrix A.) The following corollary provides one such characterization (see, e.g., Schoenberg (1938); Blumenthal (1953)):

Corollary 3.3.3 *Let G be the complete graph on $n+1$ vertices, where $n \ge 1$. Then, the set of weights $\mathbf{d} = (d_{ij}^2)_{0 \le i,j \le n}$ admits a realization in $\mathbb{R}^k$ but not in $\mathbb{R}^{k-1}$ if and only if the $(n+1) \times (n+1)$ matrix $A' = [a'_{ij}]_{0 \le i,j \le n}$ defined by $a'_{ij} = d_{ij}^2$ is negative semidefinite of rank k on the hyperplane $H = \left\{ x \in \mathbb{R}^{n+1} : e^T x = 0 \right\}$.*

Proof. It suffices to show that A' is negative semidefinite of rank k on the hyperplane H if and only if A is positive semidefinite of rank k. First, observe that:

$$\sum_{i,j=0}^n d_{ij}^2 x_i x_j = 2x_0 \sum_{i=0}^n d_{0i}^2 x_i + \sum_{i,j=1}^n d_{ij}^2 x_i x_j$$

and hence for all $x = (x_0, x_1, \ldots, x_n) \in H$, we have:

$$\begin{aligned}
\sum_{i,j=0}^n d_{ij}^2 x_i x_j &= -2 \left(\sum_{i=1}^n x_i \right) \left(\sum_{i=0}^n d_{0i}^2 x_i \right) + \sum_{i,j=1}^n d_{ij}^2 x_i x_j \\
&= -2 \sum_{i,j=1}^n d_{0i}^2 x_i x_j + \sum_{i,j=1}^n d_{ij}^2 x_i x_j \\
&= -\sum_{i,j=1}^n d_{0i}^2 x_i x_j - \sum_{i,j=1}^n d_{0j}^2 x_i x_j + \sum_{i,j=1}^n d_{ij}^2 x_i x_j \\
&= -\sum_{i,j=1}^n \left(d_{0i}^2 + d_{0j}^2 - d_{ij}^2 \right) x_i x_j
\end{aligned}$$

This completes the proof. ∎

We remark that Corollary 3.3.3 also provides a characterization of the set of *Euclidean distance matrices* (an $n \times n$ matrix $A = [a_{ij}]_{1 \le i,j \le n}$ is called an Euclidean distance matrix if there exist vectors $p_1, \ldots, p_n \in \mathbb{R}^k$ for some $k \geq 1$ such that $a_{ij} = \|p_i - p_j\|^2$ for $1 \leq i, j \leq n$). Specifically, an $n \times n$ matrix A is an Euclidean distance matrix if and only if it is negative semidefinite on the hyperplane $H = \{x \in \mathbb{R}^n : e^T x = 0\}$. In the sequel, we shall use the phrase "A is an *Euclidean distance matrix of rank* k" to mean that A is negative semidefinite of rank k *on* H. Note that A need not have rank k itself.

3.3.2 Euclidean Distance Matrix Completion

Although Corollary 3.3.3 applies only to the complete graph, it provides an important idea for tackling the general case, namely that of Euclidean distance matrix completion. Specifically, let $G = (V, E)$ be a graph with n vertices, and let $\mathbf{d} = (d_{ij}^2)_{(i,j) \in E}$ be a set of weights on the edges. Let A be a partially specified $n \times n$ matrix whose (i, j)–th entry is d_{ij}^2 if $(i, j) \in E$ and is left unspecified otherwise. Then, the problem (GRP) is equivalent to the question of whether the matrix A can be *completed* to an Euclidean distance matrix of rank at most k, i.e. whether there exists a choice for the unspecified entries of A so that A is an Euclidean distance matrix of rank at most k. Of course, since (GRP) is NP–hard, the rank–constrained completion problem is also NP–hard. On the other hand, the problem may become more tractable if we relax the rank requirement. This leads us to the following problem:

Euclidean Distance Matrix Completion Problem (EDMCP). *Given a graph $G = (V, E)$ with n vertices and m edges, and a vector $\mathbf{d} = (d_{ij}^2)_{(i,j) \in E} \in \mathbb{Q}_+^m$, decide if there exist vectors $p_1, \ldots, p_n \in \mathbb{R}^k$ for* some $k \geq 1$ *such that $d_{ij}^2 = \|p_i - p_j\|^2$ for all $(i, j) \in E$.*

The problem (EDMCP) is closely related to the *Positive Semidefinite Matrix Completion Problem (PSMCP)*, and both problems have received a lot of attention in the literature (see, e.g., the article by Laurent (1998a) and the surveys by Laurent (1997, 1998b, 2001)). Currently, the exact complexity of (EDMCP) is not known (Laurent (1998b)). In particular, it is not known whether (EDMCP) belongs to NP in the Turing machine model. On the other hand, several special cases of (EDMCP) are polynomial time solvable. Before we introduce those cases, we need some definitions from graph theory. A *chord* of a cycle C is an edge not in C that has endpoints in C. A *chordless cycle* in a graph G is a cycle of length at least 4 in G that has no chord. We say that a graph is *chordal* if it has no chordless cycle. The *minimum fill–in* of a graph G is the minimum number of edges that need to be added to G in order to obtain a chordal graph. A *minor* of a graph G is a graph that can be obtained from G via a sequence of edge deletions and edge contractions.

We are now ready to state the following result:

Theorem 3.3.4 *(Laurent (2000))*

(1) For any fixed integer $m \geq 0$, the problem (EDMCP) can be solved in polynomial time in the Turing machine model when restricted to the class of graphs whose minimum fill–in is m. Moreover, a rational *completion can be found in polynomial time if it exists. For the cases where $m = 0, 1$, the existence of a completion implies the existence of a* rational *completion.*

(2) The problem (EDMCP) can be solved in polynomial time in the real RAM model when restricted to the class of graphs that do not contain K_4 as a minor.

We remark that in general, it is not known whether the existence of a completion of a partially specified rational matrix implies the existence of a rational completion (Laurent (2000)). Theorem 3.3.4 gives an affirmative answer only for the graphs whose minimum fill–in is at most 1. Also, note that for the graphs considered above, there may be many non–congruent realizations. The algorithms behind Theorem 3.3.4 only guarantee that one realization will be returned.

At first sight the results of Theorem 3.3.4 may seem restrictive. However, they apply to a large class of graphs. For instance, it is known that k–trees (see Chapter 4 for the definition) are chordal graphs, and that the class of graphs that do not contain K_4 as a minor is precisely the class of series–parallel graphs (again, see Chapter 4 for the definition). Although Theorem 3.3.4 do not apply to the rank–constrained version of (EDMCP) in general, it turns out that the existence of a completion for an k–tree (resp. a series–parallel graph) implies the existence of a completion of rank at most k (resp. at most 2). We shall investigate this in much greater detail in Chapter 4.

A Semidefinite Programming Formulation

As we mentioned earlier, the exact complexity of (EDMCP) for general graphs is not known. However, it is not hard to see that (EDMCP) can be formulated as a semidefinite program, and hence the weak optimization problem associated with it can be solved efficiently (see Section 2.3.3). Indeed, let $G = (V, E)$ be a connected graph with $V = \{0, 1, \ldots, n\}$, and let $\mathbf{d} = (d_{ij}^2)_{(i,j)\in E}$ be a set of weights on the edges. Consider the following SDP, where we assume without loss that vertex 0 is pinned at the origin:

$$\begin{array}{rcll} E_{ij} \bullet X & = & d_{ij}^2 & \text{for } (i,j) \in E;\ i,j \neq 0 \\ E_j \bullet X & = & d_{0j}^2 & \text{for } (0,j) \in E \\ X & \in & \mathcal{S}_+^n & \end{array} \tag{3.2}$$

Here, $E_{ij} = (e_i - e_j)(e_i - e_j)^T$ for $1 \leq i < j \leq n$ and $E_j = e_j e_j^T$ for $1 \leq j \leq n$, where $e_1, \ldots, e_n$ are the standard basis vectors of $\mathbb{R}^n$. We claim that the set of solutions to (3.2) corresponds precisely to the set of realizations of $(G, \mathbf{d})$. To see this, suppose that $p_1, \ldots, p_n \in \mathbb{R}^k$ is a realization of $(G, \mathbf{d})$ for some $k \geq 1$. Let V be the $k \times n$ matrix whose i–th column is p_i, where $1 \leq i \leq n$, and set $X = V^T V \succeq \mathbf{0}$. Then, we have:

$$E_{ij} \bullet X = \|V(e_i - e_j)\|^2 = \|p_i - p_j\|^2 = d_{ij}^2$$

and

$$E_j \bullet X = \|V e_j\|^2 = \|p_j\|^2 = d_{0j}^2$$

i.e. X is feasible for (3.2). Conversely, suppose that X is feasible for (3.2) and has rank k. By Theorem 2.2.2, there exists an $k \times n$ matrix V such that $X = V^T V$. Then, the columns of V provide a realization of $(G, \mathbf{d})$ in $\mathbb{R}^k$. As a corollary, we see that the problem (EDMCP) is equivalent to the problem of determining the feasibility of (3.2).

To show that the weak optimization problem associated with (3.2) can be solved efficiently, we still need to show that the feasible set, if non–empty, contains a point whose norm is polynomially bounded. However, this is straightforward from the geometry of the problem. Indeed, since G is connected, no vertex can be further than a distance of $M \equiv \sum_{(i,j)\in E} d_{ij}$ away from the origin. Thus, any solution X to (3.2) must satisfy $X_{ii} \leq M^2$ for $1 \leq i \leq n$, which, together with the fact that X is positive semidefinite, implies that $\|X\|$ is polynomially bounded (see Theorem 2.2.7).

Rank Considerations of the SDP Formulation

One useful feature of the formulation (3.2) is that the rank of the solution matrix X corresponds to the dimension of the realization that can be obtained from X. (In fact, the matrix X is simply the Gram matrix of some realization.) Since ultimately we are interested in finding a solution that has a given rank, it is natural to ask whether the system (3.2) admits such a rank–constrained solution. This question was first studied by Barvinok (1995) in the context of quadratic maps, and independently by Pataki (1996, 1998) in the context of semidefinite programs. They showed that a solution to (3.2) of rank at most

$$k^* = \left\lfloor \frac{\sqrt{8m+1}-1}{2} \right\rfloor = \Theta(\sqrt{m}) \tag{3.3}$$

exists whenever (3.2) is feasible. In fact, they showed that the above result holds under a more general setting. Specifically, let $A_1, \ldots, A_m \in \mathcal{S}^n$ and $b \in \mathbb{R}^m$ be given. Consider the following system of equations:

$$\mathcal{A}X = b, \;\; X \in \mathcal{S}^n_+ \tag{3.4}$$

Then, the following is true:

Theorem 3.3.5 *(Barvinok (1995); Pataki (1996, 1998)) Suppose that (3.4) is feasible. Then, there exists a solution to (3.4) of rank at most k^*, where k^* is given by (3.3).*

We remark that Barvinok's proof of Theorem 3.3.5 is non–constructive. In contrast, Pataki (1998) gave an efficient procedure that converts any solution to (3.4) into one that has the desired rank bound. The proof below essentially follows that of Pataki (1998).

Proof. Let X be feasible for (3.4), and assume without loss of generality that $\mathrm{rank}(X) = k > k^*$. Note that the definition of k^* implies that $k(k+1)/2 > m$. By Theorem 2.2.2, there exists an $k \times n$ matrix V such that $X = V^T V$. Now, consider the following system of *homogeneous* linear equations:

$$(VA_iV^T) \bullet W = 0 \qquad \text{for } 1 \le i \le m \tag{3.5}$$

where $W \in \mathcal{S}^k$ but not necessarily definite. The system (3.5) has $k(k+1)/2$ unknowns and m equations. Thus, if $k(k+1)/2 > m$, then there exists a symmetric matrix $\tilde{W} \neq \mathbf{0}$ that satisfies all m equations. Without loss of generality, suppose that $\tilde{W}$ has at least one negative eigenvalue (otherwise, by the homogeneity of (3.5), we can consider $-\tilde{W}$ instead). Set $\tilde{W}(\alpha) = I + \alpha\tilde{W}$, and let $\alpha^* = 1/|\lambda|$, where λ is the smallest eigenvalue of $\tilde{W}$. Note that $\tilde{W}(\alpha^*) \succeq \mathbf{0}$, and it has at least one zero eigenvalue. It follows that $\mathrm{rank}(\tilde{W}(\alpha^*)) < k$. Moreover, we have:

$$(VA_iV^T) \bullet \tilde{W}(\alpha^*) = (VA_iV^T) \bullet I = A_i \bullet (V^T V) = b_i \qquad \text{for } 1 \le i \le m$$

whence $V^T\tilde{W}(\alpha^*)V$ is feasible for (3.4), and that $\text{rank}(V^T\tilde{W}(\alpha^*)V) < k$. Now, we can repeat the procedure until (3.5) has only the zero solution. Then, we necessarily have $k(k+1)/2 \leq m$, or $k \leq k^*$ as desired.

Note that the number of iterations needed in the above procedure is bounded by $n-1$. Moreover, each iteration requires $O(m^2n)$ arithmetic operations. ■

It is not hard to see that the bound given in Theorem 3.3.5 is sharp in the worst case. Indeed, a complete graph on n vertices with unit weight on all of its $m = n(n-1)/2$ edges can be realized in $\mathbb{R}^k$ only if $k \geq n-1$. On the other hand, Barvinok (2001) showed that this example is in some sense the only one that defies the conclusion of Theorem 3.3.5. Specifically, in the context of (EDMCP), he proved the following sharpening of Theorem 3.3.5:

Theorem 3.3.6 *(Barvinok (2001)) Let $G = (V, E)$ be a connected graph with $m = (k+2)(k+1)/2$ edges for some $k \geq 1$, and let $\mathbf{d}$ be a set of weights on the edges. Suppose that G is not a complete graph on $k+2$ vertices. Then, there exists a solution to (3.2) of rank at most k, provided that the system (3.2) is feasible.*

In contrast, Theorem 3.3.5 only guarantees that the instance $(G, \mathbf{d})$, where G has $m \leq \frac{(k+2)(k+1)}{2} - 1$ edges, can be realized in $\mathbb{R}^k$ if it can be realized at all.

We remark that Barvinok's proof of Theorem 3.3.6 is non–constructive. A constructive proof is recently discovered by Ai et al. (2006). More precisely, under the setting of Theorem 3.3.6, Ai et al. (2006) were able to develop an efficient procedure that, for any $\epsilon > 0$, computes an ϵ–approximate solution X to (3.2) of rank at most k, provided that the system (3.2) is feasible.

3.4 A Semidefinite Programming Approach to the Graph Realization Problem

The discussion in the previous section suggests that the semidefinite programming formulation (3.2) of (EDMCP) can be used as a *convex relaxation* of (GRP). Such a relaxation is often useful in practice, as it provides efficiently computable solutions that can serve as starting points for many rank–reduction procedures. Indeed, several other relaxations of (GRP) have been proposed in the literature (see, e.g., Doherty et al. (2001); Tseng (2007); Nie (2006); Wang et al. (2006)), and they have varying degrees of success in experimental settings. However, from a theoretical perspective, it is interesting to study under what circumstances is a relaxation *exact*, i.e. it solves the original problem. We now address this question for the relaxation given in Section 3.3.2. The materials are based on our work in So and Ye (2007), which is, to the best of our knowledge, the first to establish theoretical results concerning the exactness of a general realization algorithm. Furthermore, in the process of developing those results, we are able to develop new notions of rigidity that are more computationally–friendly. We believe that those notions will be of independent interest.

3.4.1 A Semidefinite Relaxation of the Graph Realization Problem

Since we will be generalizing the formulation (3.2) to allow multiple *pinned* vertices (i.e. vertices whose positions are known), let us begin by setting up the notations. Let $G = (V, E)$ be a graph, and let $k \geq 1$ be an integer. Let $V_1 = \{1, \ldots, n\}$ and $V_2 = \{n+1, \ldots, n+m\}$ be a partition of V. The vertices in V_1 (resp. V_2) are said to be *unpinned* (resp. *pinned*). Specifically, let $\mathbf{a} = (a_i)_{i \in V_2}$ be given, where $a_i \in \mathbb{R}^k$ for all $i \in V_2$. Then, the vertex $i \in V_2$ is constrained to be at a_i, while there are no such restrictions on the vertices in V_1. For our purposes, we may assume that $V_2 \neq \emptyset$, since we can always pin one vertex at the origin. We may also assume that $E' = \{(i,j) : i, j \in V_2\} \subset E$, since the distance between any two pinned vertices is trivially known. Now, let $E_1 = \{(i,j) \in E : i, j \in V_1\}$ be the set of edges between two unpinned vertices, and let $E_2 = \{(i,j) \in E : i \in V_2,\ j \in V_1\}$ be the set of edges between a pinned and an unpinned vertex. Let $\mathbf{d} = (d_{ij}^2)_{(i,j) \in E_1}$ (resp. $\bar{\mathbf{d}} = (\bar{d}_{ij}^2)_{(i,j) \in E_2}$) be a set of weights on the edges in E_1 (resp. E_2). We are then interested in finding vectors $x_1, \ldots, x_n \in \mathbb{R}^k$ such that:

$$\begin{aligned} \|x_i - x_j\|^2 &= d_{ij}^2 && \text{for } (i,j) \in E_1 \\ \|a_i - x_j\|^2 &= \bar{d}_{ij}^2 && \text{for } (i,j) \in E_2 \end{aligned} \tag{3.6}$$

We say that $\mathbf{p} = (p_1, \ldots, p_n) \in \mathbb{R}^{kn}$ is a *realization* of $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a})$ in $\mathbb{R}^k$ if it satisfies (3.6). In general, the system (3.6) is a non–convex optimization problem that is difficult to solve. In the past, global optimization techniques were developed for solving (3.6) (see, e.g., Hendrickson (1995); Moré and Wu (1997)). However, one drawback of those approaches is that convergence is not guaranteed. An alternative approach based on semidefinite programming is proposed in Biswas and Ye (2004) and a semidefinite relaxation of (3.6) is developed. We now briefly review this approach.

Let $X = [x_1\ x_2\ \ldots\ x_n]$ be the $k \times n$ matrix that needs to be determined. Then, for all $(i,j) \in E_1$, we have:

$$\|x_i - x_j\|^2 = (e_i - e_j)^T X^T X (e_i - e_j) = (e_i - e_j)(e_i - e_j)^T \bullet (X^T X)$$

and for all $(i,j) \in E_2$, we have:

$$\|a_i - x_j\|^2 = \begin{pmatrix} a_i \\ -e_j \end{pmatrix}^T [I_k\ X]^T [I_k\ X] \begin{pmatrix} a_i \\ -e_j \end{pmatrix} = \begin{pmatrix} a_i \\ -e_j \end{pmatrix} \begin{pmatrix} a_i \\ -e_j \end{pmatrix}^T \bullet \begin{bmatrix} I_k & X \\ X^T & X^T X \end{bmatrix}$$

Here, e_i is the i–th standard basis vector of $\mathbb{R}^n$ and I_k is the k–dimensional identity matrix. Thus, problem (3.6) becomes that of finding a symmetric matrix $Y \in \mathbb{R}^{n \times n}$ and a matrix $X \in \mathbb{R}^{k \times n}$ that satisfy the following system:

$$\begin{array}{rcll} (e_i - e_j)(e_i - e_j)^T \bullet Y & = & d_{ij}^2 & \text{for } (i,j) \in E_1 \\ \begin{pmatrix} a_i \\ -e_j \end{pmatrix} \begin{pmatrix} a_i \\ -e_j \end{pmatrix}^T \bullet \begin{bmatrix} I_k & X \\ X^T & Y \end{bmatrix} & = & \bar{d}_{ij}^2 & \text{for } (i,j) \in E_2 \\ Y & = & X^T X & \end{array} \tag{3.7}$$

Observe that $Y = X^T X$ is the only non–convex constraint in (3.7). In order to obtain a convex formulation, we relax it to $Y \succeq X^T X$. Note that the resulting problem is indeed convex, since by Theorem 2.2.5, the condition $Y \succeq X^T X$ is equivalent to:

$$Z = \begin{bmatrix} I_k & X \\ X^T & Y \end{bmatrix} \succeq \mathbf{0} \tag{3.8}$$

Thus, we can write the relaxed problem as a standard SDP problem, namely that of finding an $(k+n) \times (k+n)$ symmetric matrix Z to:

$$\begin{array}{lll} \sup & 0 & \\ \text{subject to} & E_{ij} \bullet Z = d_{ij}^2 & \text{for } (i,j) \in E_1 \\ & \bar{E}_{ij} \bullet Z = \bar{d}_{ij}^2 & \text{for } (i,j) \in E_2 \\ & Z \succeq \mathbf{0},\ Z_{1:k,1:k} = I_k & \end{array} \tag{3.9}$$

where $Z_{1:k,1:k}$ is the $k \times k$ principal submatrix of Z indexed by the first k rows (columns),

$$E_{ij} = \begin{pmatrix} \mathbf{0} \\ e_i - e_j \end{pmatrix} \begin{pmatrix} \mathbf{0} \\ e_i - e_j \end{pmatrix}^T \quad \text{and} \quad \bar{E}_{ij} = \begin{pmatrix} a_i \\ -e_j \end{pmatrix} \begin{pmatrix} a_i \\ -e_j \end{pmatrix}^T$$

Note that this formulation forces any feasible solution matrix to have rank at least k.

To derive the dual of (3.9), let $(\theta_{ij})_{(i,j)\in E_1}$ and $(w_{ij})_{(i,j)\in E_2}$ be the dual multipliers of the constraints on E_1 and E_2, respectively. Then, the dual of (3.9) is given by:

$$\begin{array}{ll} \inf & I_k \bullet V + \sum\limits_{(i,j)\in E_1} \theta_{ij} d_{ij}^2 + \sum\limits_{(i,j)\in E_2} w_{ij} \bar{d}_{ij}^2 \\ \text{subject to} & U \equiv \begin{bmatrix} V & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \sum\limits_{(i,j)\in E_1} \theta_{ij} E_{ij} + \sum\limits_{(i,j)\in E_2} w_{ij} \bar{E}_{ij} \succeq \mathbf{0} \\ & \theta_{ij} \in \mathbb{R} \text{ for all } (i,j) \in E_1; \quad w_{ij} \in \mathbb{R} \text{ for all } (i,j) \in E_2 \end{array} \tag{3.10}$$

Note that the dual is always feasible, as $V = \mathbf{0}$, $\theta_{ij} = 0$ for all $(i,j) \in E_1$ and $w_{ij} = 0$ for all $(i,j) \in E_2$ is a feasible solution. Moreover, this solution has a dual objective value of 0. Thus, by the SDP strong duality theorem (Theorem 2.3.6), if the primal is also feasible, then there is no duality gap between (3.9) and (3.10). Moreover, if Z is feasible for (3.9) and U is optimal for (3.10), then by complementarity, we have $\text{rank}(Z) + \text{rank}(U) \le k + n$. In particular, since $\text{rank}(Z) \ge k$, we must have $\text{rank}(U) \le n$.

3.4.2 Analysis of the SDP Relaxation

We now investigate some of the properties of the SDP relaxation (3.9). Our goal is to derive conditions under which the relaxation (3.9) is exact for (3.7). To proceed, let us first investigate in what sense (3.9) is a relaxation of (3.7) and characterize the set of solutions to (3.9). Suppose that $\mathbf{p} = (p_1, \ldots, p_n)$ is a realization of $(G, (\mathbf{d}, \bar{\mathbf{d}}), \bar{\mathbf{a}})$ in $\mathbb{R}^l$ for some $l \ge k$, where $\bar{\mathbf{a}} = (\bar{a}_i)_{i\in V_2}$ and

$$\bar{a}_i = \begin{pmatrix} a_i \\ \mathbf{0} \end{pmatrix} \in \mathbb{R}^l \tag{3.11}$$

We write:

$$p_i = \begin{pmatrix} p_i^1 \\ p_i^2 \end{pmatrix} \qquad \text{for } 1 \le i \le n$$

where $p_i^1 \in \mathbb{R}^k$ and $p_i^2 \in \mathbb{R}^{l-k}$. Let $X^1 = [p_1^1 \; p_2^1 \; \ldots \; p_n^1] \in \mathbb{R}^{k \times n}$ and $X^2 = [p_1^2 \; p_2^2 \; \ldots \; p_n^2] \in \mathbb{R}^{(l-k)\times n}$, and set $\tilde{Y} = (X^1)^T X^1 + (X^2)^T X^2 \in \mathbb{R}^{n \times n}$. Then, we claim that the matrix:

$$\tilde{Z} = \begin{bmatrix} I_k & X^1 \\ (X^1)^T & \tilde{Y} \end{bmatrix} \succeq \mathbf{0} \tag{3.12}$$

is feasible for (3.9). Indeed, for any $(i,j) \in E_1$, we have:

$$\begin{aligned} E_{ij} \bullet \tilde{Z} &= X_{ii}^1 + X_{jj}^1 - 2X_{ij}^1 + X_{ii}^2 + X_{jj}^2 - 2X_{ij}^2 \\ &= \|p_i^1 - p_j^1\|^2 + \|p_i^2 - p_j^2\|^2 \\ &= \|p_i - p_j\|^2 \\ &= d_{ij}^2 \end{aligned}$$

and for any $(i,j) \in E_2$, we have:

$$\begin{aligned} \bar{E}_{ij} \bullet \tilde{Z} &= \|a_i\|^2 - 2\sum_{l=1}^{k} a_{il} X_{lj}^1 + \tilde{Y}_{jj} \\ &= \|a_i\|^2 - 2a_i^T p_j^1 + \|p_j^1\|^2 + \|p_j^2\|^2 \\ &= \|\bar{a}_i - p_j\|^2 \\ &= \bar{d}_{ij}^2 \end{aligned}$$

This establishes the claim. Conversely, suppose that a matrix Z of the form (3.8) is feasible for (3.9). Let $Y - X^T X = (X')^T X'$, where $X = [p_1^1 \; p_2^1 \; \ldots \; p_n^1] \in \mathbb{R}^{k \times n}$ and $X' = [p_1^2 \; p_2^2 \; \ldots \; p_n^2] \in \mathbb{R}^{s \times n}$ with $s = \text{rank}(Y - X^T X)$ (note that such an X' exists because $Y - X^T X \succeq \mathbf{0}$). Then, by repeating the above arguments, we conclude that the vectors

$$p_i = \begin{pmatrix} p_i^1 \\ p_i^2 \end{pmatrix} \in \mathbb{R}^{k+s} \qquad \text{for } 1 \le i \le n \tag{3.13}$$

yield a realization of $(G, (\mathbf{d}, \bar{\mathbf{d}}), \bar{\mathbf{a}})$ in $\mathbb{R}^{k+s}$, where each pinned vertex $a_i \in \mathbb{R}^k$ is augmented to $\bar{a}_i \in \mathbb{R}^{k+s}$ as in (3.11).

Now, recall from Theorem 2.3.9 that the central path associated with the self–dual embedding of (3.9) and (3.10) converges to a pair of maximally complementary solutions.

This fact and the discussion above motivate the following definition:

Definition 3.4.1 *We say that an instance* $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a})$ *is* uniquely k–realizable *if (i) there is a unique realization* $\mathbf{p} = (p_1, \ldots, p_n)$ *of* $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a})$ *in* $\mathbb{R}^k$*, and (ii) there is no realization* $\mathbf{p}'$ *of* $(G, (\mathbf{d}, \bar{\mathbf{d}}), \bar{\mathbf{a}})$ *in* $\mathbb{R}^l$*, where* $l > k$ *and* $\bar{\mathbf{a}}$ *is the augmentation of* $\mathbf{a}$ *to* $\mathbb{R}^l$ *(defined via (3.11)), such that the matrix* $\tilde{Z}$ *associated with* $\mathbf{p}'$ *(defined via (3.12)) has rank greater than* k*.*

Definition 3.4.1 can be viewed as a new notion of rigidity which takes into account both the combinatorial and the geometric aspects of the Graph Realization Problem. Before we proceed, several remarks on the above definition are in order. Informally, condition (ii) requires that the given instance has no realizations in $\mathbb{R}^l$ for $l > k$. Of course, every realization $\mathbf{p}$ in $\mathbb{R}^k$ can be viewed as a realization $\mathbf{p}'$ in $\mathbb{R}^l$ by augmenting $\mathbf{p}$ in the straightforward manner (cf. (3.11)). However, the matrix $\tilde{Z}$ associated with $\mathbf{p}'$ will have rank k, and hence $\mathbf{p}'$ does not violate condition (ii). On the other hand, even if there exists a realization $\mathbf{p}'$ in $\mathbb{R}^l$ whose associated matrix $\tilde{Z}$ has rank greater than k, the resulting frameworks $G(\mathbf{p})$ and $G(\mathbf{p}')$ may be related by a rigid motion in $\mathbb{R}^l$. In other words, the realization $\mathbf{p}'$ is still in some sense trivial. This may occur if the number of pinned vertices is small, or if the unpinned vertices are not well–connected to the pinned ones. For instance, consider a two–dimensional setting, i.e. $k = 2$. Let $V_1 = \{1\}$ and $V_2 = \{2, 3\}$ with $a_2 = (0, 0)$ and $a_3 = (1, 0)$. Let $d_{12} = d_{13} = 1$. Then, $p_1 = (1/2, \sqrt{3}/2)$ is a realization in $\mathbb{R}^2$ and $p_1' = (1/2, 0, \sqrt{3}/2)$ is a realization in $\mathbb{R}^3$. Moreover, it can be easily verified that:

$$\tilde{Z} = \begin{bmatrix} 1 & 0 & \frac{1}{2} \\ 0 & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{bmatrix}$$

and that $\text{rank}(\tilde{Z}) = 3$. However, the resulting frameworks $G(p_1)$ and $G(p_1')$ are related via rotation along the axis formed by the vertices 2 and 3.

Based on the above discussion, we see that condition (ii) in Definition 3.4.1 can be equivalently stated as follows:

(ii') there does not exist $p_1',\ldots,p_n' \in \mathbb{R}^l$, *where* $l > k$, *such that:*

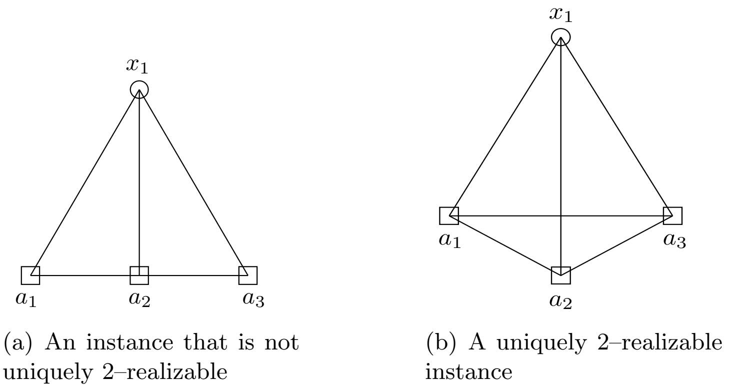
$$\begin{aligned} \|p_i' - p_j'\|^2 &= d_{ij}^2 && \textit{for } (i,j) \in E_1 \\ \|\bar{a}_i - p_j'\|^2 &= \bar{d}_{ij}^2 && \textit{for } (i,j) \in E_2 \\ p_i' &\neq \begin{pmatrix} p_i \\ \mathbf{0} \end{pmatrix} && \textit{for some } 1 \le i \le n \end{aligned}$$

The advantage of this formulation is that it brings out the geometric content of the notion of unique k–realizability.

At this point it is fair to ask whether Definition 3.4.1 is vacuous, i.e. whether uniquely k–realizable instances exist at all. It is not hard to see that they do exist for all $k \ge 1$. For example, let $k \ge 1$ be fixed, and consider the complete graph K_n on $n \ge k+1$ vertices. Suppose that the vertices are numbered $\{1,\ldots,n\}$, and let $\{n-k,\ldots,n\}$ be the set of pinned vertices. Furthermore, suppose that all the pairwise distances are such that the matrix A defined in (3.1) is positive semidefinite of rank k. Then, by Theorem 3.3.1, there exists a unique realization $\mathbf{p} = (p_1,\ldots,p_{n-k-1})$ of the instance $\mathcal{I} = (K_n, (\mathbf{d},\bar{\mathbf{d}}), \mathbf{a})$ in $\mathbb{R}^k$. Now, if every $(k+1)$–point subset of $\{p_1,\ldots,p_{n-k-1},a_{n-k},\ldots,a_n\}$ is affinely independent, then the instance $\mathcal{I}$ is uniquely k–realizable. We remark that without the affine independence assumption, the above conclusion may not hold. For example, the instance shown in Figure 3.2(a) is not uniquely 2–realizable, since we can rotate the framework along the axis formed by the collinear pinned vertices 1, 2 and 3. On the other hand, the instance shown in Figure 3.2(b) is uniquely 2–realizable.

From the above discussion, it may be tempting to conclude that the graph G of any uniquely k–realizable instance is dense, i.e. it has $\Omega(n^2)$ edges, where n is the number of vertices in G. However, we now exhibit a family of uniquely k–realizable instances in which the number of edges scales linearly with the number of vertices. In particular, this refutes a common belief in the literature (see, e.g., Aspnes et al. (2006); Basu et al. (2006)) that the graph of any uniquely k–realizable instance must have $\Omega(n^2)$ edges.

Our construction is based on the so–called *k–trees* and proceeds as follows. Let $k \ge 1$ be fixed, and consider the class $\mathcal{G}_k$ of graphs defined inductively as follows. The complete graph K_{k+1} on $k+1$ vertices belongs to $\mathcal{G}_k$. A graph G with $n+1$ vertices, where $n > k$, belongs to $\mathcal{G}_k$ if it can be obtained by adding a vertex v to some n–vertex graph $G' \in \mathcal{G}_k$ and



(a) An instance that is not uniquely 2–realizable

(b) A uniquely 2–realizable instance

Figure 3.2: Importance of the Affine Independence Assumption

making v to be adjacent to all the vertices of an $(k+1)$–vertex complete subgraph of G'. In particular, given a graph $G \in \mathcal{G}_k$, we can number its vertices according to their order of appearance in the above construction (the first $k+1$ vertices can be numbered arbitrarily). It is straightforward to see that if an $(n+k+1)$–vertex graph G belongs to $\mathcal{G}_k$, then it has $k(k+1)/2 + kn$ edges. Moreover, we have the following theorem, which shows that for any fixed $k \geq 1$, uniquely k–realizable instances whose graphs are sparse (i.e. with $O(n)$ edges) exist:

Theorem 3.4.2 *Let $k \geq 1$ be fixed, and let $G = (V,E)$ be a graph in $\mathcal{G}_k$ with $n+k+1$ vertices. Suppose that the vertices are numbered according to their order of insertion. Let $\{1,\ldots,k+1\}$ be the set of pinned vertices, and let $\mathbf{a} = (a_1,\ldots,a_{k+1})$ be their positions in $\mathbb{R}^k$. Let $\mathbf{p} = (p_{k+2},\ldots,p_n)$ be a realization of the instance $\mathcal{I} = (G, (\mathbf{d},\bar{\mathbf{d}}), \mathbf{a})$ in $\mathbb{R}^k$, and suppose that every $(k+1)$–point subset of $\{a_1,\ldots,a_{k+1},p_{k+2},\ldots,p_n\}$ is affinely independent. Then, the instance $\mathcal{I}$ is uniquely k–realizable, and $\mathbf{p}$ is the unique realization of $\mathcal{I}$ in $\mathbb{R}^k$.*

Proof. The proof proceeds by considering the vertices of G in the order they are inserted. First, consider vertex $k+2$. It is connected to vertices $1,\ldots,k+1$, and hence we have the following constraints in the SDP (see (3.9)):

$$\|a_i\|^2 - 2a_i^T x_{k+2} + Y_{11} = \bar{d}_{i,k+2}^2 \qquad \text{for } 1 \leq i \leq k+1$$

Upon eliminating Y_{11}, we obtain the following system of linear equations:

$$(a_i - a_1)^T x_{k+2} = \frac{1}{2}\left(\|a_i\|^2 - \|a_1\|^2 + \bar{d}_{1,k+2}^2 - \bar{d}_{i,k+2}^2\right) \qquad \text{for } 2 \leq i \leq k+1 \tag{3.14}$$

Since the set $\{a_1, a_2, \ldots, a_{k+1}\}$ is affinely independent, the set $\{a_2 - a_1, \ldots, a_{k+1} - a_1\}$ is linearly independent. Hence, there exists a unique solution $x_{k+2} \in \mathbb{R}^k$ to the system (3.14). It follows that $x_{k+2} = p_{k+2}$, which in turn implies that $Y_{11} = \|p_{k+2}\|^2$. Since $Y - X^T X \succeq \mathbf{0}$ and $(Y - X^T X)_{11} = 0$, by Theorem 2.2.7, we also have $Y_{1l} = Y_{l1} = p_{k+2}^T x_{k+l+1}$ for $1 \le l \le n$.

Now, suppose that for all $k+2 \le i \le j$, we have $x_i = p_i$, $Y_{i-(k+1),i-(k+1)} = \|p_i\|^2$ and $Y_{i-(k+1),l} = Y_{l,i-(k+1)} = p_i^T x_{k+l+1}$ for $1 \le l \le n$. Consider vertex $j+1$. It is connected to $k+1$ of the vertices $\{1, \ldots, j\}$. Thus, we have the following constraints in the SDP:

$$\begin{aligned} \|p_l\|^2 - 2Y_{l-(k+1),j-k} + Y_{j-k,j-k} &= d_{l,j+1}^2 \quad \text{for } (l, j+1) \in E,\ k+2 \le l \le j \\ \|a_i\|^2 - 2a_i^T x_{j+1} + Y_{j-k,j-k} &= \bar{d}_{i,j+1}^2 \quad \text{for } (i, j+1) \in E,\ 1 \le i \le k+1 \end{aligned} \tag{3.15}$$

and the total number of equations in (3.15) is $k+1$. By the inductive hypothesis, we have $Y_{l-(k+1),j-k} = p_l^T x_{j+1}$ for $k+2 \le l \le j$. Moreover, by assumption, any $(k+1)$–point subset of $\{a_1, \ldots, a_{k+1}, p_{k+2}, \ldots, p_j\}$ is affinely independent. Thus, we can proceed as before and conclude that $x_{j+1} = p_{j+1}$ and $Y_{j-k,j-k} = \|p_{j+1}\|^2$. This in turn implies that $Y_{j-k,l} = Y_{l,j-k} = p_{j+1}^T x_{k+l+1}$ for $1 \le l \le n$, and the inductive step is completed. In particular, the above argument shows that $\mathbf{p}$ is the unique realization of $\mathcal{I}$ in $\mathbb{R}^k$, and that condition (ii') of Definition 3.4.1 is satisfied. Hence, the proof is completed. ■

Having established the existence of uniquely k–realizable instances, we are now ready to state and prove the main theorem of this section:

Theorem 3.4.3 *Let $G = (V, E)$ be connected, and let $\mathbf{d}$, $\bar{\mathbf{d}}$ and $\mathbf{a}$ be given. Then, the following are equivalent:*

(1) The instance $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a})$ is uniquely k–realizable.

(2) The max–rank solution matrix of (3.9) has rank k.

(3) The solution matrix of (3.9), represented by (3.8), satisfies $Y = X^T X$.

Proof. The equivalence of (2) and (3) is straightforward. To prove (2)⇒(1), we first observe that any rank–k solution to (3.9) is a solution to (3.7). Now, suppose that there are two rank–k solutions to (3.9), say:

$$Z_1 = \begin{bmatrix} I_k & X_1 \\ X_1^T & X_1^T X_1 \end{bmatrix} \quad \text{and} \quad Z_2 = \begin{bmatrix} I_k & X_2 \\ X_2^T & X_2^T X_2 \end{bmatrix}$$

Then, for any $\alpha \in (0,1)$, the matrix $Z(\alpha) = \alpha Z_1 + (1-\alpha)Z_2$ is feasible for (3.9). Moreover, we must have $\text{rank}(Z(\alpha)) = k$, since any solution to (3.9) has rank at least k but the max–rank is assumed to be k. Therefore, we have:

$$Z(\alpha) = \left[\begin{array}{cc} I_k & \alpha X_1 + \beta X_2 \\ \alpha X_1^T + \beta X_2^T & \alpha X_1^T X_1 + \beta X_2^T X_2 \end{array} \right] = \left[\begin{array}{cc} I_k & B \\ B^T & B^T B \end{array} \right]$$

where $B = \alpha X_1 + (1-\alpha)X_2$. It follows that $(X_1 - X_2)^T(X_1 - X_2) = \mathbf{0}$, or $\|X_1 - X_2\| = 0$, i.e. $Z_1 = Z_2$, which is a contradiction.

To prove (1)⇒(2), suppose that there exists a solution Z to (3.9) whose rank is greater than k. Then, we can extract from Z a realization of $(G, (\mathbf{d}, \bar{\mathbf{d}}), \bar{\mathbf{a}})$ in $\mathbb{R}^{k+s}$ for some $s \geq 1$; see (3.13). This contradicts the unique k–realizability of the given instance. ■

Although unique k–realizability is a useful notion in determining the solvability of the Graph Realization Problem, it is not stable under perturbation. As we shall see, there exist instances that are uniquely k–realizable, but may no longer be so after small perturbation of the unpinned vertices. This motivates us to define another notion called *strong k–realizability*:

Definition 3.4.4 *We say that an instance* $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a})$ *is* strongly k–realizable *if (3.10) has a rank–n optimal dual slack matrix.*

Note that if an instance is strongly k–realizable, then it is uniquely k–realizable by complementarity (see Theorem 2.3.6) and Theorem 3.4.3, since the rank of any solution to (3.9) is equal to k.

Given an instance $\mathcal{I} = (G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a})$, we say that the instance $(G', (\mathbf{d}', \bar{\mathbf{d}}'), \mathbf{a})$ is a *sub–instance* of $\mathcal{I}$ if G' is a subgraph of G that includes all the pinned vertices, and $(\mathbf{d}', \bar{\mathbf{d}}')$ is the restriction of $(\mathbf{d}, \bar{\mathbf{d}})$ on G'. As indicated by the following theorem, the notion of strong k–realizability is very useful in identifying the uniquely k–realizable sub–instances of a given instance:

Theorem 3.4.5 *Suppose that a given instance $\mathcal{I}$ contains a sub–instance $\mathcal{I}'$ that is strongly k–realizable. Then, in any solution to (3.9), the submatrix that corresponds to $\mathcal{I}'$ has rank k.*

Proof. Suppose that the sub–instance $\mathcal{I}'$ has n_s unpinned vertices that are indexed by $1, \ldots, n_s$. Let Z_s be a solution to the SDP relaxation associated with $\mathcal{I}'$. Since $\mathcal{I}'$ is strongly

k–realizable, the dual of the SDP relaxation associated with $\mathcal{I}'$ has a rank–n optimal slack matrix U_s. Now, let

$$U = \begin{bmatrix} U_s & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \succeq \mathbf{0}$$

be a slack matrix for the dual of the SDP relaxation associated with the original instance $\mathcal{I}$. We claim that U is optimal. It is clear that U is dual feasible. Now, observe that any matrix Z that is feasible for the SDP relaxation associated with $\mathcal{I}$ must have the form:

$$Z = \begin{bmatrix} Z_s & * \\ * & * \end{bmatrix} \succeq \mathbf{0} \qquad \text{where } Z_s = \begin{bmatrix} I_d & X_s \\ X_s^T & Y_s \end{bmatrix}$$

since the sub–instance $\mathcal{I}'$ is uniquely k–realizable. Thus, the dual slack matrix U is complementary to any primal feasible matrix Z and hence must be optimal by Theorem 2.3.6. Moreover, since $\text{rank}(U_s) = n_s$, we must have $\text{rank}(Z_s) = k$ by complementarity. In particular, we have $Y_s = X_s^T X_s$, whence X_s is the unique realization in $\mathbb{R}^2$ of the sub–instance. ■

Unfortunately, a rank–n optimal dual slack matrix does not always exist. Thus, it would be useful to have a recipe for determining whether one exists, and if so, constructing one. We begin with the following decomposition result. It is particularly useful for showing that certain instances are *not* strongly k–realizable.

Proposition 3.4.6 *Let Z be feasible for (3.9), and suppose that U is an optimal dual slack matrix of rank n. Then, we have $Z = [I_k \ X]^T[I_k \ X]$. Moreover, we have $U = [-X^T \ I_n]^T W[-X^T \ I_n]$ for some $n \times n$ positive definite matrix W.*

Proof. Since $\text{rank}(U) = n$, by complementarity, we have $\text{rank}(Z) = k$. Hence, the first statement follows. Now, since $U \succ \mathbf{0}$ has rank n, by Theorems 2.2.1 and 2.2.3, it can be decomposed as $U = B\Lambda B^T$, where Λ is an $n \times n$ diagonal matrix with positive diagonal elements, and B is some $n \times n$ matrix. By complementarity, we have $Z \bullet U = 0$, which implies that $(B^T ZB) \bullet \Lambda = 0$. Since $\Lambda \succ \mathbf{0}$ is diagonal and $B^T ZB \succeq \mathbf{0}$, we must have $B^T ZB = \mathbf{0}$. By Theorem 2.2.7, this implies that $ZB = \mathbf{0}$. In other words, we have the following system:

$$ZB = \begin{bmatrix} I_k & X \\ X^T & X^T X \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} = \mathbf{0} \tag{3.16}$$

Clearly, any B that satisfies $B_1 = -XB_2$ is a solution to (3.16). It follows that $U = [-X^T \; I_n]^T (B_2 \Lambda B_2^T)[-X^T \; I_n]$ and $B_2 \Lambda B_2^T \succ \mathbf{0}$ as desired. ■

Now, consider the following block decomposition of the dual slack matrix U:

$$U = \begin{bmatrix} U_{11} & U_{12} \\ U_{12}^T & U_{22} \end{bmatrix} \tag{3.17}$$

where U_{22} is an $n \times n$ matrix. Note that U can be decomposed as $U_{22} = A + D$, where $A_{ij} = -\theta_{ij}$ if $(i,j) \in E_1$, $A_{ii} = \sum_{j:(i,j)\in E_1} A_{ij}$; and D is a diagonal matrix where $D_{ii} = \sum_{(i,j)\in E_2} w_{ij}$. (If there does not exist an j such that $(i,j) \in E_2$, then $D_{ii} = 0$.) Note that if we impose the constraints $\theta_{ij} \geq 0$ for all $(i,j) \in E_1$ and $w_{ij} \geq 0$ for all $(i,j) \in E_2$, then both A and D are positive semidefinite. Moreover, we have the following sufficient condition for strong k–realizability:

Proposition 3.4.7 *Suppose that G is connected, and that the set of pinned vertices is non–empty. Furthermore, suppose that $\theta_{ij} > 0$ for all $(i,j) \in E_1$, and that $w_{ij} > 0$ for all $(i,j) \in E_2$. Then, we have $U_{22} \succ \mathbf{0}$. In particular, we have rank$(U_{22}) = n$.*

Proof. Since A and D are positive semidefinite, we have $x^T U_{22} x \geq 0$ for all $x \in \mathbb{R}^n$. We now show that there is no $x \in \mathbb{R}^n \backslash \{\mathbf{0}\}$ such that $x^T A x = x^T D x = 0$. Suppose to the contrary that we have such an x. Then, since D is diagonal, we have $x^T D x = \sum_{i=1}^n D_{ii} x_i^2 = 0$. In particular, for $D_{ii} > 0$, we have $x_i = 0$. Now, note that:

$$x^T A x = \sum_{(i,j)\in E_1} (x_i - x_j)^2 A_{ij}$$

Thus, $x^T A x = 0$ implies that $x_i = x_j$ for all $(i,j) \in E_1$. Since the set of pinned vertices is non–empty, there exists an i such that $D_{ii} > 0$, whence $x_i = 0$. Since G is connected, it follows that $x = \mathbf{0}$. ■

3.4.3 A Comparison of Notions of Rigidity in $\mathbb{R}^2$

In this section, we show that the notions of unique 2–realizability, strong 2–realizability and global rigidity in $\mathbb{R}^2$ are all distinct.

Unique 2–Realizability $\not\Rightarrow$ Strong 2–Realizability

We have already remarked that a strongly 2–realizable graph is necessarily uniquely 2–realizable. However, the converse is not true. To see this, consider the graph G_1 as shown in Figure 3.3(a), where vertices 1 and 2 are unpinned, and vertices 3, 4 and 5 are pinned at distinct positions a_3, a_4 and a_5 in $\mathbb{R}^2$, respectively. Now, consider an instance $\mathcal{I} = (G_1, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a})$ with the following properties:

(1) $\bar{d}_{23} + \bar{d}_{25} = \|a_3 - a_5\|$, i.e. the unpinned vertex 2 is collinear with the pinned vertices 3 and 5;

(2) $\bar{d}_{25} > 0$, i.e. the unpinned vertex 2 does not co–locate with the pinned vertex 5;

(3) the distance between x_1 and x_2 forces vertex 1 to lie in $\mathbb{R}^2$;

(4) $d_{12} + \bar{d}_{15} > \bar{d}_{25}$ and $d_{12} + \bar{d}_{25} > \bar{d}_{15}$, i.e. the unpinned vertex 1 is not collinear with the pinned vertices 3 and 5;

(5) $\bar{d}_{14} + \bar{d}_{15} > \|a_4 - a_5\|$, i.e. the unpinned vertex 1 is not collinear with the pinned vertices 4 and 5.

It is not hard to see that the instance $\mathcal{I}$ is uniquely 2–realizable under properties (1)–(3). Indeed, by property (1), the position x_2 of vertex 2 is uniquely determined by a_3 and a_5. On the other hand, since a_4 and a_5 are distinct, there are only two possible values for the position x_1 of vertex 1 in $\mathbb{R}^2$, and properties (2) and (3) imply that x_1 is uniquely determined by a_4, a_5 and x_2.

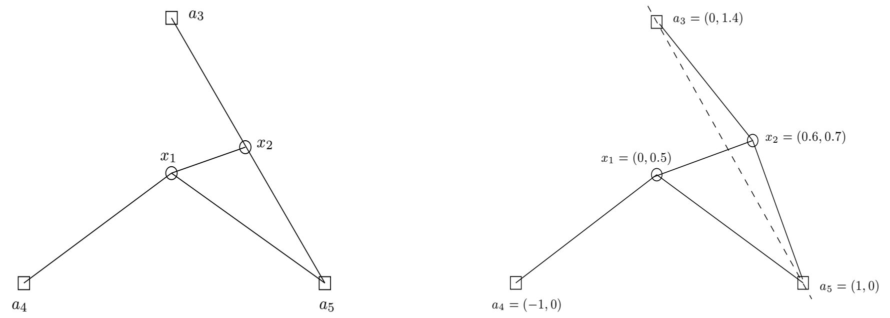
Now, suppose that the instance $\mathcal{I}$ is strongly 2–realizable. Using the block decomposition (3.17), it is easy to verify that:

$$\begin{aligned} U_{12} &= -\left[w_{14}a_4 + w_{15}a_5, w_{23}a_3 + w_{25}a_5\right] \\ U_{22} &= \begin{bmatrix} w_{14} + w_{15} + \theta_{12} & -\theta_{12} \\ -\theta_{12} & w_{23} + w_{25} + \theta_{12} \end{bmatrix} \end{aligned}$$

Moreover, Proposition 3.4.6 requires that $U_{12} = -XU_{22}$. This gives the following system of linear equations:

$$(x_1 - a_4)w_{14} + (x_1 - a_5)w_{15} = (x_2 - x_1)\theta_{12} \tag{3.18}$$

$$(x_2 - a_3)w_{23} + (x_2 - a_5)w_{25} = -(x_2 - x_1)\theta_{12} \tag{3.19}$$



(a) A uniquely 2–realizable, but not strongly 2–realizable instance

(b) A globally rigid instance in $\mathbb{R}^2$ that is not uniquely 2–realizable

Figure 3.3: A Comparison of Notions of Rigidity in $\mathbb{R}^2$

Since x_2 lies on the affine space spanned by a_3 and a_5 but x_1 does not (by properties (1) and (4)), equation (3.19) implies that $\theta_{12} = 0$. However, equation (3.18) would then imply that x_1 lies on the affine space spanned by a_4 and a_5, which contradicts property (5). Thus, we conclude that $\mathcal{I}$ is not strongly k–realizable.

Global Rigidity in $\mathbb{R}^2 \not\Rightarrow$ Unique 2–Realizability

Let $\mathbf{p}$ be the realization of a uniquely 2–realizable instance $\mathcal{I} = (G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a})$ in $\mathbb{R}^2$. Let G' be the graph obtained from G by adding an edge between every pair of pinned vertices. Then, by definition, the framework $G'(\mathbf{p}, \mathbf{a})$ is globally rigid in $\mathbb{R}^2$. However, the converse is not true. To see this, consider the instance $\mathcal{I} = (G_2, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a})$ as shown in Figure 3.3(b) with the given realization $\mathbf{p}$ in $\mathbb{R}^2$. Let G_2' be the graph obtained from G_2 by adding an edge between every pair of pinned vertices. We claim that $G_2'(\mathbf{p}, \mathbf{a})$ is globally rigid in $\mathbb{R}^2$. To see this, let $\mathbf{q}$ be another realization of G_2' in $\mathbb{R}^2$. Without loss of generality, we may assume that $\mathbf{q}$ assigns the same coordinates to the vertices $\{3, 4, 5\}$ as $\mathbf{a}$. Then, $\mathbf{q}$ must satisfy the following system of equations:

$$\|q_1 - (-1, 0)\|^2 = \|q_1 - (1, 0)\|^2 = \frac{5}{4}$$

$$\|q_2 - (0, 1.4)\|^2 = \frac{17}{20}, \quad \|q_2 - (1, 0)\|^2 = \frac{13}{20}$$

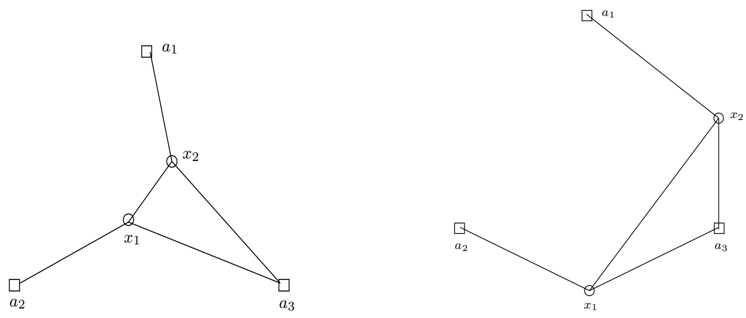
$$\|q_1 - q_2\|^2 = \frac{2}{5}$$

It is straightforward albeit tedious to verify that the above system has a unique solution. It follows that $(\mathbf{p}, \mathbf{a}) \equiv \mathbf{q}$, as desired.

Now, by Theorem 3.4.3, the instance $\mathcal{I}$ can fail to be uniquely 2–realizable only if it has a realization in some higher dimension. Indeed, the above instance has a three–dimensional realization. The idea for constructing such a realization is as follows. Let us first remove the edge $(1, 2)$. Then, reflect the subgraph induced by $\{2, 3, 5\}$ across the dotted line. Now, consider two spheres, one centered at a_4 and the other centered at a_5, both having radius $\sqrt{5}/2$. The intersection of these spheres is a circle, and we can move vertex 1 along this circle until the distance between vertices 1 and 2 is equal to the pre–specified value. Then, we can put the edge $(1, 2)$ back and obtain a three–dimensional realization of the instance.

More precisely, for the above realization, the reflected version of vertex 2 has coordinates $x_2' = \left(\frac{173}{370}, \frac{112}{185}, 0\right)$. Now, let $x_1' = \left(0, \frac{23}{64}, \frac{\sqrt{495}}{64}\right)$. Then, it is straightforward to verify that the new coordinates satisfy all the distance constraints. Hence, we conclude that $\mathcal{I}$ is not uniquely 2–realizable.

It would be nice to have a characterization on those instances that are globally rigid in $\mathbb{R}^2$ but have higher dimensional realizations. However, finding such a characterization remains a challenging task, for such a characterization will necessarily be non–combinatorial and will depend heavily on the geometry of the given instance. For example, the instances shown in Figure 3.4, while having the same combinatorial property as the one shown in Figure 3.3(b), are uniquely 2–realizable (in fact, both are strongly 2–realizable).



(a) (b)

Figure 3.4: Strongly 2–Realizable Instances

3.4.4 Computational Aspects of the Semidefinite Relaxation

Let us now turn to the computational aspects of the semidefinite relaxation (3.9). The following proposition shows that the set of solutions to (3.9) lies within a ball whose radius is polynomially bounded. Thus, the weak optimization problem associated with (3.9) can be solved in polynomial time. In particular, we can obtain an exact realization of a nearby instance in polynomial time.

Proposition 3.4.8 *Let G be such that every unpinned vertex is connected, either directly or indirectly, to a pinned vertex. Then, the set of solutions to (3.9) is contained in a ball whose radius is polynomially bounded. In particular, for any feasible Z, its diagonal entries are polynomially bounded.*

Proof. Suppose that the unpinned vertex $j \in V_1$ is adjacent to a pinned vertex, say $i \in V_2$. Then, we have:

$$\|x_j\|^2 - 2a_i^T x_j + \|a_i\|^2 \le Y_{jj} - 2a_i^T x_j + \|a_i\|^2 = \bar{d}_{ij}^2$$

By the Cauchy–Schwarz inequality, we have $a_i^T x_j \le \|a_i\| \cdot \|x_j\|$. Thus, we have:

$$\|x_j\|^2 - 2\|a_i\| \cdot \|x_j\| \le \bar{d}_{ij}^2 - \|a_i\|^2$$

which implies that $\|x_j\|^2$ is bounded by, say, $2\max\{8\|a_i\|^2, \bar{d}_{ij}^2 - \|a_i\|^2\}$. This in turn implies that:

$$Y_{jj} \le \bar{d}_{ij}^2 + 2\|a_i\| \cdot \|x_j\| - \|a_i\|^2 \le 4\max\left\{8\|a_i\|^2, \bar{d}_{ij}^2 - \|a_i\|^2\right\} \qquad (3.20)$$

Now, consider an unpinned vertex $l \in V_1$. Suppose that it is $h \ge 1$ hops away from the nearest pinned vertex $i \in V_2$ (in terms of hop count). We claim that:

$$Y_{ll} \le \left(\sqrt{4\max\left\{8\|a_i\|^2, D^2\right\}} + (h-1)D\right)^2 \qquad (3.21)$$

where $D^2 = \max\{\max_{(i,j)\in E_1} d_{ij}^2, \max_{(i,j)\in E_2} \bar{d}_{ij}^2\}$. The proof is by induction on h. For $h = 1$, the result follows directly from (3.20). Now, let $j \in V_1$ be the vertex preceding l on the shortest path from i to l. Since j is $h-1$ hops away from i, by the inductive hypothesis, we have:

$$Y_{jj} \le \left(\sqrt{4\max\left\{8\|a_i\|^2, D^2\right\}} + (h-2)D\right)^2$$

Moreover, since $(l,j) \in E_1$, we have:

$$\left(\sqrt{Y_{ll}} - \sqrt{Y_{jj}}\right)^2 = Y_{ll} - 2\sqrt{Y_{ll}Y_{jj}} + Y_{jj} \leq Y_{ll} - 2Y_{lj} + Y_{jj} = d_{lj}^2$$

whence:

$$Y_{ll} \leq \left(\sqrt{Y_{jj}} + d_{lj}\right)^2 \leq \left(\sqrt{4\max\{8\|a_i\|^2, D^2\}} + (h-1)D\right)^2$$

as desired. Since $h \leq n + m$, we conclude that the bound in (3.21) is polynomial in the input sizes. This completes the proof. ■

Finally, let us consider the problem of testing the unique k–realizability of an instance $\mathcal{I}$ for any $k \geq 1$. As mentioned earlier, the definition of unique k–realizability is partly motivated by the fact that the central path associated with the self–dual embedding of (3.9) and (3.10) converges to a pair of maximally complementary solutions. This suggests that we can use a path–following interior–point algorithm for SDP (see, e.g., Alizadeh et al. (1998); Nesterov and Todd (1998); Monteiro and Tsuchiya (1999); Monteiro and Zanjácomo (1999); Nesterov et al. (1999)) to test the unique k–realizability of a nearby instance in polynomial time. Furthermore, we can approximate the realization of a uniquely k–realizable instance up to arbitrary accuracy in polynomial time. Hence, we may argue that when compared to generic global rigidity, unique k–realizability is a more computationally–friendly notion of rigidity.

Chapter 4

Applications to Tensegrity Theory and Graph Realizability

4.1 Introduction

As we saw in the previous chapter, even though the Graph Realization Problem (GRP) is NP–hard in general, there exist families of instances for which the realization problem can be efficiently solved. For example, in Chapter 3, we showed that for every $k \geq 1$, the class of uniquely k–realizable instances can be efficiently realized (up to arbitrary accuracy) in $\mathbb{R}^k$ by using interior–point algorithms for SDP. Now, recall that the unique k–realizability of an instance depends on both the combinatorial and geometric aspects of that instance. In particular, given a graph G and the positions $\mathbf{a}$ of the pinned vertices in $\mathbb{R}^k$, there may exist weight vectors $(\mathbf{d}, \bar{\mathbf{d}})$ and $(\mathbf{d}', \bar{\mathbf{d}}')$ such that both $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a})$ and $(G, (\mathbf{d}', \bar{\mathbf{d}}'), \mathbf{a})$ are realizable in $\mathbb{R}^k$, but one is uniquely k–realizable and the other is not (see Figure 3.3 for an example). Consequently, there can be no combinatorial characterization for the class of uniquely k–realizable instances. The above discussion then raises the following question: for each $k \geq 1$, can one define a family $\mathcal{F}_k$ of instances of the realization problem in $\mathbb{R}^k$ such that (i) $\mathcal{F}_k$ has a combinatorial characterization (i.e. the membership of an instance $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a})$ in $\mathcal{F}_k$ depends only on G), and (ii) instances in $\mathcal{F}_k$ can be (approximately) realized in $\mathbb{R}^k$ efficiently? One of the advantages of such a definition is that it would allow us to approach the Graph Realization Problem from a combinatorial perspective. It turns out that a satisfactory answer to the above question is available for the cases where $k = 1, 2, 3$, which are of course most relevant to practical applications. In Belk and Connelly (2007) and

Belk (2007), the authors introduced the class of *k–realizable graphs* and provided a complete characterization for the cases where $k = 1, 2, 3$. Informally, a graph G is k–realizable if, given *any* set $\mathbf{d}$ of edge weights, whenever $(G, \mathbf{d})$ is realizable at all, then it can also be realized in $\mathbb{R}^k$. As far as algorithmics are concerned, the characterizations by Belk and Connelly immediately yield efficient algorithms for realizing k–realizable graphs in $\mathbb{R}^k$ for the cases where $k = 1, 2$. However, finding a corresponding algorithm for 3–realizable graphs is posed as an open question in Belk and Connelly (2007).

In this chapter we show how the semidefinite programming approach of the previous chapter can be used to design an efficient algorithm that, given an 3–realizable graph G and a set $\mathbf{d}$ of weights on the edges, finds an approximate realization of $(G, \mathbf{d})$ in $\mathbb{R}^3$. Besides answering the question of Belk and Connelly, our approach reveals a little–explored connection between SDP and tensegrity theories, and provides an alternative proof technique for problems in the latter area. Before we proceed, let us review the definition and properties of k–realizable graphs, as well as the theorem of Belk and Connelly on the characterization of k–realizable graphs for the cases where $k = 1, 2, 3$.

4.2 Realizability of Graphs

We begin with the definition of an k–realizable graph.

Definition 4.2.1 *Let $k \geq 1$ be an integer, and let $G = (V, E)$ be a graph with n vertices. We say that G is k–*realizable *if any configuration $\mathbf{p} = (p_1, \ldots, p_n)$ of G in some finite–dimensional Euclidean space gives rise to a configuration $\mathbf{q} = (q_1, \ldots, q_n)$ in $\mathbb{R}^k$ such that $\|p_i - p_j\| = \|q_i - q_j\|$ for all $(i, j) \in E$.*

Equivalently, a graph G is k–realizable if the existence of a realization $\mathbf{p}$ of $(G, \mathbf{d})$ for some weight vector $\mathbf{d}$ implies the existence of a realization $\mathbf{q}$ of $(G, \mathbf{d})$ in $\mathbb{R}^k$.

Note that the above definition allows edges to have zero length. In particular, if a graph G is k–realizable and H is a minor of G, then H is also k–realizable. Hence, the property of being k–realizable is *minor–monotone*, i.e. it is closed under the operation of taking minors. By the celebrated Graph Minor Theorem of Robertson and Seymour (2004), it then follows that there exists a finite list of *forbidden minors* for k–realizability, i.e. there exists a finite list of graphs $G_1, \ldots, G_l$ such that G is k–realizable if and only if G does not have any G_i $(1 \leq i \leq l)$ as a minor. Thus, in principle, we can characterize the class of k–realizable graphs by listing all of its forbidden minors. This is indeed the approach taken by Belk and Connelly (2007), which we now briefly review.

To begin, we need the notions of *k–tree* and *partial k–tree*. We have already encountered k–trees in Section 3.4.2, where we used them to demonstrate the existence of sparse uniquely k–realizable instances.

Definition 4.2.2 *Let $k \geq 1$ be an integer. An* k–tree *is defined recursively as follows. The complete graph on k vertices is an k–tree. An k–tree on $n+1$ vertices (where $n \geq k$) is constructed from an k–tree G with n vertices by adding a vertex v to G and making v adjacent to all the vertices of one of the k–vertex complete subgraphs of G. A* partial k–tree *is a subgraph of an k–tree.*

It is clear that k–trees are k–realizable, and hence partial k–trees, being subgraphs of k–realizable graphs, are also k–realizable. Moreover, the property of being a partial k–tree is minor–monotone (see, e.g., Theorem 1.1 of Arnborg et al. (1990)). Thus, any forbidden minor for k–realizability must also be a forbidden minor for partial k–trees. The upshot of this observation is that the forbidden minors for partial k–trees are well–studied. In particular, the complete list of forbidden minors for partial k–trees has been found for the cases where $k = 1, 2, 3$, which allows Belk and Connelly (2007) to obtain the forbidden minor characterization of k–realizability for those cases.

4.2.1 1–Realizable and 2–Realizable Graphs: Characterization and Algorithmic Issues

For $k = 1$, it is easy to see that 1–trees are simply trees. Hence, partial 1–trees are equivalent to forests, and K_3 is the only minimal forbidden minor for partial 1–trees. Now, observe that K_3 is not 1–realizable, since by stipulating the edges to have length 1, the resulting instance can be realized in $\mathbb{R}^2$ but not in $\mathbb{R}$. In fact, a similar argument shows that for every $k \geq 1$, K_{k+2} is a minimal forbidden minor for k–realizability. Thus, a graph is 1–realizable if and only if it is a forest. Now, given a graph G, one can decide whether it is a forest in linear time (e.g., via depth–first search). Furthermore, it is clear that any forest with any specified edge weights can be realized in $\mathbb{R}$ using a linear number of arithmetic operations. Hence, we have the following theorem:

Theorem 4.2.3 *(cf. Belk and Connelly (2007)) A graph is* 1*–realizable if and only if it is a forest. Given a graph G, one can decide whether it is* 1*–realizable in linear time. Moreover,* *if G is 1–realizable, then given any weight vector $\mathbf{d}$ on the edges, an exact realization of $(G, \mathbf{d})$ in $\mathbb{R}$ can be found in linear time in the real RAM model.*

Now, consider the case where $k = 2$. It is known (Wald and Colbourn (1983)) that K_4 is the only minimal forbidden minor for partial 2–trees, and the remark in the preceding paragraph shows that it is also a forbidden minor for 2–realizability. Hence, we have the following theorem:

Theorem 4.2.4 *(Belk and Connelly (2007)) A graph is 2–realizable if and only if it does not have K_4 as a minor.*

Let us now address two algorithmic questions related to 2–realizability, namely, recognition and realization. We begin with a definition:

Definition 4.2.5 *The class of* series–parallel *graphs is defined recursively as follows:*

- *(Basis) A graph consisting of two vertices joined by a single edge is series–parallel. One vertex is designated as the* source*, and the other is designated as the* sink*.*
- *Given two series–parallel graphs G_1 and G_2 with source–sink pairs (s_1, t_1) and (s_2, t_2), the graphs constructed via the following operations are also series–parallel:*
  - *(Parallel Composition) Identify s_1 with s_2 and t_1 with t_2. The source and sink of the new graph are $s_1 = s_2$ and $t_1 = t_2$, respectively.*
  - *(Series Composition) Identify t_1 with s_2. The source and sink of the new graph are s_1 and t_2, respectively.*

It is well–known (see, e.g., Diestel (2005)) that the class of graphs without an K_4–minor is precisely the class of series–parallel graphs. Thus, the problem of recognizing 2–realizable graphs is equivalent to that of recognizing series–parallel graphs. The latter is a well–studied problem in algorithmic graph theory, and a linear time algorithm is available (see, e.g., Valdes et al. (1982)). Now, given an 2–realizable graph G with at least 2 vertices and a set $\mathbf{d}$ of weights on the edges, we can realize $(G, \mathbf{d})$ in $\mathbb{R}^2$ as follows. First, we use the polynomial time algorithm of Laurent (2000) to determine whether $(G, \mathbf{d})$ is realizable at all, and if so, find a realization in some finite–dimensional Euclidean space. As a by–product, we obtain a set $\mathbf{d}'$ of realizable distances between all pairs of vertices in G. Then, we use the linear time algorithm of Wald and Colbourn (1983) to complete G into an 2–tree G', where

G' has the same vertex set as G and includes all the edges of G, and obtain an ordering Π of the vertices from which G' is constructed. Finally, since $(G', \mathbf{d}'|_{E'})$ is realizable (where $\mathbf{d}'|_{E'}$ is the restriction of $\mathbf{d}'$ to the set of edges E' of G'), we can realize $(G', \mathbf{d}'|_{E'})$ in $\mathbb{R}^2$ by realizing each vertex of G' in the order given by Π. Clearly, this last step can be carried out in linear time.

To summarize, we have the following theorem:

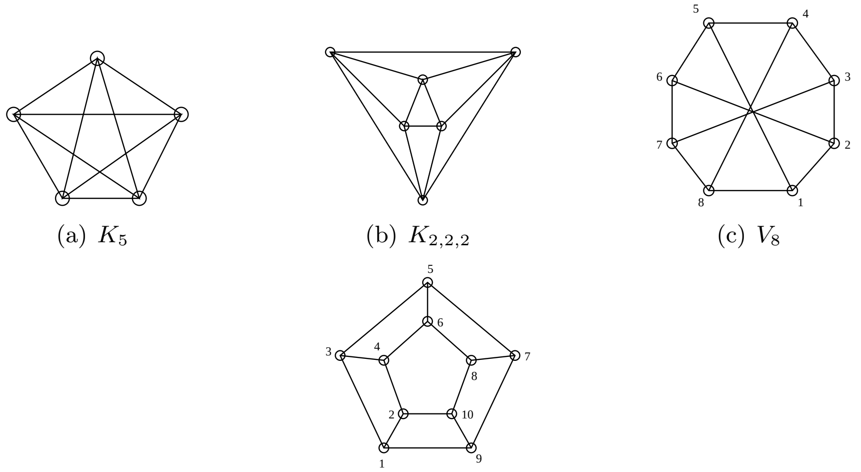
Theorem 4.2.6 *Given a graph G, one can decide whether it is 2–realizable in linear time. If G is 2–realizable, then given any weight vector $\mathbf{d}$ on the edges, one can decide whether it is realizable in polynomial time (in the real RAM model). Moreover, if $(G, \mathbf{d})$ is realizable, then an exact realization of $(G, \mathbf{d})$ in $\mathbb{R}^2$ can be found in polynomial time (again in the real RAM model).*

Before we proceed to discuss the characterization of 3–realizable graphs, let us remark that in Belk and Connelly (2007), it was mentioned that Jack Snoeyink has found a linear time algorithm for realizing 2–realizable graphs in $\mathbb{R}^2$. However, details of the purported algorithm were not given, nor are we able to find the reference for such a result.

4.2.2 3–Realizable Graphs: Characterization

As we saw above, the characterizations of 1–realizable and 2–realizable graphs are relatively straightforward, since there is only one forbidden minor in both cases. However, the situation is considerably more complicated for 3–realizable graphs. To begin our discussion, let us first recall a result of Arnborg et al. (1990), who proved that the graphs shown in Figure 4.1 are precisely the set of minimal forbidden minors for partial 3–trees.

As remarked earlier, K_5 is a forbidden minor for 3–realizability. It turns out that $K_{2,2,2}$ is also a forbidden minor for 3–realizability. Specifically, Belk and Connelly (2007) showed that there exists a realization of $K_{2,2,2}$ in $\mathbb{R}^4$ that cannot be realized in $\mathbb{R}^3$. On the other hand, Belk (2007) showed that both V_8 and $C_5 \times C_2$ are 3–realizable, and hence they are not forbidden minors for 3–realizability. The proof employs elements of tensegrity theory, and in particular, the notion of stress. The main idea is that the realization obtained by stretching apart a certain pair of non–adjacent vertices in $G \in \{V_8, C_5 \times C_2\}$ will be "flat" enough that it can be forced into $\mathbb{R}^3$. In order to make this precise, we need some definitions from the theory of tensegrities (see, e.g., Roth and Whiteley (1981); Connelly (1982)):



(a) K_5

(b) $K_{2,2,2}$

(c) V_8

(d) $C_5 \times C_2$

Figure 4.1: Forbidden Minors for Partial 3–Trees

Definition 4.2.7 *A* tensegrity $G(\mathbf{p})$ *is a graph* $G = (V, E)$ *together with a configuration* $\mathbf{p} = (p_1, \ldots, p_n) \in \mathbb{R}^{kn}$ *such that each edge is labelled as a cable, strut, or bar; each vertex is labelled as pinned or unpinned; and vertex* $i \in V$ *is assigned the coordinates* $p_i \in \mathbb{R}^k$ *for* $1 \leq i \leq n$.

The label on each edge is intended to indicate its functionality. Cables (resp. struts) are allowed to decrease (resp. increase) in length (or stay the same length), but not to increase (resp. decrease) in length. Bars are forced to remain the same length. As before, a pinned vertex is forced to remain where it is. Given a graph $G = (V, E)$ and a set $\mathbf{d}$ of weights on the edges, if (i, j) is a cable (resp. strut), then d_{ij} will be the upper (resp. lower) bound on its length. If (i, j) is a bar, then d_{ij} will simply be its length.

An important concept in the study of tensegrities is that of an *equilibrium stress*:

Definition 4.2.8 *An* equilibrium stress *for* $G(\mathbf{p})$ *is an assignment of real numbers* $\omega_{ij} = \omega_{ji}$ *to each edge* $(i, j) \in E$ *such that for each unpinned vertex* i *of* G*, we have:*

$$\sum_{j:(i,j)\in E} \omega_{ij}(p_i - p_j) = \mathbf{0} \tag{4.1}$$

Furthermore, we say that the equilibrium stress $\omega = \{\omega_{ij}\}$ *is* proper *if* $\omega_{ij} = \omega_{ji} \geq 0$ *(resp.* ≤ 0*) if* (i, j) *is a cable (resp. strut).*

Note that the notion of an equilibrium stress has both combinatorial and geometric components. Specifically, it depends on the graph G and on the configuration $\mathbf{p}$.

Clearly, the zero stress $\omega = \mathbf{0}$ is a proper equilibrium stress, but it is not too interesting. On the other hand, suppose that $G(\mathbf{p})$ has a non–zero equilibrium stress, and that at least one of the incident edges of vertex i has a non–zero stress. Then, equation (4.1) implies that the set of vectors $\{p_j - p_i : (i,j) \in E\}$ is linearly dependent, and hence those vectors span a lower dimensional space. Thus, it would be nice to have conditions that guarantee the existence of a non–zero proper equilibrium stress. It turns out that the concept of an *unyielding tensegrity* is useful for that purpose.

Definition 4.2.9 *Let $G = (V,E)$ be a graph, and let $\mathbf{p}$ and $\mathbf{q}$ be two configurations of G. We say that $G(\mathbf{p})$* dominates *$G(\mathbf{q})$ (denoted by $G(\mathbf{p}) \trianglerighteq G(\mathbf{q})$) if for every pinned vertex i, we have $p_i = q_i$, and for every edge $(i,j) \in E$, we have:*

$$\|p_i - p_j\| \left\{ \begin{array}{c} \geq \\ = \\ \leq \end{array} \right\} \|q_i - q_j\| \quad \textit{if } (i,j) \textit{ is a } \left\{ \begin{array}{c} \textit{cable} \\ \textit{bar} \\ \textit{strut} \end{array} \right\}$$

We call $G(\mathbf{p})$ an unyielding tensegrity *and $\mathbf{p}$ an* unyielding configuration *if any other configuration $\mathbf{q}$ with $G(\mathbf{p}) \trianglerighteq G(\mathbf{q})$ satisfies $\|p_i - p_j\| = \|q_i - q_j\|$ for all $(i,j) \in E$.*

We are now ready to state the following theorem due to Belk (2007), which plays a crucial role in the characterization of 3–realizable graphs:

Theorem 4.2.10 *(Belk (2007)) If $G(\mathbf{p})$ is an unyielding tensegrity with exactly one strut or cable, then $G(\mathbf{p})$ has an equilibrium stress that is non–zero on at least one edge.*

Belk's proof of Theorem 4.2.10 uses the Inverse Function Theorem and hence is not constructive. In the next section, we shall prove a generalized and qualitatively improved version of Theorem 4.2.10 using SDP duality theory. Before we do that, let us indicate how Theorem 4.2.10 can be used to establish the 3–realizability of V_8.

Let $\mathbf{p}$ be a configuration of V_8, let G be the graph obtained from V_8 by adding an edge between vertices 1 and 4 (see Figure 4.1(c)). Consider the tensegrity $G(\mathbf{p})$ where all the edges of V_8 are bars and the edge $(1,4)$ is a strut. It is not hard to show that there exists a configuration $\mathbf{q}$ of V_8 such that $G(\mathbf{q})$ is unyielding (see, e.g., Belk (2007)). Hence, by Theorem 4.2.10, the unyielding tensegrity $G(\mathbf{q})$ admits a non–zero stress. The idea now is to show that no matter which of the edges of G have non–zero stress, one can "fold" the realization $\mathbf{q}$ of V_8 into $\mathbb{R}^3$. The complete proof is via a tedious case analysis. To illustrate the idea, we will analyze a simple case and refer the reader to Belk (2007) for further details.

Suppose that all the edges that are incident to vertices $5, 6, 7, 8$ have zero stress. Then, the non–zero stress lies entirely on the edges $(1, 2), (2, 3), (3, 4), (1, 4)$. This implies that the vertices $1, 2, 3, 4$ are collinear. In particular, we may assume that the coordinates q_1, q_2, q_4, q_5, q_7 of vertices $1, 2, 4, 5, 7$ are in $\mathbb{R}^3$. Then, it can be readily verified that each of the remaining vertices can be rotated about the plane spanned by its neighbors into $\mathbb{R}^3$.

The proof of the 3–realizability of $C_5 \times C_2$ follows the same approach and is again extremely tedious. We refer the reader to Belk (2007) for the details.

To summarize, we have the following theorem:

Theorem 4.2.11 *(Belk and Connelly (2007); Belk (2007)) A graph is* 3*–realizable if and only if it does not have either* K_5 *or* $K_{2,2,2}$ *as a minor.*

We shall study the algorithmic questions related to 3–realizability in Section 4.2.5.

4.2.3 Proof of Theorem 4.2.10 Using Semidefinite Programming Duality

Although Theorem 4.2.10 shows that an unyielding tensegrity always has a non–zero equilibrium stress, it does not say how an unyielding configuration or a non–zero equilibrium stress can be computed. In particular, Belk's proof of Theorem 4.2.10 does not provide an algorithm for realizing 3–realizable graphs. We now show that the problem of computing an unyielding configuration $\mathbf{p}$ of a graph G can be formulated as an SDP. What is even more interesting is that the optimal dual multipliers of the SDP will give rise to a non–zero proper equilibrium stress for $G(\mathbf{p})$. Consequently, we are able to provide a constructive proof of Theorem 4.2.10. In fact, our proof yields more information than that offered by Belk's proof. Finally, as we shall see, Theorem 4.2.10 can be easily generalized to include the case of multiple struts and cables using our SDP formulation.

A Semidefinite Programming Formulation

To motivate our approach, let us observe that given a graph G and a set $\mathbf{d}$ of weights on the edges, the relation $\trianglerighteq$ induces a partial order on the set of tensegrities $G(\mathbf{p})$, where $\mathbf{p}$ is a configuration that satisfies the edge constraints. In particular, an unyielding tensegrity can be viewed as an extremal element in this partial order. This observation suggests that we may formulate the problem of finding an unyielding tensegrity as an optimization problem. Before we present the SDP formulation, let us recall some notation from Section 3.4 and introduce some new ones.

Let $G = (V, E)$ be a graph, where $V_1 = \{1, \ldots, n\}$ and $V_2 = \{n+1, \ldots, n+m\}$ form a partition of V. Let $\mathbf{a} = (a_i)_{i \in V_2}$ be the positions of the pinned vertices in $\mathbb{R}^k$, where $k \geq 1$ is a fixed integer. Let $E_1 = \{(i,j) \in E : i, j \in V_1\}$ and $E_2 = \{(i,j) \in E : i \in V_2,\, j \in V_1\}$, and let $\mathbf{d}$ (resp. $\bar{\mathbf{d}}$) be a set of weights on the edges in E_1 (resp. E_2).

Now, set $E_1^c = \{(i,j) \notin E : i, j \in V_1\}$ and $E_2^c = \{(i,j) \notin E : i \in V_2,\, j \in V_1\}$. Let C_1, S_1 be disjoint subsets of E_1^c, and let C_2, S_2 be disjoint subsets of E_2^c. The pairs in C_i are intended to be *cables*, and those in S_i are intended to be *struts*. We remark that we do not assume the sets C_1, C_2, S_1, S_2 to be non–empty.

Now, consider the following SDP, where we augment the formulation (3.9) with an objective function:

$$\begin{array}{lll} \sup & \displaystyle\sum_{(i,j)\in S_1} E_{ij} \bullet Z + \sum_{(i,j)\in S_2} \bar{E}_{ij} \bullet Z & \\ & \displaystyle - \sum_{(i,j)\in C_1} E_{ij} \bullet Z - \sum_{(i,j)\in C_2} \bar{E}_{ij} \bullet Z & \\ \text{subject to} & E_{ij} \bullet Z = d_{ij}^2 & \text{for } (i,j) \in E_1 \\ & \bar{E}_{ij} \bullet Z = \bar{d}_{ij}^2 & \text{for } (i,j) \in E_2 \\ & Z \succeq \mathbf{0}, \; Z_{1:k,1:k} = I_k & \end{array} \tag{4.2}$$

where, as we recall from Section 3.4, $Z_{1:k,1:k}$ is the $k \times k$ principal submatrix of Z indexed by the first k rows (columns),

$$E_{ij} = \begin{pmatrix} \mathbf{0} \\ e_i - e_j \end{pmatrix} \begin{pmatrix} \mathbf{0} \\ e_i - e_j \end{pmatrix}^T \quad \text{and} \quad \bar{E}_{ij} = \begin{pmatrix} a_i \\ -e_j \end{pmatrix} \begin{pmatrix} a_i \\ -e_j \end{pmatrix}^T$$

In essence, we are placing cables between pairs of vertices in $C_1 \cup C_2$, and struts between pairs of vertices in $S_1 \cup S_2$. The objective function is trying to push apart as much as possible pairs of vertices that are connected by struts, while at the same time trying to pull together as much as possible pairs of vertices that are connected by cables. Thus, we may regard problem (4.2) as that of finding a realization $\mathbf{p}$ such that the tensegrity $G(\mathbf{p})$ is unyielding (recall Definition 4.2.9). Note that $\mathbf{p}$ need not lie in $\mathbb{R}^k$.

It turns out that it is very instructive to consider the dual of (4.2), which is given by:

$$\begin{array}{ll} \inf & I_k \bullet V + \sum_{(i,j)\in E_1} \theta_{ij} d_{ij}^2 + \sum_{(i,j)\in E_2} w_{ij} \bar{d}_{ij}^2 \\ \text{subject to} & U \equiv -\sum_{(i,j)\in S_1} E_{ij} - \sum_{(i,j)\in S_2} \bar{E}_{ij} + \sum_{(i,j)\in C_1} E_{ij} + \sum_{(i,j)\in C_2} \bar{E}_{ij} \\ & \quad + \begin{bmatrix} V & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \sum_{(i,j)\in E_1} \theta_{ij} E_{ij} + \sum_{(i,j)\in E_2} w_{ij} \bar{E}_{ij} \succeq \mathbf{0} \end{array} \tag{4.3}$$

As we shall see, the optimal dual multipliers are closely related to an equilibrium stress for a certain realization of $(G, (\mathbf{d}, \bar{\mathbf{d}}), \bar{\mathbf{a}})$ in $\mathbb{R}^l$ for some $l \geq k$.

Analysis of the SDP Formulation

Our goal in this sub–section is to establish the following theorem, which first appeared in So and Ye (2006):

Theorem 4.2.12 *Let $G = (V, E)$, $\mathbf{d}$, $\bar{\mathbf{d}}$ and $\mathbf{a}$ be given such that:*

(1) there is at least one pinned vertex, and

(2) the graph $G\backslash\{n+2, \ldots, n+m\}$ is connected.

Consider the SDP (4.2), where we assume that:

(3) it is strictly feasible, and

(4) the objective function is not vacuous, i.e. at least one of the sets C_1, C_2, S_1, S_2 is non–empty.

Let $\bar{\mathbf{x}} = (\bar{x}_1, \ldots, \bar{x}_n) \in \mathbb{R}^{ln}$ be the positions of the unpinned vertices in $\mathbb{R}^l$ (for some $l \geq k$), obtained from the optimal primal matrix $\bar{Z}$ via (3.13), and let $\{\bar{\theta}_{ij}, \bar{w}_{ij}\}$ be the optimal dual multipliers. Suppose that we assign the stress $\bar{\theta}_{ij}$ (resp. $\bar{w}_{ij}$) to the bar $(i,j) \in E_1$ (resp. $(i,j) \in E_2$), a stress of 1 to all the cables in $C_1 \cup C_2$, and a stress of -1 to all the struts in $S_1 \cup S_2$. Then, the resulting assignment yields a non–zero proper equilibrium stress for the tensegrity $G'(\bar{\mathbf{x}}, \bar{\mathbf{a}})$, where $G' = (V, E \cup C_1 \cup C_2 \cup S_1 \cup S_2)$ and $\bar{\mathbf{a}} = (\bar{a}_{n+1}, \ldots, \bar{a}_{n+m})$ (recall that $\bar{a}_i \in \mathbb{R}^l$ is the augmentation of $a_i \in \mathbb{R}^k$ as in (3.11)).

Before we outline the proof of Theorem 4.2.12, several remarks are in order. The intuition behind the proof of Theorem 4.2.12 is simple. Suppose that (4.2) and (4.3) achieve the same optimal value, and that the common optimal value is attained by the primal matrix $\bar{Z}$ and the dual matrix $\bar{U}$. Then, the desired result should follow from one of the conditions for strong duality, namely the identity $\bar{Z}\bar{U} = \mathbf{0}$. Of course, strong duality for SDP does not necessarily hold, and even when it does, there is no guarantee that the optimal value is attained by any matrix. Thus, some additional technical assumptions are needed, and items (2) and (3) in the statement of Theorem 4.2.12 turn out to be sufficient for our purposes. In fact, the conclusion of Theorem 4.2.12 remains valid if we may replace (3) by the following:

(3') the optimal value of (4.3) is attained by some dual feasible matrix

We remark that in most applications of Theorem 4.2.12, there will only be one pinned vertex, namely $a_{n+1} = \mathbf{0}$. Thus, primal strict feasibility can be ensured if the given weights $\mathbf{d}$ admit a realization whose vertices are in general position, and the connectivity condition is simply the statement that G is connected. However, the strict feasibility assumption (or the dual attainment assumption) does weaken the applicability of Theorem 4.2.12. In particular, Theorem 4.2.12 is not as general as Theorem 4.2.10. We will address this issue later (see Theorem 4.2.16 below).

Besides strict feasibility, we have also assumed that the given instance has at least one pinned vertex. Such an assumption is necessary in order to ensure that the entries of $\bar{Z}$ are bounded, but we can no longer argue that the net stress exerted on a pinned vertex is zero. However, if there is only one pinned vertex in the given instance, then the net stress exerted on it will be zero; see Corollary 4.2.14 below. Thus, we may assume without loss of generality that the given instance has one pinned vertex.

Finally, observe that the assumptions in the statement of Theorem 4.2.12 buy us some additional information that is not offered by Theorem 4.2.10. Specifically, the equilibrium stress obtained in Theorem 4.2.12 is non–zero on all the cables and struts, and the magnitudes of the stress on all the cables and struts can be prescribed (by assigning appropriate weights to each summand in the primal objective function). To the best of our knowledge, this is the first time such a property is observed.

We now begin our analysis with the following proposition, which, by Theorem 2.3.6, implies that there is no gap between the optimal values of (4.2) and (4.3), and that the common optimal value is attained by some primal feasible matrix:

Proposition 4.2.13 *Suppose that $a_{n+1} = \mathbf{0}$, and that the graph $\tilde{G} \equiv G \backslash \{n+2, \ldots, n+m\}$ is connected. Then, the dual problem (4.3) is strictly feasible, i.e. there exists a dual feasible U with $U \succ \mathbf{0}$.*

Proof. It suffices to show that there exists a choice of V, θ_{ij} and w_{ij} such that:

$$U' \equiv \left[\begin{array}{cc} V & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{array} \right] + \sum_{(i,j)\in E_1} \theta_{ij} E_{ij} + \sum_{(i,j)\in E_2} w_{ij} \bar{E}_{ij} - \sum_{(i,j)\in S_1} E_{ij} - \sum_{(i,j)\in S_2} \bar{E}_{ij} \succ \mathbf{0}$$

Note that:

$$U' = \left[\begin{array}{cc} U'_{11} & U'_{12} \\ (U'_{12})^T & U'_{22} \end{array} \right]$$

where:

$$\begin{aligned} U'_{22} &= \sum_{(i,j)\in E_1} \theta_{ij}(e_i - e_j)(e_i - e_j)^T + \sum_{(i,j)\in E_2} w_{ij} e_j e_j^T \\ &\quad - \sum_{(i,j)\in S_1} (e_i - e_j)(e_i - e_j)^T - \sum_{(i,j)\in S_2} e_j e_j^T \end{aligned}$$

Let us choose $\theta_{ij} = \mu_{ij} > 0$ for all $(i,j) \in E_1$, $w_{n+1,s} = \nu_s > 0$ for all $s \in V_1$ such that $(n+1, s) \in E_2$, and $w_{ij} = 0$ for all other $(i,j) \in E_2$. We claim that:

$$\begin{aligned} M &\equiv \sum_{(i,j)\in E_1} \theta_{ij}(e_i - e_j)(e_i - e_j)^T + \sum_{(i,j)\in E_2} w_{ij} e_j e_j^T \\ &= \sum_{(i,j)\in E_1} \mu_{ij}(e_i - e_j)(e_i - e_j)^T + \sum_{s:(n+1,s)\in E_2} \nu_s e_s e_s^T \succ \mathbf{0} \end{aligned} \tag{4.4}$$

It is clear that $M \succeq \mathbf{0}$. Now, let $u \in \mathbb{R}^n$ be such that $u^T M u = 0$. By definition, we have:

$$\begin{aligned} u^T M u &= \sum_{(i,j)\in E_1} \mu_{ij} u^T \left((e_i - e_j)(e_i - e_j)^T \right) u + \sum_{s:(n+1,s)\in E_2} \nu_s u^T \left(e_s e_s^T \right) u \\ &= \sum_{(i,j)\in E_1} \mu_{ij} (u_i - u_j)^2 + \sum_{s:(n+1,s)\in E_2} \nu_s u_s^2 \end{aligned}$$

Thus, we have $u^T M u = 0$ if and only if $u_s = 0$ for all s such that $(n+1, s) \in E_2$, and $u_i = u_j$ for all $(i,j) \in E_1$. We now show that these conditions imply that $u = \mathbf{0}$. Let us fix an $s \in V_1$ such that $(n+1, s) \in E_2$. In particular, we have $u_s = 0$. Since $\tilde{G}$ is connected, there exists a path between the unpinned vertex s and any other unpinned vertex j in $\tilde{G}$ using only the vertices $\{1, 2, \ldots, n, n+1\}$. Let $\mathscr{P} = \{s = v_{i_0}, v_{i_1}, \ldots, v_{i_l} = j\}$ be one such path. We claim that $u_j = 0$. Indeed, consider the vertex v_{i_1}. We either have $v_{i_1} = n+1$, or v_{i_1} is an unpinned vertex, say $v_{i_1} = j'$ for some $j' \in \{1, \ldots, n\}$. If it is the former, then v_{i_2} must be an unpinned vertex (say $v_{i_2} = j''$ for some $j'' \in \{1, \ldots, n\}$). This implies that $(n+1, j'') \in E_2$, whence $u_{j''} = 0$. If it is the latter, then $(v_{i_0}, v_{i_1}) \in E_1$, and hence $u_{j'} = u_s = 0$. By repeating this argument, we see that $u_j = 0$, as desired. Since $j \in \{1, \ldots, n\}$ is arbitrary, we have $u = \mathbf{0}$, thus establishing (4.4).

Now, let $u \in \mathbb{R}^n \backslash \{\mathbf{0}\}$. Observe that:

$$\begin{aligned} u^T \left(\sum_{(i,j) \in S_1} (e_i - e_j)(e_i - e_j)^T \right) u &= \sum_{(i,j) \in S_1} (u_i - u_j)^2 \\ &\le 2 \sum_{(i,j) \in S_1} \left(u_i^2 + u_j^2 \right) \\ &\le 4n \|u\|^2 \end{aligned} \tag{4.5}$$

and

$$u^T \left(\sum_{(i,j) \in S_2} e_j e_j^T \right) u = \sum_{(i,j) \in S_2} u_j^2 \le m \|u\|^2 \tag{4.6}$$

Let $\bar{v} = \min_{\|u\|=1} u^T M u$. Clearly, we have $\bar{v} > 0$. Let $\alpha > 0$ be such that $\alpha \bar{v} > 4n + m$. Then, it follows from (4.5) and (4.6) that for $u \in \mathbb{R}^n \backslash \{\mathbf{0}\}$ with $\|u\| = 1$, we have:

$$u^T \left(\alpha M - \sum_{(i,j) \in S_1} (e_i - e_j)(e_i - e_j)^T - \sum_{(i,j) \in S_2} e_j e_j^T \right) u \ge \alpha \bar{v} - (4n + m) > 0$$

In particular, we see that by taking $\theta_{ij} = \alpha \mu_{ij}$ and $w_{n+1,s} = \alpha \nu_s$, we have $U'_{22} \succ \mathbf{0}$.

To complete the proof of Proposition 4.2.13, let $V = I_k + \sum_{(i,j) \in S_2} a_i a_i^T$. Since $a_{n+1} = \mathbf{0}$ by assumption, we have $U'_{11} = I_k$, and $U'_{12} = -\sum_{(i,j) \in S_2} a_i e_j^T$. By Theorem 2.2.5, we have $U' \succ \mathbf{0}$ if and only if $U'_{22} - (U'_{12})^T U'_{12} \succ \mathbf{0}$. Now, note that $(U'_{12})^T U'_{12}$ does not depend on θ_{ij} and w_{ij}. Let $\bar{v}' = \max_{\|u\|=1} u^T (U'_{12})^T U'_{12} u$. Since $\{u : \|u\| = 1\}$ is compact, we see that

$\bar{v}' < \infty$. Thus, by taking a sufficiently large α (i.e. $\alpha\bar{v} - 4(n+m) - \bar{v}' > 0$), we see that:

$$u^T U_{22}' u - u^T (U_{12}')^T U_{12}' u \geq \alpha\bar{v} - 4(n+m) - \bar{v}' > 0$$

for all $u \in \mathbb{R}^n \backslash \{\mathbf{0}\}$ with $\|u\| = 1$. In particular, we have $U_{22}' - (U_{12}')^T U_{12}' \succ \mathbf{0}$, as desired. ■

We are now ready to prove Theorem 4.2.12:

Proof of Theorem 4.2.12. By Proposition 4.2.13 and our assumptions, both (4.2) and (4.3) are strictly feasible. Thus, it follows from Theorem 2.3.6 that there is no gap between the optimal values of (4.2) and (4.3), and that the common optimal value is attained in both problems. Let $\bar{Z}$ (resp. $\bar{U}$) be the optimal primal (resp. dual) matrix. Then, the absence of a duality gap implies that $\bar{Z}\bar{U} = \mathbf{0}$. Since $\bar{Z}$ takes the form:

$$\bar{Z} = \begin{bmatrix} I_k & X \\ X^T & Y \end{bmatrix}$$

we obtain:

$$\begin{aligned} \bar{Z}\bar{U} &= \begin{bmatrix} V & \mathbf{0} \\ X^T V & \mathbf{0} \end{bmatrix} + \sum_{(i,j)\in E_1} \bar{\theta}_{ij} \begin{bmatrix} \mathbf{0} & X_{ij} \\ \mathbf{0} & Y_{ij} \end{bmatrix} + \sum_{(i,j)\in E_2} \bar{w}_{ij} \begin{bmatrix} * & X_{ij}' \\ * & Y_{ij}' \end{bmatrix} \\ &\quad - \sum_{(i,j)\in S_1} \begin{bmatrix} \mathbf{0} & X_{ij} \\ \mathbf{0} & Y_{ij} \end{bmatrix} - \sum_{(i,j)\in S_2} \begin{bmatrix} * & X_{ij}' \\ * & Y_{ij}' \end{bmatrix} \\ &\quad + \sum_{(i,j)\in C_1} \begin{bmatrix} \mathbf{0} & X_{ij} \\ \mathbf{0} & Y_{ij} \end{bmatrix} + \sum_{(i,j)\in C_2} \begin{bmatrix} * & X_{ij}' \\ * & Y_{ij}' \end{bmatrix} \\ &= \mathbf{0} \end{aligned} \tag{4.7}$$

where:

(1) X_{ij} is an $k \times n$ matrix with $\bar{x}_i - \bar{x}_j$ on the i–th column and $-(\bar{x}_i - \bar{x}_j)$ on the j–th column;

(2) Y_{ij} is an $n \times n$ matrix with $(\bar{y}_{1i} - \bar{y}_{1j}, \ldots, \bar{y}_{ni} - \bar{y}_{nj})$ on the i–the column and $-(\bar{y}_{1i} - \bar{y}_{1j}, \ldots, \bar{y}_{ni} - \bar{y}_{nj})$ on the j–th column;

(3) X'_{ij} is an $k \times n$ matrix with $\bar{x}_j - a_i$ on the j–th column;

(4) Y'_{ij} is an $n \times n$ matrix with $(\bar{y}_{1j} - \bar{x}_1^T a_i, \ldots, \bar{y}_{nj} - \bar{x}_n^T a_i)$ on the j–th column.

Now, equation (4.7) implies that:

$$\mathbf{0} = \sum_{(i,j) \in E_1} \bar{\theta}_{ij} X_{ij} + \sum_{(i,j) \in E_2} \bar{w}_{ij} X'_{ij} - \sum_{(i,j) \in S_1} X_{ij} - \sum_{(i,j) \in S_2} X'_{ij} + \sum_{(i,j) \in C_1} X_{ij} + \sum_{(i,j) \in C_2} X'_{ij}$$

Consider a fixed column s of the matrix on the right–hand side. We have:

$$\begin{aligned} & \sum_{i:(i,s) \in E_1} \bar{\theta}_{is} (\bar{x}_s - \bar{x}_i) + \sum_{i:(i,s) \in E_2} \bar{w}_{is} (\bar{x}_s - a_i) \\ - & \sum_{i:(i,s) \in S_1} (\bar{x}_s - \bar{x}_i) - \sum_{i:(i,s) \in S_2} (\bar{x}_s - a_i) \\ + & \sum_{i:(i,s) \in C_1} (\bar{x}_s - \bar{x}_i) + \sum_{i:(i,s) \in C_2} (\bar{x}_s - a_i) = \mathbf{0} \end{aligned} \tag{4.8}$$

Thus, if $\text{rank}(\bar{Z}) = k$, then we have just shown that the assignment of scalars given in the statement of Theorem 4.2.12 yields a non–zero proper equilibrium stress for the realization $\{\bar{x}_1, \ldots, \bar{x}_n, a_{n+1}, \ldots, a_{n+m}\}$. Note, however, that there are no restrictions placed on the pinned vertices $a_{n+1}, \ldots, a_{n+m}$.

Now, suppose that $\text{rank}(\bar{Z}) > k$. Then, following the arguments in Section 3.4.2, we can extract from $\bar{Z}$ vectors:

$$\bar{x}_i = \begin{pmatrix} \bar{x}_i^1 \\ \bar{x}_i^2 \end{pmatrix} \in \mathbb{R}^l \qquad \text{for } 1 \le i \le n$$

such that $(\bar{x}_1, \ldots, \bar{x}_n)$ is a realization of $(G, (\mathbf{d}, \bar{\mathbf{d}}), \bar{\mathbf{a}})$ in $\mathbb{R}^l$, where $l > k$. Using equation (4.7) again, we have:

$$\mathbf{0} = \sum_{(i,j) \in E_1} \bar{\theta}_{ij} Y_{ij} + \sum_{(i,j) \in E_2} \bar{w}_{ij} Y'_{ij} - \sum_{(i,j) \in S_1} Y_{ij} - \sum_{(i,j) \in S_2} Y'_{ij} + \sum_{(i,j) \in C_1} Y_{ij} + \sum_{(i,j) \in C_2} Y'_{ij}$$

Consider a fixed column s of the matrix on the right–hand side. Using the identities:

$$\begin{aligned}
\bar{y}_{ts} - \bar{y}_{ti} &= (\bar{x}_t^1)^T \bar{x}_s^1 + \left(\bar{x}_t^2\right)^T \bar{x}_s^2 - \left((\bar{x}_t^1)^T \bar{x}_i^1 + \left(\bar{x}_t^2\right)^T \bar{x}_i^2\right) \\
&= (\bar{x}_t^1)^T (\bar{x}_s^1 - \bar{x}_i^1) + \left(\bar{x}_t^2\right)^T \left(\bar{x}_s^2 - \bar{x}_i^2\right) \\
\bar{y}_{ts} - (\bar{x}_t^1)^T a_i &= (\bar{x}_t^1)^T (\bar{x}_s^1 - a_i) + \left(\bar{x}_t^2\right)^T \bar{x}_s^2
\end{aligned}$$

we conclude that for each $t = 1, 2, \ldots, n$, we have:

$$\begin{aligned}
\left(\bar{x}_t^2\right)^T \Bigg\{ &\sum_{i:(i,s)\in E_1} \bar{\theta}_{is} \left(\bar{x}_s^2 - \bar{x}_i^2\right) + \sum_{i:(i,s)\in E_2} \bar{w}_{is} \bar{x}_s^2 \\
&- \sum_{i:(i,s)\in S_1} \left(\bar{x}_s^2 - \bar{x}_i^2\right) - \sum_{i:(i,s)\in S_2} \bar{x}_s^2 \\
&+ \sum_{i:(i,s)\in C_1} \left(\bar{x}_s^2 - \bar{x}_i^2\right) + \sum_{i:(i,s)\in C_2} \bar{x}_s^2 \Bigg\} = 0
\end{aligned} \tag{4.9}$$

In other words, the bracketed term is perpendicular to $\bar{x}_t^2$ for all $t = 1, 2, \ldots, n$. Thus, we conclude that the bracketed term is zero. In particular, the assignment of scalars given in the statement of Theorem 4.2.12 yields a non–zero proper equilibrium stress for the tensegrity $G'(\mathbf{x}, \bar{\mathbf{a}})$. This completes the proof. ■

Corollary 4.2.14 *Let $\bar{x}_1, \ldots, \bar{x}_n \in \mathbb{R}^l$ and $\{\bar{\theta}_{ij}, \bar{w}_{ij}\}$ be as in Theorem 4.2.12. Define:*

$$\omega(\bar{a}_i) \equiv \sum_{s:(i,s)\in E_2} \bar{w}_{is}(\bar{a}_i - \bar{x}_s) - \sum_{s:(i,s)\in S_2} (\bar{a}_i - \bar{x}_s) + \sum_{s:(i,s)\in C_2} (\bar{a}_i - \bar{x}_s)$$

to be the net stress exerted on the pinned vertex $i \in V_2$. Then, we have $\sum_{i=1}^m \omega(\bar{a}_{n+i}) = \mathbf{0}$.

Proof. The desired result follows upon summing (4.8) and the bracketed term in (4.9) over $s = 1, \ldots, n$ and then interchanging the order of summation. ■

In particular, we see that if there is only one pinned vertex, then the net stress exerted on it is zero at the solution $(\bar{x}_1, \ldots, \bar{x}_n, \bar{\theta}_{ij}; \bar{w}_{ij})$.

A Pathological Case for the SDP

In the discussion of Theorem 4.2.12, we have assumed that problem (4.2) is strictly feasible. It is interesting to see what could go wrong if that assumption is not satisfied. Consider the graph in Figure 4.2.

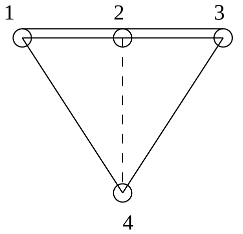


Figure 4.2: A Pathological Case for the SDP

Here, the vertices 1, 2 and 3 are collinear, and the edges $(1,2)$, $(2,3)$ and $(1,3)$ are all present. We would like to maximize the length of the strut $(2,4)$. Clearly, such a configuration can span at most two dimensions, and hence the corresponding primal SDP (4.2) is not strictly feasible. Suppose now that we pin vertex 1 at the origin and try to solve the SDP. As argued before, the algorithm will assign a stress of -1 on the strut $(2,4)$. However, the stress around vertex 2 will not be at equilibrium. The issue here is that when (4.2) is not strictly feasible, the optimal value may not be attained by any dual matrix U, even though there is a sequence of dual feasible matrices $\{U_n\}$ whose associated objective values tend to the optimal value.

Refinement of Theorem 4.2.12

Recall that our original goal is to give an alternative proof of Theorem 4.2.10 using semidefinite programming. We have almost achieved that goal in Theorem 4.2.12, except that the statement of the theorem requires a strict feasibility assumption. As we have just seen, Theorem 4.2.12 is not true without such an assumption. Thus, we have not yet recovered Theorem 4.2.10 entirely. Our goal now is to remedy this situation and complete the proof of Theorem 4.2.10.

To begin, suppose that problem (4.2) is feasible, $a_{n+1} = \mathbf{0}$, and that the graph $G\backslash\{n+2,\ldots,n+m\}$ is connected. Then, by Proposition 4.2.13, the common optimal value $\bar{v} < \infty$ of (4.2) and (4.3) is attained by some primal feasible matrix $\bar{Z}$. If $\bar{v}$ is also attained by some dual feasible matrix $\bar{U}$, then the conclusion of Theorem 4.2.12 still holds, since the proof of Theorem 4.2.12 only requires the existence of an optimal dual matrix. On the other hand, the following proposition shows that even if $\bar{v}$ is not attained by any dual feasible matrix, we can still extract some useful information:

Proposition 4.2.15 *Let $\bar{v} < \infty$ be as above, and suppose that it is not attained by any matrix U that is feasible for (4.3). Then, there exist $\tilde{V}$, $(\tilde{\theta}_{ij})_{(i,j)\in E_1}$ and $(\tilde{w}_{ij})_{(i,j)\in E_2}$,* not all of which are zero*, such that:*

$$I_k \bullet \tilde{V} + \sum_{(i,j)\in E_1} \tilde{\theta}_{ij} d_{ij}^2 + \sum_{(i,j)\in E_2} \tilde{w}_{ij} \bar{d}_{ij}^2 = 0 \tag{4.10}$$

and

$$\tilde{U} \equiv \begin{bmatrix} \tilde{V} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \sum_{(i,j)\in E_1} \tilde{\theta}_{ij} E_{ij} + \sum_{(i,j)\in E_2} \tilde{w}_{ij} \bar{E}_{ij} \succeq \mathbf{0} \tag{4.11}$$

In particular, not all of the multipliers $\{\tilde{\theta}_{ij}, \tilde{w}_{ij}\}$ can be zero.

Proof. By assumption, there exists a sequence $\{q(t) \equiv (V(t), \theta_{ij}(t), w_{ij}(t))\}_{t=1,2,\ldots}$ such that:

$$\left(I_k \bullet V(t) + \sum_{(i,j)\in E_1} \theta_{ij}(t) d_{ij}^2 + \sum_{(i,j)\in E_2} w_{ij}(t) \bar{d}_{ij}^2 \right) \to \bar{v} \qquad \text{as } t \to \infty$$

and

$$\begin{aligned} & - \sum_{(i,j)\in S_1} E_{ij} - \sum_{(i,j)\in S_2} \bar{E}_{ij} + \sum_{(i,j)\in C_1} E_{ij} + \sum_{(i,j)\in C_2} \bar{E}_{ij} \\ & + \begin{bmatrix} V(t) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \sum_{(i,j)\in E_1} \theta_{ij}(t) E_{ij} + \sum_{(i,j)\in E_2} w_{ij}(t) \bar{E}_{ij} \succeq \mathbf{0} \end{aligned}$$

for all $t \geq 1$. Note that we must have $\|q(t)\| \to \infty$ as $t \to \infty$, for otherwise the infimum in (4.3) will be attained by any limit point of the set $\{q(t) : t \geq 1\}$. Now, consider the sequence $\tilde{q}(t) = q(t)/\|q(t)\|$ of points on the unit sphere. By the Bolzano–Weierstrass theorem, this sequence has a convergent subsequence, and the limit point $\tilde{q} = (\tilde{V}, \tilde{\theta}_{ij}, \tilde{w}_{ij})$ satisfies (4.10) and (4.11), since $\bar{v}$ is finite.

Now, suppose that all the multipliers $\{\tilde{\theta}_{ij}, \tilde{w}_{ij}\}$ are zero. Then, (4.11) implies that the diagonal entries of $\tilde{V}$ must be non–negative, and (4.10) implies that $\tilde{V} = \mathbf{0}$, which is a contradiction. Thus, not all of the multipliers $\{\tilde{\theta}_{ij}, \tilde{w}_{ij}\}$ can be zero. ■

Now, let $\bar{Z}$ be optimal for the primal (4.2), and suppose that $\bar{v}$ is not attained by any matrix that is feasible for the dual (4.3). Then, by Proposition 4.2.15, there exists a non–zero vector $(\tilde{V}, \tilde{\theta}_{ij}, \tilde{w}_{ij})$ that satisfies (4.10) and (4.11). Consider the following primal–dual pair of SDPs:

$$(\mathrm{P}): \quad \begin{array}{lll} \sup & 0 & \\ \text{subject to} & E_{ij} \bullet Z = d_{ij}^2 & \text{for } (i,j) \in E_1 \\ & \bar{E}_{ij} \bullet Z = \bar{d}_{ij}^2 & \text{for } (i,j) \in E_2 \\ & Z \succeq \mathbf{0}, \;\; Z_{1:k,1:k} = I_k & \end{array}$$

and

$$(\mathrm{D}): \quad \begin{array}{ll} \inf & I_k \bullet V + \displaystyle\sum_{(i,j)\in E_1} \theta_{ij} d_{ij}^2 + \sum_{(i,j)\in E_2} w_{ij} \bar{d}_{ij}^2 \\ \text{subject to} & U \equiv \begin{bmatrix} V & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \displaystyle\sum_{(i,j)\in E_1} \theta_{ij} E_{ij} + \sum_{(i,j)\in E_2} w_{ij} \bar{E}_{ij} \succeq \mathbf{0} \\ & \theta_{ij} \in \mathbb{R} \text{ for all } (i,j) \in E_1; \quad w_{ij} \in \mathbb{R} \text{ for all } (i,j) \in E_2 \end{array}$$

Observe that $\bar{Z}$ (resp. $\tilde{U}$) is feasible for (P) (resp. (D)), and that they have the same objective value. Thus, by Theorem 2.3.6, they are optimal for their respective problems, and $\bar{Z}\tilde{U} = \mathbf{0}$. Note that the above argument would still work if $\bar{Z}$ is replaced by any matrix $\tilde{Z}$ that is feasible for (P). In particular, we would still have $\tilde{Z}\tilde{U} = \mathbf{0}$ for such an $\tilde{Z}$. Now, the arguments in the proof of Theorem 4.2.12 give the following theorem:

Theorem 4.2.16 *Consider the setting of Theorem 4.2.12, and suppose that assumptions (i), (ii) and (iv) hold. Suppose that the common optimal value of (4.2) and (4.3) is not attained by any dual feasible matrix. Let $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots, \tilde{x}_n) \in \mathbb{R}^{ln}$ be the positions of the unpinned vertices in $\mathbb{R}^l$ (for some $l \geq k$), obtained from some primal feasible matrix $\tilde{Z}$ via (3.13), and let $\{\tilde{\theta}_{ij}, \tilde{w}_{ij}\}$ be the dual multipliers given by Proposition 4.2.15. Suppose that we assign the stress $\tilde{\theta}_{ij}$ (resp. $\tilde{w}_{ij}$) to the bar $(i,j) \in E_1$ (resp. $(i,j) \in E_2$) and a stress of* 0 *to all the cables and struts. Then, the resulting assignment yields a non–zero proper equilibrium stress for the tensegrity $G'(\tilde{\mathbf{x}}, \bar{\mathbf{a}})$, where G' and $\bar{\mathbf{a}}$ are as in Theorem 4.2.12.*

It is curious to note that if an instance $(G, (\mathbf{d}, \bar{\mathbf{d}}), \mathbf{a})$ satisfies the hypotheses of Theorem 4.2.16, then the set of stress defined in Theorem 4.2.16 is a non–zero proper equilibrium stress for *any* realization of $(G, (\mathbf{d}, \bar{\mathbf{d}}), \bar{\mathbf{a}})$.

Together with Theorem 4.2.12, this completes the proof of Theorem 4.2.10.

4.2.4 3–Realizable Graphs: Structural Properties

Having characterized the class of 3–realizable graphs, a natural next step would be to develop algorithms for recognizing them and for realizing them in $\mathbb{R}^3$. However, this is not as straightforward as in the case of 1–realizable or 2–realizable graphs, since not every forbidden minor for partial 3–trees is a forbidden minor for 3–realizability. Thus, in order to obtain efficient algorithms for the recognition and realization problems, we need to examine the structure of 3–realizable graphs more closely.

We begin with some definitions. Let H be a graph. The graph G obtained by replacing the edges of H with independent paths between their ends (so that none of these paths has an inner vertex on another path or in H) is called a *subdivision* of H. We say that G *contains a subdivision* of H if there is a subgraph of G that is isomorphic to a subdivision of H. The following result provides a starting point for establishing various structural properties of 3–realizable graphs. For a proof, see, e.g., Proposition 1.7.2 of Diestel (2005).

Proposition 4.2.17 *Let H be a graph of maximum degree 3. Then, G has H as a minor if and only if G contains a subdivision of H.*

It is straightforward to verify that V_8 and $C_5 \times C_2$ satisfy the hypothesis of Theorem 4.2.17. Thus, if an 3–realizable graph G contains $H \in \{V_8, C_5 \times C_2\}$ as a minor, then it contains a subdivision of H. The following result indicates how this subdivision is related to other parts of the graph G:

Theorem 4.2.18 *(Connelly and Sloughter (2004)) Let G be an 3–realizable graph. Suppose that G contains a subdivision of H, where $H \in \{V_8, C_5 \times C_2\}$. Remove the subdivision of H from G, and consider the components of the resulting graph. Then, each component is connected in G to exactly one of the subdivided edges of H.*

Although Theorem 4.2.18 does not immediately yield an algorithm for recognizing 3–realizable graphs, it does suggest a divide–and–conquer strategy for realizing an 3–realizable graph G in $\mathbb{R}^3$. Indeed, let $\mathbf{d}$ be a set of weights on the edges, and suppose that $\mathbf{p}$ is a realization of $(G, \mathbf{d})$ in some finite–dimensional Euclidean space. Then, we can realize $(G, \mathbf{d})$

in $\mathbb{R}^3$ as follows. First, we find a subdivision H' of $H \in \{V_8, C_5 \times C_2\}$ in G, and remove it from G. (If G does not contain a subdivision of either V_8 or $C_5 \times C_2$, then it is a partial 3–tree, which can be realized in $\mathbb{R}^3$ directly; see Section 4.2.5.) Then, we realize each of the resulting components in $\mathbb{R}^3$. Finally, we realize H' in $\mathbb{R}^3$ and stitch everything back together to obtain a realization of $(G, \mathbf{d})$ in $\mathbb{R}^3$.

Of course, in order to implement the above strategy, we would still need an algorithm for finding a subdivision of $H \in \{V_8, C_5 \times C_2\}$ in G. However, it turns out that by analyzing the structure of an 3–realizable graph more carefully, we can obtain a simpler and more efficient divide–and–conquer algorithm for the realization problem. Moreover, our result immediately yields an algorithm for the recognition problem, as well as an algorithm for finding a subdivision of $H \in \{V_8, C_5 \times C_2\}$ in G. Before we state our result, we need some more definitions. Let $G = (V, E)$ be a connected graph. We say that G is *2–connected* if there does not exist a vertex u such that $G \backslash \{u\}$ has strictly more components than G. If such a vertex exists, then it is called an *articulation point*. A pair $\{u, v\}$ of distinct vertices in an 2–connected graph G is called a *separation pair* of G if there exist two subgraphs G'_1 and G'_2 such that:

(1) $V(G) = V(G'_1) \cup V(G'_2)$ and $V(G'_1) \cap V(G'_2) = \{u, v\}$;

(2) $E(G) = E(G'_1) \cup E(G'_2)$, $E(G'_1) \cap E(G'_2) = \emptyset$ and $|E(G'_1)|, |E(G'_2)| \geq 2$;

(3) for some $e_1 \in E(G'_1)$ and $e_2 \in E(G'_2)$, there is a cycle of G containing e_1 and e_2.

For $i = 1, 2$, we define G_i to be the graph obtained from G'_i by adding a new edge $e = (u, v)$. The graphs G_1 and G_2 are called *split graphs* of G with respect to $\{u, v\}$, and the edge $e = (u, v)$ is called a *virtual edge*. The process of decomposing a graph into two split graphs is called *splitting*.

Suppose now that we repeatedly split the graph G until no more splits are possible. The graphs constructed this way are called the *split components* of G. Note that each split component is 3–connected, but the split components of G are not necessarily unique. In order to get a unique set of 3–connected components of G, we must partially reassemble the split components. Let G_1 and G_2 be two split graphs with a common virtual edge $e = (u, v)$. We call the graph G defined by:

$$V(G) = V(G_1) \cup V(G_2) \quad \text{and} \quad E(G) = (E(G_1) \cup E(G_2)) \backslash \{e\}$$

the *merged graph* of G_1 and G_2, obtained by *merging* the virtual edge e. Note that merging is the inverse of splitting. Now, observe that the split components of G are of three types:

(1) triple bonds of the form $(\{u,v\},(u,v),(u,v),(u,v))$

(2) triangles of the form $(\{u,v,w\},(u,v),(v,w),(w,u))$

(3) 3–connected graphs

Let $\mathcal{B}_3$ denote the set of triple bonds, $\mathcal{T}$ denote the set of triangles, and $\mathcal{G}$ denote the set of 3–connected graphs in the split components of G. Suppose that the triple bonds $\mathcal{B}_3$ are merged as much as possible to give a set $\mathcal{B}$ of bonds, and the set $\mathcal{T}$ of triangles are merged as much as possible to give a set $\mathcal{P}$ of polygons. Then, the set of graphs $\mathcal{B} \cup \mathcal{P} \cup \mathcal{G}$ is the set of 3*–connected components* of G.

Our interest in the notion of 3–connected components stems from the following result:

Theorem 4.2.19 *(Asano (1985)) Let H be an* 3*–connected graph. Then, a graph G contains a subdivision of H if and only if there is an* 3*–connected component of G that contains a subdivision of H.*

Upon combining Theorems 4.2.18 and 4.2.19, we obtain the following structural result concerning 3–realizable graphs. In particular, it implies that we only need to focus on V_8 and $C_5 \times C_2$ (instead of their subdivisions) when designing algorithms for 3–realizable graphs.

Theorem 4.2.20 *Let G be an* 3*–realizable graph containing $H \in \{V_8, C_5 \times C_2\}$ as a minor. Then, any* 3*–connected component of G that contains H as a minor is isomorphic to H.*

Proof. Let $H \in \{V_8, C_5 \times C_2\}$. Since H is 3–connected, if G has contains a subdivision of H, then by Theorem 4.2.19, one of the 3–connected components of G will contain a subdivision of H. Let G' be one such component. We claim that G' is isomorphic to H. Suppose that this is not the case. Let us remove the subdivision of H from G', and let G'' be one of the resulting components. (We may assume without loss that G'' is non–null, for a subdivision of H is either isomorphic to H or is not 3–connected.) Since G' is 3–realizable (it is a minor of G), by Theorem 4.2.18, we see that G'' is connected in G' to only one of the subdivided edges of H. However, if we remove the two end vertices of that subdivided edge in G', then we will disconnect G'' from the rest of G', which contradicts the fact that G' is 3–connected. This establishes the claim. ■

In the next section, we will see how Theorem 4.2.20 can be used to solve various algorithmic problems on 3–realizable graphs.

4.2.5 3–Realizable Graphs: Algorithmic Issues

Recognizing 3–Realizable Graphs

Let us begin by considering the recognition problem of 3–realizable graphs: given a graph G, determine whether it is 3–realizable or not. The algorithm proceeds as follows. First, we find the set $\mathcal{S}$ of 3–connected components of G using the linear time algorithm of Hopcroft and Tarjan (1973). Now, note that if each of the 3–connected components of G is 3–realizable, then so is G. Thus, if G is not 3–realizable, then one of the 3–connected components of G must contain either K_5 or $K_{2,2,2}$ as a minor. In particular, we can first run the linear time algorithm of Matoušek and Thomas (1991) on each of the 3–connected components $H \in \mathcal{S}$ to determine whether H is a partial 3–tree. If H is a partial 3–tree, then it is 3–realizable. Otherwise, H will contain one of K_5, $K_{2,2,2}$, V_8 or $C_5 \times C_2$ as a minor. Now, we check whether H is isomorphic to either V_8 or $C_5 \times C_2$. If H is isomorphic to either V_8 or $C_5 \times C_2$, then it is 3–realizable. Otherwise, by Theorem 4.2.20, we may declare that G is not 3–realizable. If every 3–connected component of G is 3–realizable, then we declare that G is 3–realizable.

To summarize, we have the following theorem:

Theorem 4.2.21 *Given a graph G, one can decide whether it is 3–realizable in linear time.*

We remark that this is more efficient than the general fixed minor containment algorithm of Robertson and Seymour (1995), which runs in cubic time.

Finding a Subdivision of V_8 or $C_5 \times C_2$ in an 3–Realizable Graph

We note that the above strategy can also be employed to find a subdivision of $H \in \{V_8, C_5 \times C_2\}$ in an 3–realizable graph G. Indeed, we first decompose G into 3–connected components as before and check each of them for the presence or absence of V_8 or $C_5 \times C_2$. The latter can be achieved by running the linear time algorithm of Matoušek and Thomas (1991) on each of the 3–connected components and see if the component reduces to a null graph or not. If the component does not reduce to a null graph, then it is isomorphic to either V_8 or $C_5 \times C_2$, and the number of vertices in the component will determine which one it is. The desired subdivision can then be extracted from G using the linear time procedure outlined in Lemma 4.1 of Asano (1985). We summarize as follows:

Theorem 4.2.22 *Let G be an 3–realizable graph. Then, a subdivision of V_8 or $C_5 \times C_2$ in G can be found in linear time.*

Realizing 3–Realizable Graphs in $\mathbb{R}^3$

We now address the realization problem of 3–realizable graphs. To begin, let us consider the case of an 3–tree G. Let $\mathbf{d}$ be a set of realizable weights on the edges of G, and suppose that the vertices of G are numbered according to their order of insertion (recall Definition 4.2.2), say $V = \{1, \ldots, n\}$. Then, we can realize $(G, \mathbf{d})$ in $\mathbb{R}^3$ as follows. The first four vertices can be realized in $\mathbb{R}^3$ in a straightforward manner. Suppose that vertices $1, \ldots, l$ have been realized, and let $a_1, \ldots, a_l \in \mathbb{R}^3$ be their positions. Then, we can compute the position of vertex $l+1$ via its neighbors i_1, i_2, i_3, where $1 \leq i_1, i_2, i_3 \leq l$. Clearly, the above procedure can be done in linear time. If the order of the vertices is not given, we can first find an ordering using the linear time algorithm of Rose et al. (1976) and then proceed as above.

Now, suppose that we are given a partial 3–tree G and a set $\mathbf{d}$ of realizable weights on the edges. We first solve the SDP (3.9) to find an (approximate) realization of $(G, \mathbf{d})$ in $\mathbb{R}^l$ for some $l \geq k$. As a by–product, we obtain a set $\mathbf{d}'$ of realizable distances between all pairs of vertices in G. Then, we use the polynomial time algorithm of Arnborg and Proskurowski (1986) to complete G into an 3–tree G'. Since this completion procedure does not introduce any extra vertices, we can find a realization of $(G', \mathbf{d}'|_{E(G')})$, and hence an (approximate) realization of $(G, \mathbf{d})$, in $\mathbb{R}^3$ using the algorithm given in the preceding paragraph. We summarize as follows:

Theorem 4.2.23 *Let G be a graph and $\mathbf{d}$ be a set of weights on the edges.*

(1) If G is an 3–tree, then a realization of $(G, \mathbf{d})$ in $\mathbb{R}^3$ can be found in linear time.

(2) If G is a partial 3–tree, then an approximate realization of $(G, \mathbf{d})$ in $\mathbb{R}^3$ can be found in polynomial time.

Next, we consider the problem of realizing V_8 in $\mathbb{R}^3$. Let $\mathbf{d}$ be a set of realizable weights on the edges of V_8. Let V_8' be the graph obtained from V_8 by adding a strut between vertices 1 and 4 (refer to Figure 4.1(c) for the numbering of the vertices). Belk (2007) showed that once an unyielding realization $\mathbf{p}$ of $(V_8', \mathbf{d})$ and its associated non–zero equilibrium stress ω are available, one can "fold" the realization $\mathbf{p}$ of $(V_8, \mathbf{d})$ into $\mathbb{R}^3$ in constant time. By Theorems 4.2.12 and 4.2.16, an optimal solution to the following SDP (here, we assume without loss that vertex 1 is pinned at some known position a_1):

$$\begin{array}{lll} \sup & \bar{E}_{14} \bullet Z & \\ \text{subject to} & E_{ij} \bullet Z = d_{ij}^2 & (i,j) \in E(V_8);\ i,j \neq 1 \\ & \bar{E}_{1j} \bullet Z = \bar{d}_{1j}^2 & (1,j) \in E(V_8) \\ & Z \succeq 0,\ Z_{1:3,1:3} = I_3 & \end{array} \tag{4.12}$$

will give the desired unyielding realization, and the corresponding dual multipliers will supply the desired non–zero equilibrium stress. Thus, we have the following result:

Theorem 4.2.24 *Suppose that an optimal solution to (4.12) and its corresponding dual multipliers are available. Then, one can find a realization of* $(V_8, \mathbf{d})$ *in* $\mathbb{R}^3$ *in constant time.*

We remark that a similar approach works for $C_5 \times C_2$ as well. In particular, once an optimal solution to an appropriate SDP and its corresponding dual multipliers are available, one can find a realization of $(C_5 \times C_2, \mathbf{d})$ in $\mathbb{R}^3$ in constant time.

Finally, let us consider an arbitrary 3–realizable graph G and a set $\mathbf{d}$ of realizable weights on its edges. We first solve the SDP (3.9) to find an (approximate) realization of $(G, \mathbf{d})$ in $\mathbb{R}^l$ for some $l \geq k$. As a by–product, we obtain a set $\mathbf{d}'$ of realizable distances between all pairs of vertices in G. Now, motivated by Theorem 4.2.20, we find the set $\mathcal{S}$ of 3–connected components of G. For each component $H \in \mathcal{S}$, we check whether it is isomorphic to either V_8 or $C_5 \times C_2$. If it is not, then it is a partial 3–tree. In either case, we can use the information in $\mathbf{d}'$ to realize H in $\mathbb{R}^3$. Finally, by merging the components along the virtual edges or the articulation points, we obtain an (approximate) realization of $(G, \mathbf{d})$ in $\mathbb{R}^3$.

4.3 Relation to the Maximum Variance Unfolding Method

In the last section we saw how the idea of stretching apart pairs of non–adjacent vertices can be used to establish the 3–realizability of V_8 and $C_5 \times C_2$. As a corollary, we obtain a characterization of the class of 3–realizable graphs. It turns out that such an idea has also been used in the artifical intelligence community to detect and discover low–dimensional structure in high–dimensional data. For instance, in Weinberger and Saul (2006) (see also Weinberger et al. (2004)), the authors proposed the so–called *Maximum Variance Unfolding* (MVU) method for the problem of manifold learning. The idea is to map a given set of high–dimensional vectors $p_1, \ldots, p_n \in \mathbb{R}^l$ to a set of low–dimensional vectors $q_1, \ldots, q_n \in \mathbb{R}^k$ (where $1 \le k \ll l$ are given) with maximum total variance, while at the same time preserves the local distances. More precisely, consider an n–vertex connected graph $G = (V, E)$, where the set E of edges represents the set of distances that need to be preserved. The desired set of low–dimensional vectors can then be obtained by solving the following quadratic program:

$$\begin{array}{lll} \text{maximize} & \displaystyle\sum_{i=1}^{n} \|x_i\|^2 & \\ \text{subject to} & \displaystyle\sum_{i=1}^{n} x_i = \mathbf{0} & \\ & \|x_i - x_j\|^2 = \|p_i - p_j\|^2 & \text{for } (i,j) \in E \\ & x_i \in \mathbb{R}^k & \text{for } 1 \le i \le n \end{array} \tag{4.13}$$

To explain the rationale behind the above formulation, we observe that the first constraint centers the solution vectors at the origin and eliminates the translational degree of freedom. Moreover, it implies that the objective function of (4.13) can be written as:

$$\sum_{i=1}^{n} \|x_i\|^2 = \frac{1}{2n} \sum_{i,j=1}^{n} \|x_i - x_j\|^2$$

Thus, we see that the MVU method attempts to "unfold" the manifold by pulling the data points as far apart as possible while preserving the local distances. We remark that such a technique has also been used for the problem of sensor network localization (see, e.g., Biswas et al. (2006b); Weinberger et al. (2007)). Now, using the ideas in Chapter 3, we can formulate a semidefinite relaxation of (4.13) as follows:

$$\begin{array}{lll} \sup & I \bullet X & \\ \text{subject to} & ee^T \bullet X = 0 & \\ & E_{ij} \bullet X = \|v_i - v_j\|^2 & \text{for } (i,j) \in E \\ & X \succeq \mathbf{0} & \end{array} \tag{4.14}$$

Here, $e = (1, 1, \ldots, 1)$, $E_{ij} = (e_i - e_j)(e_i - e_j)^T$, and e_i is the i–th standard basis vector of $\mathbb{R}^n$. It turns out that problem (4.14) and its dual are closely related to the problem of finding the fastest mixing Markov process on a graph, as well as to various spectral methods for dimensionality reduction. We shall not elaborate on these results here and refer the interested reader to Sun et al. (2006); Xiao et al. (2006) for further details. Instead, we will show that the MVU problem (4.14) can be viewed as a problem of finding an unyielding configuration of a certain tensegrity. To begin, suppose that we are given an n–vertex connected graph $G = (\{1, \ldots, n\}, E)$ and a configuration $\mathbf{p} = (p_1, \ldots, p_n) \in \mathbb{R}^{ln}$ of the vertices. Consider the tensegrity $G'(\mathbf{p}')$, where G' is obtained from G by adding a new vertex $n+1$ and connecting it to all the vertices of G, and $\mathbf{p}' = (\mathbf{p}, \mathbf{0}) \in \mathbb{R}^{l(n+1)}$, i.e. vertex $n+1$ is located at the origin. Furthermore, we label the edges in E as bars and the edges in $S \equiv \{(n+1, i) : 1 \leq i \leq n\}$ as struts. Suppose that we pin vertex $n+1$ at the origin, i.e. $a_{n+1} = \mathbf{0}$. Now, consider the following SDP:

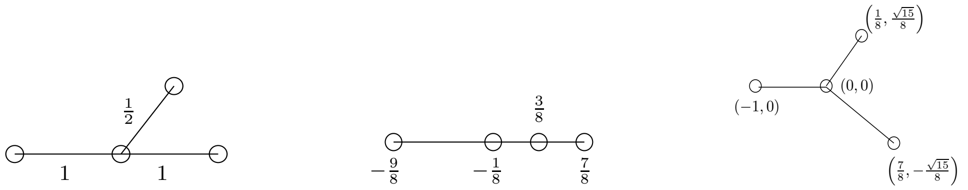
$$\begin{array}{lll} \sup & \displaystyle\sum_{i:(n+1,i)\in S} \bar{E}_{n+1,i} \bullet Z & \\ \text{subject to} & E_{ij} \bullet Z = \|p_i - p_j\|^2 & \text{for } (i,j) \in E \\ & Z \succeq \mathbf{0}, \; Z_{1:k,1:k} = I_k & \end{array} \tag{4.15}$$

where:

$$E_{ij} = \begin{pmatrix} \mathbf{0} \\ e_i - e_j \end{pmatrix} \begin{pmatrix} \mathbf{0} \\ e_i - e_j \end{pmatrix}^T \quad \text{and} \quad \bar{E}_{n+1,i} = \begin{pmatrix} \mathbf{0} \\ -e_i \end{pmatrix} \begin{pmatrix} \mathbf{0} \\ -e_i \end{pmatrix}^T$$

It is clear that (4.15) is an instance of (4.2). Moreover, by Theorem 4.2.12, Theorem 4.2.16 and Corollary 4.2.14, the positions $\bar{\mathbf{x}} \in \mathbb{R}^{ln}$ of the unpinned vertices obtained from the optimal primal matrix $\bar{Z}$ are automatically centered at the origin, even though such a constraint is not explicitly enforced. Thus, we see that problem (4.15) is equivalent to the MVU problem (4.14).

From the above discussion, we see that our formulation (4.2) is more general than the MVU formulation (4.14). Moreover, the flexibility in our formulation often allows us to achieve the desired dimensionality reduction which the MVU formulation cannot achieve. For instance, consider the case where the input graph G is a tree. By Theorem 4.2.3, G is 1–realizable, and it is not hard to show that there is a placement of struts such that *all* the optimal solutions to (4.2) have rank 1 and hence they all give rise to one–dimensional realizations. On the other hand, the MVU formulation may yield a two–dimensional realization. To see this, consider the instance shown in Figure 4.3(a). One of the one–dimensional configurations that achieves maximum variance is shown in Figure 4.3(b), where the variance is $\frac{35}{16}$. However, the two–dimensional configuration shown in Figure 4.3(c) has a variance of $\frac{9}{4}$. Since the MVU method always returns a realization of a tree that is at most two–dimensional (Xiao et al. (2006)), we conclude that the MVU method will yield a two–dimensional realization of the instance shown in Figure 4.3(a).


$\frac{1}{2}$
1
1
$\frac{3}{8}$
$-\frac{9}{8}$
$-\frac{1}{8}$
$\frac{7}{8}$
$\left(\frac{1}{8}, \frac{\sqrt{15}}{8}\right)$
$(0,0)$
$(-1,0)$
$\left(\frac{7}{8}, -\frac{\sqrt{15}}{8}\right)$

(a) Input Instance

(b) Optimal 1–d Configuration

(c) An 2–d Configuration with Higher Variance

Figure 4.3: Failure of Dimensionality Reduction by the MVU Method

It will be interesting to see how our approach can be applied to other dimensionality reduction problems. In particular, determining how the placement of struts and/or cables affects dimensionality reduction is a very interesting subject for further research.

Chapter 5

Rank Reduction in Semidefinite Programming

5.1 Introduction

In the previous chapters we studied the Graph Realization Problem (GRP) and presented notions such as unique k–realizability and k–realizability, which guarantee the existence of a realization of a given instance in $\mathbb{R}^k$. Moreover, using semidefinite programming techniques, we showed that such a realization can be found efficiently (up to arbitrary accuracy). So far our investigations have focused on the combinatorial and geometric aspects of (GRP). However, it is instructive to view our results from an algebraic perspective and interpret them in the context of *rank–constrained* semidefinite programming. Specifically, our results provide conditions that guarantee the feasibility and efficient solvability of certain system of linear matrix equations *with rank constraints*, i.e. systems of the form:

$$A_i \bullet X = b_i \qquad \text{for } 1 \le i \le m; \; X \succeq \mathbf{0}, \; \operatorname{rank}(X) \le k \tag{5.1}$$

Here, $k \ge 1$ is an integer, $A_1, \ldots, A_m$ are some $n \times n$ positive semidefinite matrices, and $b_1, \ldots, b_m$ are some non–negative real numbers. Indeed, recall the setting of the Graph Realization Problem, where we are given an integer $k \ge 1$, a connected graph $G = (V, E)$ with $V = \{0, 1, \ldots, n\}$, and a set $\mathbf{d} = (d_{ij}^2)_{(i,j) \in E}$ of weights on the edges. Suppose that we pin vertex 0 at the origin. Then, any matrix $\tilde{X}$ that is feasible for the following rank–constrained SDP:

$$\begin{array}{rcll} (e_i - e_j)(e_i - e_j)^T \bullet X & = & d_{ij}^2 & \text{for } (i,j) \in E;\ i,j \neq 0 \\ e_j e_j^T \bullet X & = & d_{0j}^2 & \text{for } (0,j) \in E \\ X & \succeq & \mathbf{0} & \\ \text{rank}(X) & \leq & k & \end{array}$$

will give rise to a realization of $(G, \mathbf{d})$ in $\mathbb{R}^k$. In general, system (5.1) is guaranteed to be feasible if $k \geq \lfloor (\sqrt{8m+1} - 1)/2 \rfloor$ (recall Theorem 3.3.5). However, it may not be feasible for smaller values of k. Moreover, it is NP–hard to determine the feasibility of (5.1), as it includes (GRP) as a special case. Thus, even if (5.1) is feasible, one may not be able to find a feasible solution efficiently. In view of the above discussion, it is then natural to ask whether one can *efficiently* find an $X_0 \succeq \mathbf{0}$ of rank at most k (where $k \geq 1$ is fixed) such that X_0 satisfies (5.1) *approximately*, i.e.:

$$\beta(m,n,d) \cdot b_i \leq A_i \bullet X_0 \leq \alpha(m,n,d) \cdot b_i \qquad \text{for } 1 \leq i \leq m \tag{5.2}$$

for some functions $\alpha \geq 1$ and $\beta \in (0,1]$. Of course, the main issue here is the quality of approximation, which will be determined by how close α and β are to 1. In the context of (GRP), system (5.2) says that we are allowed to find a realization in $\mathbb{R}^k$ that *distorts* the edge weights by a certain amount. This is a well–studied problem in the area of metric embedding (see, e.g., Chapter 15 of Matoušek (2002)), and we shall discuss it in more detail later.

Our goal of this chapter is to prove the following theorem, which first appeared in So et al. (2006):

Theorem 5.1.1 *Let $A_1, \ldots, A_m$ be $n \times n$ positive semidefinite matrices, and let $b_1, \ldots, b_m \geq 0$. Suppose that there exists an $X \succeq \mathbf{0}$ such that $A_i \bullet X = b_i$ for $1 \leq i \leq m$. Let $r = \min\{\sqrt{2m}, n\}$. Then, for any $k \geq 1$, there exists an $X_0 \succeq \mathbf{0}$ with rank$(X_0) \leq k$ such that:*

$$\beta(m,n,k) \cdot b_i \leq A_i \bullet X_0 \leq \alpha(m,n,k) \cdot b_i \qquad \textit{for } 1 \leq i \leq m$$

where:

$$\alpha(m,n,k) = \begin{cases} 1 + \dfrac{12\log(4mr)}{k} & \textit{for } 1 \le k \le 12\log(4mr) \\[2ex] 1 + \sqrt{\dfrac{12\log(4mr)}{k}} & \textit{for } k > 12\log(4mr) \end{cases} \tag{5.3}$$

and

$$\beta(m,n,k) = \begin{cases} \dfrac{1}{5e} \cdot \dfrac{1}{m^{2/k}} & \textit{for } 1 \le k \le \dfrac{2\log m}{\log\log(2m)} \\[2ex] \dfrac{1}{4e} \cdot \dfrac{1}{\log^{f(m)/k}(2m)} & \textit{for } \dfrac{2\log m}{\log\log(2m)} < k \le 4\log(4mr) \\[2ex] 1 - \sqrt{\dfrac{4\log(4mr)}{k}} & \textit{for } k > 4\log(4mr) \end{cases} \tag{5.4}$$

and $f(m) = \frac{3\log m}{\log\log(2m)}$*. Moreover, such an* X_0 *can be found in randomized polynomial time.*

Before we discuss the proof and the applications of Theorem 5.1.1, several remarks are in order.

Remarks.

(1) Note that $f(m)/k \le 3/2$ in the region $k > \frac{2\log m}{\log\log(2m)}$.

(2) The constants can be improved if we only consider an one–sided inequality.

(3) While the upper bound (5.3) depends on the parameter r (which can be viewed as a worst–case bound on $\max_{1\le i\le m}\operatorname{rank}(A_i)$), the lower bound (5.4) does not have such a dependence when $k \in \{1,\ldots,2\log m/\log\log(2m)\}$.

(4) From the definition of r, we see that the bounds above can be made independent of n and the ranks of $A_1,\ldots,A_m$.

(5) If $\max_{1\le i\le m}\operatorname{rank}(A_i) = O(1)$, then the lower bound can be sharpened to $\Omega\left(m^{-2/k}\right)$ for *all* $k \in \{1,\ldots,4\log(4mr)\}$; see the proof of Proposition 5.3.2.

5.2 Related Work

It turns out that Theorem 5.1.1 provides a unified treatment of and generalizes several results in the literature. These results in turn give some indication on the sharpness of the bounds derived in Theorem 5.1.1:

(1) *(Metric Embedding)* Let ℓ_2^l be the space $\mathbb{R}^l$ equipped with the Euclidean norm, and let ℓ_2 be the space of infinite sequences $x = (x_1, x_2, \ldots)$ of real numbers such that $\|x\|_2 \equiv \left(\sum_{j \geq 1} |x_j|^2\right)^{1/2} < \infty$. Given an n–point set $V = \{v_1, \ldots, v_n\}$ in ℓ_2^l, we would like to embed it into a low–dimensional Euclidean space as faithfully as possible. Specifically, we say that a map $f : V \to \ell_2$ is an *D–embedding* (where $D \geq 1$) if there exists a number $r > 0$ such that for all $u, v \in V$, we have:

$$r \cdot \|u - v\|_2 \leq \|f(u) - f(v)\|_2 \leq D \cdot r \cdot \|u - v\|_2$$

The goal is to find an f such that D is as small as possible. It is known that for any *fixed* $k \geq 1$, an $O\left(n^{2/k}\left(k^{-1}\log n\right)^{1/2}\right)$–embedding into ℓ_2^k exists; see, e.g., Matoušek (1990); Dasgupta and Gupta (1999). We now show how to derive this result from Theorem 5.1.1. Let e_i be the i–th standard basis vector in ℓ_2^n, and define $E_{ij} = (e_i - e_j)(e_i - e_j)^T$ for $1 \leq i < j \leq n$. Let U be the $l \times n$ matrix whose i–th column is the vector v_i, where $1 \leq i \leq n$. Then, it is clear that the matrix $X = U^T U$ satisfies the following system of equations:

$$E_{ij} \bullet X = \|v_i - v_j\|_2^2 \qquad \text{for } 1 \leq i < j \leq n$$

Now, Theorem 5.1.1 implies that we can find an $X_0 \succeq \mathbf{0}$ of rank at most k such that:

$$\Omega\left(n^{-4/k}\right) \cdot \|v_i - v_j\|_2^2 \leq E_{ij} \bullet X_0 \leq O\left(\frac{\log n}{k}\right) \cdot \|v_i - v_j\|_2^2 \qquad \text{for } 1 \leq i < j \leq n$$

Upon taking the Cholesky factorization $X_0 = U_0^T U_0$, we recover a set of points $u_1, \ldots, u_n \in \ell_2^k$ such that:

$$\Omega\left(n^{-2/k}\right) \cdot \|v_i - v_j\|_2 \leq \|u_i - u_j\|_2 \leq O\left(\sqrt{\frac{\log n}{k}}\right) \cdot \|v_i - v_j\|_2 \qquad \text{for } 1 \leq i < j \leq n$$

as desired. Clearly, any improvements on either (5.3) or (5.4) will immediately yield an improved bound on D for embeddings into ℓ_2^k. On the other hand, for any $k \geq 1$, there exists an n–point set V in ℓ_2^{k+1} such that *any* embedding of V into ℓ_2^k requires $D = \Omega\left(n^{1/\lfloor (k+1)/2 \rfloor}\right)$ (Matoušek (2002)). We should also point out that by using different techniques, Matoušek (1990) was able to show that in fact an $\Theta(n)$–embedding into ℓ_2^k exists for the cases where $k = 1, 2$.

Observe that in the above discussion, the affine span of the range of f is constrained to have dimension at most k. If there is no such constraint, then by the Johnson–Lindenstrauss lemma (see, e.g., Johnson and Lindenstrauss (1984); Dasgupta and Gupta (1999)), for any $\epsilon > 0$ and any n–point set V in ℓ_2, there exists an $(1+\epsilon)$–embedding of V into ℓ_2^k, where $k = O(\epsilon^{-2} \log n)$. Barvinok (2002) generalized this result and showed that if the assumptions of Theorem 5.1.1 are satisfied, then for any $\epsilon \in (0,1)$ and $k \geq 8\epsilon^{-2} \log(4m)$, there exists an $X_0 \succeq \mathbf{0}$ of rank at most k such that:

$$(1-\epsilon)b_i \leq A_i \bullet X_0 \leq (1+\epsilon)b_i \qquad \text{for } 1 \leq i \leq m$$

Thus, Theorem 5.1.1 complements Barvinok's result and generalizes the corresponding results in the study of bi–Lipschitz embeddings into low–dimensional Euclidean space (see, e.g., Matoušek (1990); Dasgupta and Gupta (1999)). We remark that the dependence of d on ϵ in the Johnson–Lindenstrauss lemma is almost tight. Specifically, Alon (2003) showed that there exists an n–point set V in ℓ_2 such that for any $\epsilon \in (n^{-1/2}, 1/2)$, say, an $(1+\epsilon)$–embedding of V into ℓ_2^k will require $k = \Omega((\epsilon^2 \log(1/\epsilon))^{-1} \log n)$.

(2) *(Quadratic Optimization with Homogeneous Quadratic Constraints)* Consider the following optimization problems:

$$\begin{array}{rcll} v^*_{maxqp} & = & \text{maximize} & x^T A x \\ & & \text{subject to} & x^T A_i x \leq 1 \qquad \text{for } 1 \leq i \leq m \end{array} \tag{5.5}$$

$$\begin{array}{rcll} v^*_{minqp} & = & \text{minimize} & x^T A x \\ & & \text{subject to} & x^T A_i x \geq 1 \qquad \text{for } 1 \leq i \leq m \end{array} \tag{5.6}$$

where $A_1, \ldots, A_m$ are $n \times n$ positive semidefinite matrices. Both of these problems arise from various applications (see Nemirovski et al. (1999); Luo et al. (2007)) and are NP–hard. Their natural SDP relaxations are given by:

$$\begin{array}{rcll} v^*_{maxsdp} & = & \text{maximize} & A \bullet X \\ & & \text{subject to} & A_i \bullet X \le 1 \qquad \text{for } 1 \le i \le m \\ & & & X \succeq \mathbf{0} \end{array} \tag{5.7}$$

$$\begin{array}{rcll} v^*_{minsdp} & = & \text{minimize} & A \bullet X \\ & & \text{subject to} & A_i \bullet X \ge 1 \qquad \text{for } 1 \le i \le m \\ & & & X \succeq \mathbf{0} \end{array} \tag{5.8}$$

It is clear that if $X = xx^T$ is a rank–1 feasible solution to (5.7) (resp. (5.8)), then x is a feasible solution to (5.5) (resp. (5.6)). Now, let X^*_{maxsdp} be an optimal solution to (5.7). It has been shown in Nemirovski et al. (1999) that one can extract a rank–1 matrix X_0 from X^*_{maxsdp} such that (i) X_0 is feasible for (5.7) and (ii) $A \bullet X_0 \ge \Omega\left(\frac{1}{\log m}\right) \cdot v^*_{maxqp}$. We now derive a similar result using Theorem 5.1.1. By definition, the matrix X^*_{maxsdp} satisfies the following system:

$$A \bullet X^*_{maxsdp} = v^*_{maxsdp}; \; A_i \bullet X^*_{maxsdp} = b_i \le 1 \qquad \text{for } 1 \le i \le m$$

As we shall see from the proof of Theorem 5.1.1, one can find a rank–1 matrix $X'_0 \succeq \mathbf{0}$ such that:

$$\mathbb{E}\left[A \bullet X'_0\right] = v^*_{maxsdp}; \; A_i \bullet X'_0 \le O(\log m) \cdot b_i \qquad \text{for } 1 \le i \le m$$

It follows that the matrix $X_0 = \Omega\left(\frac{1}{\log m}\right) \cdot X'_0 \succeq 0$ is feasible for (5.7), and that $\mathbb{E}\left[A \bullet X_0\right] = \Omega\left(\frac{1}{\log m}\right) \cdot v^*_{maxsdp} \ge \Omega\left(\frac{1}{\log m}\right) \cdot v^*_{maxqp}$. We remark that the gap between v^*_{maxsdp} and v^*_{maxqp} can be as large as $\Omega(\log m)$; see Nemirovski et al. (1999).

In a similar fashion, if X^*_{minsdp} is an optimal solution to (5.8), then one can extract a rank–1 matrix $X'_0 \succeq \mathbf{0}$ from X^*_{minsdp} such that $X_0 = O(m^2) \cdot X'_0$ is feasible for (5.8) and $\mathbb{E}\left[A \bullet X_0\right] = O(m^2) \cdot v^*_{minqp}$, thus recovering a result of Luo et al. (2007). In addition, the gap between v^*_{minqp} and v^*_{minsdp} can be as large as $\Omega(m^2)$; see Luo et al. (2007).

In Luo et al. (2007) the authors also considered complex versions of (5.5) and (5.6), in which the matrices A and A_i are complex positive semidefinite, and the components of the decision vector x can take on complex values. They showed that if X^*_{maxsdp} (resp. X^*_{minsdp}) is an optimal solution to the corresponding SDP relaxation (5.7) (resp. (5.8)), then one can extract a complex rank–1 solution that achieves $\Omega\left(\frac{1}{\log m}\right)$ (resp. $O(m)$) times the optimum value. Our result shows that these bounds are also achievable for the real versions of (5.7) and (5.8) if we allow the solution matrix to have rank at most 2. In particular, the complex versions of (5.5) and (5.6) with real positive semidefinite A and A_i's (i.e. only the decision vector takes on complex values) correspond precisely to the real versions of (5.7) and (5.8) with a rank–2 constraint on X.

5.3 Some Preliminary Probabilistic Results

We first make some standard preparatory moves (see, e.g., Barvinok (2002); Nemirovski et al. (1999); Luo et al. (2007)). Let $X \succeq \mathbf{0}$ be a solution to the system (5.1). By Theorem 3.3.5, we may assume that $r_0 \equiv \text{rank}(X) < \sqrt{2m}$. Let $X = UU^T$ for some $U \in \mathbb{R}^{n \times r_0}$, and set $A'_i = U^T A_i U \in \mathbb{R}^{r_0 \times r_0}$, where $1 \le i \le m$. Then, we have $A'_i \succeq \mathbf{0}$, $\text{rank}(A'_i) \le \min\{\text{rank}(A_i), r_0\}$, and

$$b_i = A_i \bullet X = \left(U^T A_i U\right) \bullet I = A'_i \bullet I = \text{Tr}(A'_i)$$

Moreover, if $X'_0 \succeq \mathbf{0}$ satisfies the inequalities:

$$\beta(m,n,d) \cdot b_i \le A'_i \bullet X'_0 \le \alpha(m,n,d) \cdot b_i \qquad \text{for } 1 \le i \le m$$

then upon setting $X_0 = UX'_0U^T \succeq 0$, we see that $\text{rank}(X_0) \le \text{rank}(X'_0)$, and

$$A_i \bullet X_0 = \left(U^T A_i U\right) \bullet X'_0 = A'_i \bullet X'_0$$

i.e. X_0 satisfies the inequalities in (5.2). Thus, in order to establish Theorem 5.1.1, it suffices to establish the following:

Theorem 5.1.1' *Let $A_1, \ldots, A_m$ be $n \times n$ positive semidefinite matrices, where $n < \sqrt{2m}$.*

Then, for any $k \geq 1$, there exists an $X_0 \succeq \mathbf{0}$ with rank$(X_0) \leq k$ such that:

$$\beta(m,n,k) \cdot \mathit{Tr}(A_i) \leq A_i \bullet X_0 \leq \alpha(m,n,k) \cdot \mathit{Tr}(A_i) \qquad \text{for } 1 \leq i \leq m$$

where $\alpha(m,n,k)$ and $\beta(m,n,k)$ are given by (5.3) and (5.4), respectively.

The proof of Theorem 5.1.1' relies on the following estimates of a chi–square random variable.

Proposition 5.3.1 *Let $\xi_1, \ldots, \xi_n$ be i.i.d. standard Gaussian random variables. Let $\alpha \in (1,\infty)$ and $\beta \in (0,1)$ be constants, and set $U_n = \sum_{i=1}^n \xi_i^2$. Note that $U_n \sim \chi_n^2$. Then, the following hold:*

$$\Pr\left(U_n \geq \alpha^2 n\right) \leq \left[\alpha \exp\left(\frac{1-\alpha^2}{2}\right)\right]^n = \exp\left[\frac{n}{2}\left(1-\alpha^2+2\log\alpha\right)\right] \tag{5.9}$$

$$\Pr\left(U_n \leq \beta^2 n\right) \leq \left[\beta \exp\left(\frac{1-\beta^2}{2}\right)\right]^n = \exp\left[\frac{n}{2}\left(1-\beta^2+2\log\beta\right)\right] \tag{5.10}$$

We defer the proof of Proposition 5.3.1 to the appendix.

In the sequel, let $k \geq 1$ be a given integer. Consider the following randomized procedure for generating an $X_0 \succeq \mathbf{0}$ of rank at most k. First, we generate i.i.d. Gaussian random variables ξ_i^j with mean 0 and variance $1/k$, and define $\xi^j = (\xi_1^j, \ldots, \xi_n^j)$, where $1 \leq i \leq n$ and $1 \leq j \leq k$. We then return $X_0 = \sum_{j=1}^k \xi^j \left(\xi^j\right)^T$. It is clear that $X_0 \succeq \mathbf{0}$ and has rank at most k. Also, note that the above procedure is different from those presented in Nemirovski et al. (1999); Luo et al. (2007).

Now, let $X_0 \succeq \mathbf{0}$ be the output of the above randomized procedure. Note that for any $H \in \mathbb{R}^{n\times n}$, we have $\mathbb{E}\left[H \bullet X_0\right] = \mathrm{Tr}(H)$. The following proposition forms the heart of our analysis.

Proposition 5.3.2 *Let H be an $n \times n$ positive semidefinite matrix. Consider the spectral decomposition $H = \sum_{l=1}^r \lambda_l v_l v_l^T$, where $r =$ rank(H) and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0$. Set $\bar{\lambda}_l = \lambda_l/(\lambda_1 + \cdots + \lambda_r)$. Then, for any $\alpha > 1$ and $\beta \in (0,1)$, we have:*

$$\Pr\left(H \bullet X_0 \geq \alpha\,\mathit{Tr}(H)\right) \leq r \cdot \exp\left[\frac{k}{2}\left(1-\alpha+\log\alpha\right)\right] \tag{5.11}$$

and

$$\Pr\left(H \bullet X_0 \leq \beta\, Tr(H)\right) \leq r \cdot \exp\left[\frac{k}{2}\left(1-\beta+\log\beta\right)\right] \leq r \cdot \exp\left[\frac{k}{2}\left(1+\log\beta\right)\right] \tag{5.12}$$

On the other hand, if β satisfies $e\beta\log r \leq 1/5$, then (5.12) can be sharpened to:

$$\Pr\left(H \bullet X_0 \leq \beta\, Tr(H)\right) \leq \left(\frac{5e\beta}{2}\right)^{k/2} \tag{5.13}$$

Proof. We first establish (5.11) and (5.12). Let $q_l = \sqrt{\lambda_l} \cdot v_l$. Then, we have $H = \sum_{l=1}^{r} q_l q_l^T$. Observe that $q_l^T \xi^j$ is a Gaussian random variable with mean 0 and variance $\sigma_l^2 \equiv k^{-1} \sum_s \left(q_l^T e_s\right)^2$, where e_s is the s–th coordinate vector. Moreover, we have:

$$\sum_{l=1}^{r} \sigma_l^2 = \frac{1}{k}\sum_{l=1}^{r}\sum_{s}\left(q_l^T e_s\right)^2 = \frac{1}{k}\cdot \mathrm{Tr}(H) \quad \text{and} \quad \mathbb{E}\left[\sum_{j=1}^{k}\left(q_l^T \xi^j\right)^2\right] = k \cdot \sigma_l^2$$

It follows that:

$$\Pr\left(\sum_{j=1}^{k}\left(q_l^T \xi^j\right)^2 \geq \alpha k \sigma_l^2\right) = \Pr(U_k \geq \alpha k) \leq \exp\left[\frac{k}{2}\left(1-\alpha+\log\alpha\right)\right] \qquad \text{for } 1 \leq l \leq r$$

and

$$\Pr\left(\sum_{j=1}^{k}\left(q_l^T \xi^j\right)^2 \leq \beta k \sigma_l^2\right) = \Pr\left(U_k \leq \beta k\right) \leq \exp\left[\frac{k}{2}\left(1-\beta+\log\beta\right)\right] \qquad \text{for } 1 \leq l \leq r$$

Now, observe that $H \bullet X_0 = \sum_{l=1}^{r}\sum_{j=1}^{k}\left(q_l^T \xi^j\right)^2$. Hence, we conclude that:

$$\Pr\left(H \bullet X_0 \geq \alpha \mathrm{Tr}(H)\right) \leq \sum_{l=1}^{r} \Pr\left(\sum_{j=1}^{k}\left(q_l^T \xi^j\right)^2 \geq \alpha k \sigma_l^2\right) \leq r \cdot \exp\left[\frac{d}{2}\left(1-\alpha+\log\alpha\right)\right]$$

and

$$\Pr\left(H \bullet X_0 \leq \beta \mathrm{Tr}(H)\right) \leq \sum_{l=1}^{r} \Pr\left(\sum_{j=1}^{k}\left(q_l^T \xi^j\right)^2 \leq \beta k \sigma_l^2\right) \leq r \cdot \exp\left[\frac{k}{2}\left(1-\beta+\log\beta\right)\right]$$

as desired.

To establish (5.13), we first recall that $H \bullet X_0 = \sum_{l=1}^{r} \sum_{j=1}^{k} \lambda_l \left(v_l^T \xi^j \right)^2$. Now, observe that $u = \left(v_l^T \xi^j \right)_{l,j} \sim \mathcal{N}(0, k^{-1} I_{rk})$. Indeed, $v_l^T \xi^j$ is a Gaussian random variable, as it is the sum of Gaussian random variables. Moreover, we have:

$$\mathbb{E}\left[v_l^T \xi^j \right] = 0 \quad \text{and} \quad \mathbb{E}\left[\left(v_l^T \xi^j \right) \left(v_s^T \xi^{j'} \right) \right] = \frac{1}{k} \cdot v_l^T v_s \cdot \mathbf{1}_{\{j=j'\}} = \frac{1}{k} \cdot \mathbf{1}_{\{l=s, j=j'\}}$$

It follows that $H \bullet X_0$ has the same distribution as $\sum_{l=1}^{r} \sum_{j=1}^{k} \lambda_l \tilde{\xi}_{lj}^2$, where $\tilde{\xi}_{lj}$ are i.i.d. Gaussian random variables with mean 0 and variance $1/k$. Now, we compute:

$$\Pr\left(H \bullet X_0 \leq \beta \mathrm{Tr}(H) \right) = \Pr\left(\sum_{l=1}^{r} \sum_{j=1}^{k} \lambda_l \tilde{\xi}_{lj}^2 \leq \beta \sum_{l=1}^{r} \lambda_l \right) = \Pr\left(\sum_{l=1}^{r} \sum_{j=1}^{k} \bar{\lambda}_l \tilde{\xi}_{lj}^2 \leq \beta \right)$$

Define:

$$p\left(r, \bar{\lambda}, \beta \right) \equiv \Pr\left(\sum_{l=1}^{r} \sum_{j=1}^{k} \bar{\lambda}_l \tilde{\xi}_{lj}^2 \leq \beta \right)$$

Then, by Proposition 5.3.1, we have:

$$p\left(r, \bar{\lambda}, \beta \right) \leq \Pr\left(\sum_{l=1}^{r} \sum_{j=1}^{k} \bar{\lambda}_r \tilde{\xi}_{lj}^2 \leq \beta \right) = \Pr\left(k \sum_{l=1}^{r} \sum_{j=1}^{k} \tilde{\xi}_{lj}^2 \leq \frac{\beta}{r \bar{\lambda}_r} \cdot rk \right) \leq \left(\frac{e\beta}{r\bar{\lambda}_r} \right)^{rk/2} \tag{5.14}$$

On the other hand, we have:

$$p\left(r, \bar{\lambda}, \beta \right) \leq \Pr\left(\sum_{l=1}^{r-1} \sum_{j=1}^{k} \bar{\lambda}_l \tilde{\xi}_{lj}^2 \leq \beta \right) = \Pr\left(\sum_{l=1}^{r-1} \sum_{j=1}^{k} \frac{\bar{\lambda}_l}{1-\bar{\lambda}_r} \tilde{\xi}_{lj}^2 \leq \frac{\beta}{1-\bar{\lambda}_r} \right)$$

Since

$$\frac{1}{1-\bar{\lambda}_r} \sum_{l=1}^{r-1} \bar{\lambda}_l = 1$$

it follows that:

$$p\left(r, \bar{\lambda}, \beta \right) \leq p\left(r-1, \frac{\bar{\lambda}_{1:r-1}}{1-\bar{\lambda}_r}, \frac{\beta}{1-\bar{\lambda}_r} \right) \tag{5.15}$$

Upon combining (5.14) and (5.15), we have:

$$p\left(r, \bar{\lambda}, \beta \right) \leq \min\left\{ \left(\frac{e\beta}{r\bar{\lambda}_r} \right)^{rk/2}, p\left(r-1, \frac{\bar{\lambda}_{1:r-1}}{1-\bar{\lambda}_r}, \frac{\beta}{1-\bar{\lambda}_r} \right) \right\}$$

and an easy inductive argument gives:

$$p\left(r,\bar{\lambda},\beta\right) \leq \min_{1\leq l\leq r}\left\{\left(\frac{e\beta}{l\bar{\lambda}_l}\right)^{lk/2}\right\} \tag{5.16}$$

Let $\alpha = p\left(r,\bar{\lambda},\beta\right)^{2/k}$. Note that $\alpha \in (0,1)$. By (5.16), we have $\bar{\lambda}_l \leq \left(l\alpha^{1/l}\right)^{-1} e\beta$ for $1 \leq l \leq r$. Upon summing over l and using the fact that $\sum_{l=1}^{r} \bar{\lambda}_l = 1$, we obtain:

$$\sum_{l=1}^{r} \frac{1}{l\alpha^{1/l}} \geq \frac{1}{e\beta} \tag{5.17}$$

If $r = 1$, then we have $\alpha \leq e\beta$. Henceforth, we shall assume that $r \geq 2$. Note that for any $\alpha \in (0,1)$, the function $t \mapsto (t\alpha^{1/t})^{-1}$ is decreasing for all $t \geq 1$, since we have:

$$\frac{d}{dt}\left(\frac{1}{t\alpha^{1/t}}\right) = \frac{\log\alpha - t}{t^3\alpha^{1/t}} < 0$$

Hence, it follows that:

$$\sum_{l=1}^{r} \frac{1}{l\alpha^{1/l}} \leq \frac{1}{\alpha} + \int_1^r \frac{1}{t\alpha^{1/t}}\,dt = \frac{1}{\alpha} + \int_{\frac{\log(1/\alpha)}{r}}^{\log(1/\alpha)} \frac{e^t}{t}\,dt \tag{5.18}$$

where we use the change of variable $z = -t^{-1}\log(1/\alpha)$ in the last step. Using the expansion:

$$\frac{e^t}{t} = \frac{1}{t}\sum_{j\geq 0}\frac{t^j}{j!} = \frac{1}{t} + \sum_{j\geq 0}\frac{t^j}{(j+1)!}$$

and the uniform convergence of $\sum_{j\geq 0} t^j/(j+1)!$ on compact intervals, we compute:

$$\begin{aligned}
\int_{\frac{\log(1/\alpha)}{r}}^{\log(1/\alpha)} \frac{e^t}{t}\,dt &= \log r + \sum_{j\geq 0}\frac{t^{j+1}}{(j+1)(j+1)!}\bigg|_{\frac{\log(1/\alpha)}{r}}^{\log(1/\alpha)} \\
&\leq \log r + \sum_{j\geq 0}\frac{\log^{j+1}(1/\alpha)}{(j+1)!} \\
&= \log r + \frac{1}{\alpha} - 1 \\
&\leq \log r + \frac{1}{\alpha}
\end{aligned} \tag{5.19}$$

Upon combining (5.17), (5.18) and (5.19), we conclude that:

$$\frac{1}{e\beta} \leq \frac{2}{\alpha} + \log r$$

which, together with the assumption that $e\beta \log r \leq 1/5$, implies that $\alpha \leq 5e\beta/2$. ■

5.4 Proof of Theorem 5.1.1

We now complete the proof of Theorem 5.1.1' and hence of Theorem 5.1.1 using the probabilistic results developed in the previous section. We first establish the upper bound. We write $\alpha = 1 + \alpha'$ for some $\alpha' > 0$. Using the inequality $\log(1+x) \leq x - x^2/2 + x^3/3$, which is valid for all $x > 0$, it is easy to show that:

$$1 - \alpha + \log \alpha = -\alpha' + \log(1 + \alpha') \leq \begin{cases} -\dfrac{\alpha'}{6} & \text{for } \alpha' \geq 1 \\ -\dfrac{\alpha'^2}{6} & \text{for } 0 < \alpha' < 1 \end{cases} \tag{5.20}$$

Let $T = \frac{12\log(4mn)}{k}$. If $T \geq 1$, then set $\alpha' = T$; otherwise, set $\alpha' = \sqrt{T}$. In the former case, we have $\alpha' \geq 1$, and hence by Proposition 5.3.2 and the bound in (5.20), for each $1 \leq i \leq m$, we have:

$$\Pr\left(A_i \bullet X_0 \geq \alpha \mathrm{Tr}(A_i)\right) \leq \mathrm{rank}(A_i) \cdot \exp\left(-\frac{k\alpha'}{12}\right) \leq \frac{1}{4m}$$

where the last inequality follows from the fact that $\mathrm{rank}(A_i) \leq n$. In the latter case, we have $\alpha' \in (0,1)$, and a similar calculation shows that:

$$\Pr\left(A_i \bullet X_0 \geq \alpha \mathrm{Tr}(A_i)\right) \leq \mathrm{rank}(A_i) \cdot \exp\left(-\frac{k\alpha'^2}{12}\right) \leq \frac{1}{4m}$$

for each $1 \leq i \leq m$. Hence, we conclude that:

$$\Pr\left(A_i \bullet X_0 \leq \alpha(m,n,k) \cdot \mathrm{Tr}(A_i) \text{ for all } 1 \leq i \leq m\right) \geq 1 - \frac{1}{4} = \frac{3}{4} \tag{5.21}$$

where $\alpha(m,n,k)$ is given by (5.3).

Next, we establish the lower bound. We consider the following cases:

Case 1: $1 \leq k \leq \frac{2\log m}{\log\log(2m)}$

Let $\beta = \left(5em^{2/k}\right)^{-1}$ in Proposition 5.3.2. Since $r < \sqrt{2m}$, we have:

$$e\beta \log r < \frac{1}{10m^{2/k}} \log 2m \leq \frac{1}{10} < \frac{1}{5}$$

by our choice of k. It follows that (5.13) of Proposition 5.3.2 applies, and we conclude that:

$$\Pr\left(A_i \bullet X_0 \leq \beta \mathrm{Tr}(A_i)\right) \leq \left(\frac{1}{2}\right)^{k/2} \cdot \frac{1}{m} \qquad \text{for } 1 \leq i \leq m$$

Together with (5.21), we have:

$$\Pr\left(\beta\mathrm{Tr}(A_i) \leq A_i \bullet X_0 \leq \alpha(m,n,k) \cdot \mathrm{Tr}(A_i) \text{ for all } 1 \leq i \leq m\right) \geq \frac{3}{4} - \left(\frac{1}{2}\right)^{k/2} > 0$$

for all $k \geq 1$.

Case 2: $\frac{2\log m}{\log\log(2m)} < k \leq 4\log(4mn)$

Suppose that $k = \frac{l \log m}{\log\log(2m)}$ for some $l > 2$. Let $\beta = \left(4e\log^{3/l}(2m)\right)^{-1}$ in Proposition 5.3.2. Upon noting that $m^{3/k} = \log^{3/l}(2m)$ and using (5.12) of Proposition 5.3.2, we have:

$$\Pr\left(A_i \bullet X_0 \leq \beta\mathrm{Tr}(A_i)\right) \leq \mathrm{rank}(A_i) \cdot (e\beta)^{k/2} \leq \mathrm{rank}(A_i) \cdot \left(\frac{1}{2}\right)^k \cdot \frac{1}{m^{3/2}} < \sqrt{2} \cdot \left(\frac{1}{2}\right)^k \cdot \frac{1}{m}$$

Together with (5.21), we have:

$$\Pr\left(\beta\mathrm{Tr}(A_i) \leq A_i \bullet X_0 \leq \alpha(m,n,k) \cdot \mathrm{Tr}(A_i) \text{ for all } 1 \leq i \leq m\right) \geq \frac{3}{4} - \sqrt{2}\left(\frac{1}{2}\right)^k > 0$$

for all $k \geq 2$.

Case 3: $k > 4\log(4mn)$

We write $\beta = 1 - \beta'$ for some $\beta' \in (0,1)$. Using the inequality $\log(1-x) \leq -x - x^2/2$, which is valid for all $x \in [0,1]$, we have:

$$1 - \beta + \log\beta = \beta' + \log(1-\beta') \leq -\frac{\beta'^2}{2}$$

Let $\beta' = \left(\frac{4\log(4mn)}{k}\right)^{1/2}$. By assumption, we have $\beta' \in (0,1)$. By (5.12) of Proposition

5.3.2, for each $1 \leq i \leq m$, we have:

$$\Pr\left(A_i \bullet X_0 \leq \beta \mathrm{Tr}(A_i)\right) \leq \mathrm{rank}(A_i) \cdot \exp\left(-\frac{k\beta'^2}{4}\right) \leq \frac{1}{4m}$$

It follows that:

$$\Pr\left(\beta\mathrm{Tr}(A_i) \leq A_i \bullet X_0 \leq \alpha(m,n,k) \cdot \mathrm{Tr}(A_i) \text{ for all } 1 \leq i \leq m\right) \geq \frac{3}{4} - \frac{1}{4} = \frac{1}{2}$$

This completes the proof of Theorem 5.1.1' and hence of Theorem 5.1.1.

5.5 Appendix

We now complete the proof of Proposition 5.3.1. To establish (5.9), we let $t \in [0,1/2)$ and compute:

$$\begin{aligned}\Pr\left(U_n \geq \alpha^2 n\right) &= \Pr\left\{\exp\left[t\left(U_n - \alpha^2 n\right)\right] \geq 1\right\} \\ &\leq \mathbb{E}\left[\exp\left[t\left(U_n - \alpha^2 n\right)\right]\right] && \text{(by Markov's inequality)} \\ &= \exp\left(-t\alpha^2 n\right) \cdot \left(\mathbb{E}\left[\exp\left(t\xi_1^2\right)\right]\right)^n && \text{(by independence)} \\ &= \exp\left(-t\alpha^2 n\right) \cdot (1-2t)^{-n/2}\end{aligned}$$

Let $f : [0,1/2) \to \mathbb{R}$ be given by $f(t) = \exp\left(-t\alpha^2 n\right) \cdot (1-2t)^{-n/2}$. Then, we have:

$$f'(t) = -\exp\left(-t\alpha^2 n\right)\alpha^2 n(1-2t)^{-n/2} + \exp\left(-t\alpha^2 n\right) n(1-2t)^{-(n/2+1)}$$

and hence f is minimized at $t^* = \left(1-\alpha^{-1}\right)/2$. Note that $t^* \in (0,1/2)$ whenever $\alpha \in (1,\infty)$. Thus, we conclude that:

$$\Pr\left(U_n \geq \alpha^2 n\right) \leq f\left(t^*\right) = \left[\alpha \exp\left(\frac{1-\alpha^2}{2}\right)\right]^n$$

To establish (5.10), we proceed in a similar fashion. For $t \geq 0$, we have:

$$\begin{aligned} \Pr\left(U_n \leq \beta^2 n\right) &= \Pr\left\{\exp\left[t\left(\beta^2 n - U_n\right)\right] \geq 1\right\} \\ &\leq \mathbb{E}\left[\exp\left[t\left(\beta^2 n - U_n\right)\right]\right] && \text{(by Markov's inequality)} \\ &= \exp\left(t\beta^2 n\right) \cdot \left(\mathbb{E}\left[\exp\left(-t\xi_1^2\right)\right]\right)^n && \text{(by independence)} \\ &= \exp\left(t\beta^2 n\right) \cdot (1+2t)^{-n/2} \end{aligned}$$

Now, let $f : [0,\infty) \to \mathbb{R}$ be given by $f(t) = \exp\left(t\beta^2 n\right) \cdot (1+2t)^{-n/2}$. Then, we have:

$$f'(t) = \exp\left(t\beta^2 n\right)\beta^2 n(1+2t)^{-n/2} - \exp\left(t\beta^2 n\right) n(1+2t)^{-(n/2+1)}$$

and hence f is minimized at $t^* = \left(\beta^{-2} - 1\right)/2$. Moreover, we have $t^* > 0$ whenever $\beta < 1$. It follows that:

$$\Pr\left(U_n \leq \beta^2 n\right) \leq f\left(t^*\right) = \left[\beta \exp\left(\frac{1-\beta^2}{2}\right)\right]^n$$

as desired.

Bibliography

Aho, A. V., Hopcroft, J. E., and Ullman, J. D. (1974). *The Design and Analysis of Computer Algorithms*. Addison–Wesley Series in Computer Science and Information Processing. Addison–Wesley, Reading, Massachusetts.

Ai, W., Huang, Y., and Zhang, S. (2006). On the Low Rank Solutions for Linear Matrix Inequalities. Technical Report SEEM2006–08, Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong.

Alfakih, A. Y., Khandani, A., and Wolkowicz, H. (1999). Solving Euclidean Distance Matrix Completion Problems via Semidefinite Programming. *Computational Optimization and Applications*, 12:13–30.

Alfakih, A. Y. and Wolkowicz, H. (1998). On the Embeddability of Weighted Graphs in Euclidean Spaces. Technical Report CORR 1998–12, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada.

Alizadeh, F. (1995). Interior Point Methods in Semidefinite Programming with Applications to Combinatorial Optimization. *SIAM Journal on Optimization*, 5(1):13–51.

Alizadeh, F., Haeberly, J.-P. A., and Overton, M. L. (1997). Complementarity and Nondegeneracy in Semidefinite Programming. *Mathematical Programming*, 77(2):111–128.

Alizadeh, F., Haeberly, J.-P. A., and Overton, M. L. (1998). Primal–Dual Interior–Point Methods for Semidefinite Programming: Convergence Rates, Stability and Numerical Results. *SIAM Journal on Optimization*, 8(3):746–768.

Alon, N. (2003). Problems and Results in Extremal Combinatorics I. *Discrete Mathematics*, 273:31–53.

Arnborg, S. and Proskurowski, A. (1986). Characterization and Recognition of Partial 3–Trees. *SIAM Journal on Algebraic and Discrete Methods*, 7(2):305–314.

Arnborg, S., Proskurowski, A., and Corneil, D. G. (1990). Forbidden Minors Characterization of Partial 3–Trees. *Discrete Mathematics*, 80(1):1–19.

Asano, T. (1985). An Approach to the Subgraph Homeomorphism Problem. *Theoretical Computer Science*, 38:249–267.

Aspnes, J., Eren, T., Goldenberg, D. K., Morse, A. S., Whiteley, W., Yang, Y. R., Anderson, B. D. O., and Belhumeur, P. N. (2006). A Theory of Network Localization. *IEEE Transactions on Mobile Computing*, 5(12):1663–1678.

Aspnes, J., Goldenberg, D., and Yang, Y. R. (2004). On the Computational Complexity of Sensor Network Localization. In Nikoletseas, S. and Rolim, J. D. P., editors, *Proceedings of the 1st International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004)*, volume 3121 of *Lecture Notes in Computer Science*, pages 32–44. Springer–Verlag.

Barvinok, A. (2001). A Remark on the Rank of Positive Semidefinite Matrices Subject to Affine Constraints. *Discrete and Computational Geometry*, 25(1):23–31.

Barvinok, A. (2002). *A Course in Convexity*, volume 54 of *Graduate Studies in Mathematics.* American Mathematical Society, Providence, Rhode Island.

Barvinok, A. I. (1995). Problems of Distance Geometry and Convex Properties of Quadratic Maps. *Discrete and Computational Geometry*, 13:189–202.

Basu, A., Gao, J., Mitchell, J. S. B., and Sabhnani, G. (2006). Distributed Localization Using Noisy Distance and Angle Information. In *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2006)*, pages 262–273.

Belk, M. (2007). Realizability of Graphs in Three Dimensions. *Discrete and Computational Geometry*, 37(2):139–162.

Belk, M. and Connelly, R. (2007). Realizability of Graphs. *Discrete and Computational Geometry*, 37(2):125–137.

Benson, S. J., Ye, Y., and Zhang, X. (2000). Solving Large–Scale Sparse Semidefinite Programs for Combinatorial Optimization. *SIAM Journal on Optimization*, 10(2):443–461.

Bertoni, A., Mauri, G., and Sabadini, N. (1985). Simulations among Classes of Random Access Machines and Equivalence among Numbers Succintly Represented. In Ausiello, G. and Lucertini, M., editors, *Analysis and Design of Algorithms for Combinatorial Problems*, volume 25 of *Annals of Discrete Mathematics*, pages 65–90. Elsevier Science Publishers B.V.

Biswas, P., Lian, T.-C., Wang, T.-C., and Ye, Y. (2006a). Semidefinite Programming Based Algorithms for Sensor Network Localization. *ACM Transactions on Sensor Networks*, 2(2):188–220.

Biswas, P., Liang, T.-C., Toh, K.-C., Wang, T.-C., and Ye, Y. (2006b). Semidefinite Programming Approaches for Sensor Network Localization with Noisy Distance Measurements. *IEEE Transactions on Automation Science and Engineering*, 3(4):360–371.

Biswas, P., Toh, K.-C., and Ye, Y. (2007). A Distributed SDP Approach for Large–Scale Noisy Anchor–Free Graph Realization with Applications to Molecular Conformation. Manuscript.

Biswas, P. and Ye, Y. (2004). Semidefinite Programming for Ad Hoc Wireless Sensor Network Localization. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN 2004)*, pages 46–54.

Blumenthal, L. M. (1953). *Theory and Applications of Distance Geometry.* Clarendon Press, Oxford.

Boissonnat, J.-D. and Yvinec, M. (1998). *Algorithmic Geometry.* Cambridge University Press, Cambridge.

Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V. (1994). *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *SIAM Studies in Applied and Numerical Mathematics.* Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania.

Bădoiu, M., Demaine, E. D., Hajiaghayi, M., and Indyk, P. (2006). Low–Dimensional Embedding with Extra Information. *Discrete and Computational Geometry*, 36(4):609–632.

Connelly, R. (1982). Rigidity and Energy. *Inventiones Mathematicae*, 66:11–33.

Connelly, R. (2005). Generic Global Rigidity. *Discrete and Computational Geometry*, 33(4):549–563.

Connelly, R. and Sloughter, M. (2004). Realizability of Graphs. Manuscript.

Crippen, G. M. and Havel, T. F. (1988). *Distance Geometry and Molecular Conformation*, volume 15 of *Chemometrics Series*. Research Studies Press Ltd., Taunton, Somerset, England.

Dasgupta, S. and Gupta, A. (1999). An Elementary Proof of the Johnson–Lindenstrauss Lemma. Technical Report TR–99–006, International Computer Science Institute, 1947 Center St., Suite 600, Berkeley, CA 94704–1198.

de Klerk, E., Roos, C., and Terlaky, T. (1997). Initialization in Semidefinite Programming via a Self–Dual Skew–Symmetric Embedding. *Operations Research Letters*, 20:213–221.

de Klerk, E., Roos, C., and Terlaky, T. (1998). Infeasible–Start Semidefinite Programming Algorithms via Self–Dual Embeddings. In Pardalos, P. M. and Wolkowicz, H., editors, *Topics in Semidefinite and Interior–Point Methods*, volume 18 of *The Fields Institute for Research in Mathematical Sciences, Communications Series*, pages 215–236. American Mathematical Society.

Diestel, R. (2005). *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer–Verlag, Heidelberg, third edition.

Ding, Y., Krislock, N., Qian, J., and Wolkowicz, H. (2006). Sensor Network Localization, Euclidean Distance Matrix Completions, and Graph Realization. Technical Report CORR 2006–23, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada.

Doherty, L., Pister, K. S. J., and El Ghaoui, L. (2001). Convex Position Estimation in Wireless Sensor Networks. In *Proceedings of the 20th Annual IEEE Conference on Computer Communications (INFOCOM 2001)*, volume 3, pages 1655–1663.

Edmonds, J. (1967). Systems of Distinct Representatives and Linear Algebra. *Journal of Research of the National Bureau of Standards — B. Mathematics and Mathematical Physics*, 71B(4):241–245.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP–Completeness*. W. H. Freeman and Company, New York.

Gower, J. C. (1966). Some Distance Properties of Latent Root and Vector Methods Used in Multivariate Analysis. *Biometrika*, 53(3/4):325–338.

Graver, J., Servatius, B., and Servatius, H. (1993). *Combinatorial Rigidity*, volume 2 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, Rhode Island.

Grötschel, M., Lovász, L., and Schrijver, A. (1993). *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer–Verlag, Berlin/Heidelberg, second corrected edition.

Havel, T. F. (2003). Metric Matrix Embedding in Protein Structure Calculations, NMR Spectra Analysis, and Relaxation Theory. *Magnetic Resonance in Chemistry*, 41(S1):S37–S50.

Havel, T. F. and Wüthrich, K. (1985). An Evaluation of the Combined Use of Nuclear Magnetic Resonance and Distance Geometry for the Determination of Protein Conformations in Solution. *Journal of Molecular Biology*, 182(2):281–294.

Helmberg, C. (2000). Semidefinite Programming for Combinatorial Optimization. Technical Report ZR–00–34, Konrad–Zuse–Zentrum für Informationstechnik Berlin, Takustraße 7, D–14195, Berlin, Germany.

Hendrickson, B. (1992). Conditions for Unique Graph Realizations. *SIAM Journal on Computing*, 21(1):65–84.

Hendrickson, B. (1995). The Molecule Problem: Exploiting Structure in Global Optimization. *SIAM Journal on Optimization*, 5(4):835–857.

Hendrickson, B. A. (1990). *The Molecule Problem: Determining Conformation from Pairwise Distances*. PhD thesis, Department of Computer Science, Cornell University, Ithaca, NY 14853–7501. Technical Report TR90–1159.

Hiriart-Urruty, J.-B. and Lemaréchal, C. (2001). *Fundamentals of Convex Analysis.* Springer–Verlag, Berlin/Heidelberg.

Hopcroft, J. E. and Tarjan, R. E. (1973). Dividing a Graph into Triconnected Components. *SIAM Journal on Computing*, 2(3):135–158.

Horn, R. A. and Johnson, C. R. (1985). *Matrix Analysis.* Cambridge University Press, Cambridge.

Jackson, B. and Jordán, T. (2005). Connected Rigidity Matroids and Unique Realizations of Graphs. *Journal of Combinatorial Theory, Series B*, 94(1):1–29.

Johnson, W. B. and Lindenstrauss, J. (1984). Extensions of Lipschitz Mapping into Hilbert Space. In Beals, R., Beck, A., Bellow, A., and Hajian, A., editors, *Conference in Modern Analysis and Probability*, volume 26 of *Contemporary Mathematics*, pages 189–206. American Mathematical Society.

Kaptein, R., Boelens, R., Scheek, R. M., and van Gunsteren, W. F. (1988). Protein Structures from NMR. *Biochemistry*, 27(15):5389–5395.

Laurent, M. (1997). Cuts, Matrix Completions and Graph Rigidity. *Mathematical Programming*, 79:255–283.

Laurent, M. (1998a). A Connection Between Positive Semidefinite and Euclidean Distance Matrix Completion Problems. *Linear Algebra and Its Applications*, 273:9–22.

Laurent, M. (1998b). A Tour d'Horizon on Positive Semidefinite and Euclidean Distance Matrix Completion Problems. In Pardalos, P. M. and Wolkowicz, H., editors, *Topics in Semidefinite and Interior–Point Methods*, volume 18 of *The Fields Institute for Research in Mathematical Sciences, Communications Series*, pages 51–76. American Mathematical Society.

Laurent, M. (2000). Polynomial Instances of the Positive Semidefinite and Euclidean Distance Matrix Completion Problems. *SIAM Journal on Matrix Analysis and Applications*, 22(3):874–894.

Laurent, M. (2001). Matrix Completion Problems. In Floudas, C. A. and Pardalos, P. M., editors, *The Encyclopedia of Optimization*, volume 3, pages 221–229. Kluwer Academic Publishers.

Lovász, L. (2000). Semidefinite Programs and Combinatorial Optimization. Lecture notes available at `http://research.microsoft.com/users/lovasz/survey.htm`.

Luo, Z.-Q., Sidiropoulos, N. D., Tseng, P., and Zhang, S. (2007). Approximation Bounds for Quadratic Optimization with Homogeneous Quadratic Constraints. *SIAM Journal on Optimization*, 18(1):1–28.

Matoušek, J. (1990). Bi–Lipschitz Embeddings into Low–Dimensional Euclidean Spaces. *Commentationes Mathematicae Universitatis Carolinae*, 31(3):589–600.

Matoušek, J. (2002). *Lectures on Discrete Geometry*, volume 212 of *Graduate Texts in Mathematics*. Springer–Verlag, New York.

Matoušek, J. and Thomas, R. (1991). Algorithms Finding Tree–Decompositions of Graphs. *Journal of Algorithms*, 12(1):1–22.

Menger, K. (1931). New Foundation of Euclidean Geometry. *Americal Journal of Mathematics*, 53(4):721–745.

Monteiro, R. D. C. and Tsuchiya, T. (1999). Polynomial Convergence of a New Family of Primal–Dual Algorithms for Semidefinite Programming. *SIAM Journal on Optimization*, 9(3):551–577.

Monteiro, R. D. C. and Zanjácomo, P. (1999). Implementation of Primal–Dual Methods for Semidefinite Programming Based on Monteiro and Tsuchiya Newton Directions and Their Variants. *Optimization Methods and Software*, 11/12:91–140.

Moré, J. J. and Wu, Z. (1997). Global Continuation for Distance Geometry Problems. *SIAM Journal on Optimization*, 7(3):814–836.

Nemirovski, A. (2005). Lectures on Modern Convex Optimization. Lecture notes available at `http://www2.isye.gatech.edu/~nemirovs/`.

Nemirovski, A., Roos, C., and Terlaky, T. (1999). On Maximization of Quadratic Form over Intersection of Ellipsoids with Common Center. *Mathematical Programming, Series A*, 86(3):463–473.

Nesterov, Y., Todd, M. J., and Ye, Y. (1999). Infeasible–Start Primal–Dual Methods and Infeasibility Detectors for Nonlinear Programming Problems. *Mathematical Programming*, 84(2):227–267.

Nesterov, Y. E. and Todd, M. J. (1998). Primal–Dual Interior–Point Methods for Self–Scaled Cones. *SIAM Journal on Optimization*, 8(2):324–364.

Nie, J. (2006). Sum of Squares Method for Sensor Network Localization. Manuscript.

Pataki, G. (1996). Cone–LP's and Semidefinite Programs: Geometry and a Simplex–Type Method. In Cunningham, W. H., McCormick, S. T., and Queyranne, M., editors, *Proceedings of the 5th Conference on Integer Programming and Combinatorial Optimization (IPCO 1996)*, volume 1084 of *Lecture Notes of Computer Science*, pages 162–174. Springer–Verlag.

Pataki, G. (1998). On the Rank of Extreme Matrices in Semidefinite Programs and the Multiplicity of Optimal Eigenvalues. *Mathematics of Operations Research*, 23(2):339–358.

Porkolab, L. and Khachiyan, L. (1997). On the Complexity of Semidefinite Programs. *Journal of Global Optimization*, 10(4):351–365.

Ramana, M. V. (1997). An Exact Duality Theory for Semidefinite Programming and Its Complexity Implications. *Mathematical Programming*, 77(2):129–162.

Robertson, N. and Seymour, P. D. (1995). Graph Minors. XIII. The Disjoint Paths Problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110.

Robertson, N. and Seymour, P. D. (2004). Graph Minors. XX. Wagner's Conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357.

Rose, D. J., Tarjan, R. E., and Lueker, G. S. (1976). Algorithmic Aspects of Vertex Elimination on Graphs. *SIAM Journal on Computing*, 5(2):266–283.

Roth, B. and Whiteley, W. (1981). Tensegrity Frameworks. *Transactions of the American Mathematical Society*, 265(2):419–446.

Savvides, A., Han, C.-C., and Strivastava, M. B. (2001). Dynamic Fine–Grained Localization in Ad–Hoc Networks of Sensors. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom 2001)*, pages 166–179.

Saxe, J. B. (1979). Embeddability of Weighted Graphs in k–Space is Strongly NP–Hard. In *Proceedings of the 17th Allerton Conference in Communication, Control, and Computing*, pages 480–489.

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. *Annals of Mathematics*, 36(3):724–732.

Schoenberg, I. J. (1938). Metric Spaces and Positive Definite Functions. *Transactions of the American Mathematical Society*, 44(3):522–536.

Schönhage, A. (1979). On the Power of Random Access Machines. In Maurer, H. A., editor, *Proceedings of the 6th Colloquium on Automata, Languages and Programming (ICALP 79)*, volume 71 of *Lecture Notes in Computer Science*, pages 520–529. Springer–Verlag.

Shang, Y., Ruml, W., Zhang, Y., and Fromherz, M. (2004). Localization from Connectivity in Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems*, 15(11):961–974.

So, A. M.-C. and Ye, Y. (2006). A Semidefinite Programming Approach to Tensegrity Theory and Realizability of Graphs. In *Proceedings of the 17th Annual ACM–SIAM Symposium on Discrete Algorithm (SODA 2006)*, pages 766–775.

So, A. M.-C. and Ye, Y. (2007). Theory of Semidefinite Programming for Sensor Network Localization. *Mathematical Programming, Series B*, 109(2):367–384.

So, A. M.-C., Ye, Y., and Zhang, J. (2006). A Unified Theorem on SDP Rank Reduction. Manuscript.

Sun, J., Boyd, S., Xiao, L., and Diaconis, P. (2006). The Fastest Mixing Markov Process on a Graph and a Connection to a Maximum Variance Unfolding Problem. *SIAM Review*, 48(4):681–699.

Torgerson, W. S. (1952). Multidimensional Scaling: I. Theory and Method. *Psychometrika*, 17(4):401–419.

Tseng, P. (2007). Second–Order Cone Programming Relaxation of Sensor Network Localization. *SIAM Journal on Optimization*, 18(1):156–185.

Valdes, J., Tarjan, R. E., and Lawler, E. L. (1982). The Recognition of Series Parallel Digraphs. *SIAM Journal on Computing*, 11(2):298–313.

Wald, J. A. and Colbourn, C. J. (1983). Steiner Trees, Partial 2–Trees, and Minimum IFI Networks. *Networks*, 13(2):159–167.

Wang, Z., Zheng, S., Boyd, S., and Ye, Y. (2006). Further Relaxations of the SDP Approach to Sensor Network Localization. Manuscript.

Weinberger, K. Q. and Saul, L. K. (2006). Unsupervised Learning of Image Manifolds by Semidefinite Programming. *International Journal of Computer Vision*, 70(1):77–90.

Weinberger, K. Q., Sha, F., and Saul, L. K. (2004). Learning a Kernel Matrix for Nonlinear Dimensionality Reduction. In *Proceedings of the 21st International Conference on Machine Learning (ICML 2004)*, pages 839–846.

Weinberger, K. Q., Sha, F., Zhu, Q., and Saul, L. K. (2007). Graph Laplacian Regularization for Large–Scale Semidefinite Programming. In Schölkopf, B., Platt, J., and Hofmann, T., editors, *Advances in Neural Information Processing Systems 19*, pages 1489–1496. MIT Press.

Wells, J. H. and Williams, L. R. (1975). *Embeddings and Extensions in Analysis.* Ergebnisse der Mathematik und ihrer Grenzgebiete, Band 84. Springer–Verlag, Berlin/Heidelberg.

Wright, S. J. (1997). *Primal–Dual Interior–Point Methods.* Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania.

Wüthrich, K. (1989). The Development of Nuclear Magnetic Resonance Spectroscopy as a Technique for Protein Structure Determination. *Accounts of Chemical Research*, 22(1):36–44.

Xiao, L., Sun, J., and Boyd, S. (2006). A Duality View of Spectral Methods for Dimensionality Reduction. In *Proceedings of the 23rd International Conference on Machine Learning (ICML 2006)*, pages 1041–1048.

Ye, Y. (1997). *Interior Point Algorithms: Theory and Analysis.* Wiley–Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Inc., New York.

Young, G. and Householder, A. S. (1938). Discussion of a Set of Points in terms of Their Mutual Distances. *Psychometrika*, 3(1):19–22.