New Convex Relaxations for the Maximum Cut and VLSI Layout Problems

by

Miguel Nuno Ferreira Fialho dos Anjos

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

Waterloo, Ontario, Canada, 2001

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

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

Abstract

It is well known that many of the optimization problems which arise in applications are "hard", which usually means that they are NP-hard. Hence much research has been devoted to finding "good" relaxations for these hard problems. Usually a "good" relaxation is one which can be solved (either exactly or within a prescribed numerical tolerance) in polynomial-time. Nesterov and Nemirovskii showed that by this criterion, many convex optimization problems are good relaxations. This thesis presents new convex relaxations for two such hard problems, namely the Maximum-Cut (Max-Cut) problem and the VLSI (Very Large Scale Integration of electronic circuits) layout problem.

We derive and study the properties of two new strengthened semidefinite programming relaxations for the Max-Cut problem. Our theoretical results hold for every instance of Max-Cut; in particular, we make no assumptions about the edge weights. The first relaxation provides a strengthening of the well-known Goemans-Williamson relaxation, and the second relaxation is a further tightening of the first. We prove that the tighter relaxation automatically enforces the well-known triangle inequalities, and in fact is stronger than the simple addition of these inequalities to the Goemans-Williamson relaxation. We further prove that the tighter relaxation fully characterizes some low dimensional faces of the cut polytope via the rank of its feasible matrices. We also address some practical issues arising in the solution of these relaxations and present numerical results showing the remarkably good bounds computed by the tighter relaxation.

For the VLSI layout problem, we derive a new relaxation by extending the "target distance” concept recently introduced by Etawil and Vannelli. The resulting AR (Attractor-Repeller) model improves on the NLT (Non-linear optimization Layout Technique) method of van Camp et al. by efficiently finding a good initial point for the NLT solver. Because the AR model is not a convex optimization problem, we isolate the source of the non-convexity and thereby derive a convex version of the model. This derivation leads to the definition of certain generalized target distances with an interesting practical interpretation. Finally, we present numerical results that demonstrate the potential of the AR model.

Acknowledgements

> *There is one elementary truth the ignorance of which kills countless ideas and splendid plans: that the moment one definitely commits oneself, then Providence moves too. All sorts of things occur to help one that would never otherwise have occurred. A whole stream of events issues from the decision, raising in one's favour all manner of unforeseen incidents and meetings and material assistance which no man could have dreamed would have come his way. Whatever you can do or dream you can, begin it. Boldness has genius, power and magic in it.*
> *Begin it now.*
>
> – *Johann Wolfgang von Goethe*

The stream of events that led to this thesis started ten years ago, in May 1991, when Henry Wolkowicz agreed to take me on as an NSERC Summer Undergraduate Research Assistant. In the course of my Assistantship, I was truly impressed by his boundless enthusiasm, his supervisory availability, and his everflowing stream of ideas. After seven years of studying and working elsewhere, it was a privilege to return to Waterloo and complete my Ph.D. under his supervision. Thank you, Henry, for the confidence you had in me.

During my work on this thesis, several other researchers showed a keen interest in my work and encouraged me with their helpful questions and suggestions. I wish to thank Anthony Vannelli for introducing me to the VLSI layout problem and for our many inspiring discussions on the subject. I thank Michael Jünger and Frauke Liers for acquainting me with the application of Maximum-Cut to spin glasses and for providing me with the `torusgen` code for generating instances of such structures. I also thank Jim Geelen, Michel Goemans, Christoph Helmberg, Jean

Lasserre, Franz Rendl and Levent Tunçel for their insightful remarks on various parts of this thesis.

The financial assistance of the Fonds pour la Formation de Chercheurs et l'Aide à la Recherche (Fonds FCAR) of Québec, through a three-year Doctoral Research Scholarship, was greatly appreciated. The additional support provided by the Department of Combinatorics & Optimization through a University of Waterloo Graduate Scholarship and a William Tutte Postgraduate Scholarship for 2000 was also very appreciated.

Finally, it was also during my 1991 sojourn in Waterloo that I met a wonderful person who is God's greatest blessing to me. Julie, you know that this thesis would not have been completed without your unfailing love, patience, and support.

Delight yourself in the Lord
And he will give you the desires of your heart.
Commit your way to the Lord;
Trust in him and he will do this.

– *Psalm 37:4-5*

Contents

List of Tables

List of Figures

Chapter 1

Introduction

It is well known that many of the optimization problems which arise in applications are “hard”, which usually means that they are NP-hard. Hence much research has been devoted to finding “good” relaxations for these hard problems. Usually a “good” relaxation is one which can be solved (either exactly or within a prescribed numerical tolerance) in polynomial-time. Nesterov and Nemirovskii [77] showed that by this criterion, many convex optimization problems are good relaxations.

This thesis introduces two new approaches for constructing convex relaxations for hard optimization problems:

1. The construction of tighter semidefinite programming (SDP) relaxations for the Maximum-Cut (Max-Cut) problem, and hence for the general quadratic boolean optimization problem (QBP), via the use of new formulations and a second lifting procedure;

2. A generalization of the attractor-repeller paradigm to placement and layout

problems, such as the VLSI (Very Large Scale Integration of electronic circuits) layout problem.

We provide theoretical and computational evidence showing that these approaches improve on previous methods in the literature.

In this chapter we briefly introduce these two problems as well as some definitions, notation, and well-known results relevant to the material in this thesis. We then conclude with an outline of the structure of the thesis.

1.1 The Max-Cut Problem and Quadratic Boolean Optimization

The Max-Cut Problem (MC) is a discrete optimization problem on undirected graphs with weighted edges. Given such a graph, the problem consists in finding a partition of the set of nodes into two parts so as to maximize the sum of the weights on the edges that are cut by the partition (we say that an edge is cut if it has exactly one end on each side of the partition). In this thesis we will assume that the graph in question is complete (if not, edges of zero weight can be added to complete the graph without changing the problem). Furthermore, we place no restrictions on the edge weights (so, in particular, negative edge weights are permitted). We will show in Chapter 2 that this problem is equivalent to the seemingly more general Quadratic Boolean Problem (QBP), which consists of finding the global maximum (or minimum) of a quadratic function subject to boolean constraints on the variables. Quadratic boolean optimization problems are of interest for applications in circuit layout design and statistical physics (see Barahona et al. [12], Lengauer [68] for example). An application of Max-Cut to the study of spin glasses is presented in Section 3.3.

As a result of the celebrated work of Goemans and Williamson [41], the Max-Cut problem has become the flagship problem for studying applications of semidefinite programming to combinatorial optimization [40, 85, 38, 39, 50, 44, 49, 14, 22, 47, etc.]. Furthermore, the Max-Cut problem is closely related to the so-called cut polytope, an important and well-known structure in the area of integer programming. The book by Deza and Laurent [30] presents a wealth of theoretical results about the cut polytope, and some connections to general integer programming are elaborated in Helmberg et al. [48].

It is well known that Max-Cut is an NP-hard problem (see Karp [58]) and that it remains NP-hard for some restricted versions, see Garey and Johnson [37] for example. As we have already mentioned, much research has focused on finding relaxations of the Max-Cut problem. The next section presents several such relaxations.

1.2 Convex Relaxations of the Max-Cut Problem

We will derive new convex relaxations for the Max-Cut problem using semidefinite programming (SDP). Several polyhedral and semidefinite relaxations for Max-Cut have been studied in the literature [8, 9, 10, 19, 18, 20, 43, 89, 90, 64, 65, 62, 66, 67, 83, 28, 82, 81, 59, 71, 86, 87, etc.]. A copositive relaxation for general quadratic programming, which includes the QBP, was proposed in Quist et al. [84]. However, it is NP-hard in general to optimize a linear function over this relaxation.

We are particularly interested in relaxations obtained via procedures that involve some form of lifting and projecting of variables between spaces of varying dimensions. Lift-and-project procedures that find polyhedral relaxations for $\{0, 1\}$ programs have been studied by several authors, e.g. Balas et al. [9, 10, 8], Sherali and Adams [86, 87], and Lovász and Schrijver [71], whose procedure was generalized by Kojima and Tunçel [59].

Lovász and Schrijver [71] define a procedure, denoted N_+, that can be iterated to obtain tighter and tighter semidefinite relaxations of the convex hull of feasible integer points for $\{0, 1\}$ programs. A key result is that iterating the N_+ procedure d times, where d is the number of integer variables in the problem, yields exactly the convex hull of all the integer points. For Max-Cut, d equals the number of edges with non-zero weight and this convex hull is the aforementioned cut polytope. Recently, Laurent [62] studied the relaxation $N_+(M_n)$, where M_n denotes the metric polytope formally defined in Section 2.2, and its relationship to the SDP relaxations which we derive in Chapter 2.

Another way to obtain SDP relaxations is via the application of the theory of moments and its dual theory, the representation of strictly positive polynomials over compact sets. The recent work of Lasserre [61] introduces a family of semidefinite programming relaxations corresponding to liftings of polynomial boolean problems into higher and higher dimensions. Lasserre presents necessary and sufficient conditions under which the optimal value of Max-Cut is attained after a finite number of such liftings. A related SDP relaxation for Max-Cut which is also in a higher dimension was recently proposed by Parrilo [80].

Yet another way to derive these relaxations is through Lagrangian duality (see Poljak et al. [83], Lemaréchal and Oustry [67] for example). In this approach one takes a formulation of Max-Cut and forms its Lagrangian dual. The dual of the dual yields an SDP relaxation, denoted by SDP1 in this thesis. It is equivalent to the Shor relaxation [88] and the S-procedure of Yakubovitch [95, 96]. One advantage of this approach over the two previously mentioned is that it is possible to choose the (possibly redundant) constraints that are included in the primal problem formulation. This choice determines the structure and properties of the resulting SDP relaxation. This is the approach we will use to derive new strengthened SDP relaxations in Chapter 2.

1.3 Symmetric Matrices and Semidefinite Programming

The SDP relaxations are defined over the space $\mathcal{S}^n$ of $n \times n$ real symmetric matrices. This space has dimension $t(n) := n(n+1)/2$ and inner product $\langle A, B\rangle = \operatorname{trace} A\,B$. We let e denote the vector of ones (all vectors will be column vectors) and $E = e\,e^T$ denote the matrix of all ones; their dimensions will be clear from the context.

We also let $e_i \in \Re^m$ denote the i^{th} unit vector in $\Re^m$ and define the elementary matrices

$$E_{ij} := \frac{1}{2}(e_i\,e_j^T + e_j\,e_i^T).$$

For any vector $v \in \Re^m$, we let $\|v\| := \sqrt{v^T v}$ denote the ℓ_2 norm of v. We let $A \circ B$

denote the Hadamard (elementwise) matrix product.

We will consider matrices in the space $\mathcal{S}^{t(n)+1}$. If $Y \in \mathcal{S}^{t(n)+1}$, we index the rows and columns of Y by $0, 1, \ldots, t(n)$. We will be particularly interested in the vector x obtained from the first (0^{th}) row (or column) of Y with the first element dropped. Thus, in our notation, $x = Y_{1:t(n),0}$.

1.3.1 Linear Operators on Symmetric Matrices

We will sometimes make use of operator notation and operator adjoints. The adjoint of a linear operator $\mathcal{A} : X \to Y$ is denoted $\mathcal{A}^*$ and satisfies (by definition)

$$\langle \mathcal{A}\, x, y \rangle = \langle x, \mathcal{A}^* y \rangle \quad \forall x \in X, \forall y \in Y.$$

Given a matrix $S \in \mathcal{S}^n$, we now define several useful operators. The operator $\operatorname{diag}(S)$ returns a vector containing the entries on the diagonal of S. Given $v \in \Re^n$, the operator $\operatorname{Diag}(v)$ returns an $n \times n$ diagonal matrix with the vector v on the diagonal. It is straightforward to check that Diag is the adjoint operator of diag.

The operator vec applied to an $n \times n$ matrix forms a vector of length n^2 by stacking the columns of the matrix argument. The operator Mat is the inverse of vec. Other operators that will be used in this thesis are the symmetric vectorizing operator svec and its inverse operator sMat. The operator svec satisfies $s = \operatorname{svec}(S) \in \Re^{t(n)}$ where s is formed column-wise from S and the strictly lower triangular part of S is ignored. Its inverse is the operator sMat, so

$$S = \operatorname{sMat}(s) \Leftrightarrow s = \operatorname{svec}(S).$$

Note that the adjoint of sMat is not svec, but rather the operator dsvec which acts like svec except that the off-diagonal elements are multiplied by 2. We will also make use of the compound operators

$$\mathrm{vsMat}\,(x) := \mathrm{vec}\,(\mathrm{sMat}\,(x)) \quad \text{and} \quad \mathrm{sdiag}\,(x) := \mathrm{diag}\,(\mathrm{sMat}\,(x)).$$

Their adjoints are

$$\mathrm{vsMat}^*(x) = \mathrm{dsvec}\,\left[\frac{1}{2}\left(\mathrm{Mat}\,(x) + \mathrm{Mat}\,(x)^T\right)\right].$$

and

$$\mathrm{sdiag}^*(x) = \mathrm{dsvec}\,\mathrm{Diag}\,(x).$$

1.3.2 Positive Semidefinite Matrices

An $n \times n$ real symmetric matrix A is said to be *positive semidefinite* if

$$x^T A\, x \geq 0 \quad \text{for all } x \in \Re^n. \tag{1.1}$$

We denote positive semidefiniteness of A by $A \succeq 0$. If inequality (1.1) holds strictly for all non-zero $x \in \Re^n$, the matrix A is said to be *positive definite*. Equivalently, A is positive semidefinite if all its eigenvalues are non-negative, and it is positive definite if all its eigenvalues are positive. The set of positive semidefinite matrices in $\mathcal{S}^n$ form a cone, denoted by $\mathcal{P}_n$. We will omit the subscript when the dimension of the underlying space is clear from context.

We will also state and prove results involving ranks of matrices. The following facts about rank will be useful (see for example Horn and Johnson [51]):

The following statements about a given $m \times n$ matrix A are all equivalent:

1. $\operatorname{rank} A = k$;
2. There exist k, and no more than k, rows of A that constitute a linearly independent set;
3. There exist k, and no more than k, columns of A that constitute a linearly independent set;
4. There is a $k \times k$ submatrix of A with non-zero determinant, but all $(k+1) \times (k+1)$ submatrices of A have determinant 0;
5. The dimension of the range of A is k.

Furthermore, the following hold:

1. If rows and/or columns are deleted from A, then the rank of the resulting submatrix cannot be greater than the rank of A;
2. $\operatorname{rank}(A\,B) \leq \min\{\operatorname{rank}(A), \operatorname{rank}(B)\}$ whenever the product $A\,B$ is defined;
3. $\operatorname{rank}(A+B) \leq \operatorname{rank}(A) + \operatorname{rank}(B)$ whenever the sum $A+B$ is defined.

We will regularly use the *principal submatrices* of a given positive semidefinite matrix. These are defined as follows: Given $A \in \mathcal{S}^n$, for every subset $\{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$, the principal submatrix of A corresponding to this subset is the $k \times k$ submatrix obtained by deleting the $n-k$ rows and columns of A

whose indices are not in the subset. A *leading* principal submatrix is a principal submatrix corresponding to a subset of the form $\{1, 2, \ldots, l\}$. The determinant of a (leading) principal submatrix is a *(leading) principal minor*. We will make use of the following well-known results (see for example Horn and Johnson [51]):

1. Every principal submatrix of a positive semidefinite matrix is positive semidefinite.

2. Every principal submatrix of a positive definite matrix is positive definite.

3. If A is positive definite, then all principal minors of A are positive.

4. Any non-negative linear combination of positive semidefinite matrices is positive semidefinite.

5. For any matrix $C \in \mathcal{S}^n$, if A is positive semidefinite then $C^T A C$ is also positive semidefinite.

This last fact means that positive semidefiniteness is unaffected by congruences. One well-known congruence that will be useful in this thesis involves the so-called *Schur complement*. Suppose that we partition A as:

$$A = \begin{pmatrix} A_1 & A_2 \\ A_2^T & A_3 \end{pmatrix}$$

where the principal submatrices A_1 and A_3 are square and A_1 is positive definite. Then $A \succeq 0$ if and only if $A_3 - A_2^T A_1^{-1} A_2 \succeq 0$.

The matrix $A_3 - A_2^T A_1^{-1} A_2$ is the Schur complement of A_1 in A. We will mostly be interested in applying the above theorem with A_1 having dimension 1, that is, $A_1 = A_{1,1}$.

1.3.3 Semidefinite Programming

Semidefinite programming problems are optimization problems of the form

$$\begin{array}{ll} \max & < Q, X > \\ \text{s.t.} & \mathcal{A}(X) = b \\ & X \succeq 0 \\ & X \in \mathcal{S}^n \end{array}$$

where $\mathcal{A}(X) = b$ represents a linear operator on the matrix X.

From a theoretical point of view, given a semidefinite programming problem, we can find in polynomial-time an approximate solution to within any (fixed) accuracy using interior-point methods. This follows from the seminal work of Nesterov and Nemirovskii [77]. They also implemented the first interior-point method for SDP in [75]. Independently, Alizadeh extended some of the interior-point polynomial-time algorithms from linear programming to SDP and studied applications to discrete optimization [2, 3]. The recent Handbook of Semidefinite Programming by Wolkowicz et al. [94] gives an up-to-date treatment of theory and algorithms.

1.4 The VLSI Layout or Floorplanning Problem

The VLSI (Very Large Scale Integration of electronic circuits) layout problem, or floorplanning problem, consists in finding the optimal positions for a given set of modules of *fixed* area (but perhaps varying dimensions) within a facility. The objective is to minimize the distances between pairs of modules that have nonzero connection "costs". If the modules have varying dimensions, then finding their optimal shapes is also a part of the problem.

This is indeed a hard problem; even the restricted version where the shapes of the modules are fixed and the optimization is taken over a fixed finite set of possible module locations is NP-hard. (This restriction is known as the Quadratic Assignment Problem (QAP), see for example Pardalos and Wolkowicz [79].) For this reason, most of the approaches in the literature are based on heuristics.

Recently, Etawil and Vannelli [32] introduced a "target distance" model for the placement of uniformly sized modules. One important feature of this target distance model is its convexity and indeed the results reported in [32] show that the convex model significantly outperforms heuristic-based approaches in terms of both quality and efficiency. The model of Etawil and Vannelli is based on the application of an attractor-repeller model to the placement problem. For this purpose, one interprets the usual quadratic cost function as an "attractor" which wants to cluster all the modules together so as to minimize the total cost. This undesired configuration is made unrealizable by the introduction of "repeller" terms which keep the modules from clustering and therefore achieve a placement where all the modules are spread out.

1.5 Structure of this Thesis

Armed with the preliminaries set out in this chapter, we proceed in Chapter 2 to present some well-known formulations of Max-Cut and then to derive some new formulations. Following this we focus on relaxations of Max-Cut and use the new formulations to derive strengthened SDP relaxations. We then discuss some of the properties of these relaxations, including their polynomial-time solvability and their strengthening properties with respect to previous relaxations.

In Chapter 3 we examine the structure of the SDP relaxations more closely and construct a projection onto the minimal face of the positive semidefinite cone where the so-called Slater's strict feasibility condition holds for both the primal and the dual problems. This condition guarantees strong duality and hence the efficiency of practical interior-point methods. We then perform a variety of numerical experiments to compare the various Max-Cut relaxations and to explore the properties of the new relaxations. We consider small graphs with special structure, randomly generated graphs, and structured graphs arising from the application of Max-Cut to the study of Ising spin glass models in statistical physics.

The computational strength exhibited by the tightest of the SDP relaxations, denoted SDP3, is the motivation for our study of the geometry of its feasible set in Chapters 4 and 5. In Chapter 4 we show that SDP3 completely captures the geometry of the cut polytope for the complete graph on three nodes via the ranks of its feasible matrices. Then in Chapter 5 we show that this characterization via the rank of feasible matrices generalizes to some of the low dimensional faces of the cut polytope of the complete graph with four or more nodes. These results shed some light on the remarkable computational behaviour of the SDP3 relaxation.

We then proceed to derive a new convex relaxation for the VLSI layout problem in Chapter 6. Our contribution is the AR (Attractor-Repeller) model which is designed to improve upon the 3-stage approach of van Camp et al. [26]. We generalize the target distance concept of Etawil and Vannelli so that the varying areas of the modules as well as the other requirements of the floorplanning problem can be accommodated within an attractor-repeller framework. The resulting AR model replaces Stages 1 and 2 by a single mathematical model that finds a “good” initial point for the solver of the van Camp et al. floorplanning model. However, it turns out that the AR model is not a convex problem, and so in Section 6.3 we isolate the source of the non-convexity and thereby derive a convex version of the model, denoted CoAR.

Chapter 7 presents the results of our computational experiments with the AR model on macro-cell layout and facility planning problems. Since the constraints of the AR model are all linear, the model can be solved by applying a reduced-gradient approach combined with a quasi-Newton algorithm. This is a significant advantage for the AR model because in general we can expect the quasi-Newton algorithm to be superlinearly convergent. We used the implementation of this algorithm in MINOS 5.3 by Murtagh and Saunders [73, 1] Our computational results show the potential of the AR model for floorplanning problems.

Finally, in Chapter 8 we summarize the contributions of this thesis and outline some promising directions for future research.

Chapter 2

Formulations of Max-Cut and New Semidefinite Relaxations

We begin this chapter by presenting some well-known formulations of Max-Cut (MC) and also deriving some new formulations. We then introduce some well-known relaxations of the Max-Cut problem that will be of particular relevance to our work. After this introductory material, we derive a first strengthened relaxation and prove some of its properties in Section 2.4. A second, tighter strengthened relaxation is derived in Section 2.5.

2.1 Formulations of the Max-Cut Problem

Let the given graph G have node set $\{1, \ldots, n\}$ and let it be described by its weighted adjacency matrix $A = (w_{ij})$. Let $L := \mathrm{Diag}\,(Ae) - A$ denote the Laplacian matrix associated with the graph, where the linear operator Diag returns a diagonal matrix with its diagonal formed from the vector given as its argument, and e denotes the vector of all ones. Let the vector $v \in \{\pm 1\}^n$ represent any cut in the graph via the interpretation that the sets $\{i : v_i = +1\}$ and $\{i : v_i = -1\}$ form a partition of the node set of the graph. (It is well known that the formulations over $\{0,1\}^n$ and $\{\pm 1\}^n$ are equivalent, see for example Helmberg et al. [48]. We use the latter since it is usually preferred in the application of SDP to Max-Cut.)

Following Mohar and Poljak [72], we can formulate Max-Cut as:

$$\text{(MC1)} \qquad \begin{array}{rll} \mu^* = & \max & \frac{1}{4} v^T L v \\ & \text{s.t.} & v \in \{\pm 1\}^n \end{array}$$

where here and throughout this thesis μ^* denotes the optimal value of the Max-Cut problem.

It is straightforward to check that

$$\frac{1}{4} v^T L v = \sum_{i<j} w_{ij} \left(\frac{1 - v_i v_j}{2} \right)$$

and that the term multiplying w_{ij} in the sum equals one if the edge (i,j) is cut, and zero otherwise.

Further observe that $v \in \{\pm 1\}^n \Leftrightarrow v_i^2 = 1, i = 1, \ldots, n$. This immediately yields our second formulation for Max-Cut:

$$\text{(MC2)} \qquad \begin{array}{rll} \mu^* = & \max & \frac{1}{4} v^T L v \\ & \text{s.t.} & v_i^2 = 1, i = 1, \ldots, n. \end{array}$$

MC2 is a well-known and commonly used formulation for Max-Cut.

We now show that MC1 is equivalent to the seemingly more general quadratic boolean problem:

$$\text{(QBP)} \qquad \begin{array}{rll} \mu^* = & \max & v^TQv - 2c^Tv \\ & \text{s.t.} & v \in \{\pm 1\}^n. \end{array}$$

Clearly, MC1 corresponds to the choice $Q = \frac{1}{4}L$ and $c = 0$. Conversely, we interpret MC1 as the problem of maximizing a homogeneous quadratic function of v over the set $\{\pm 1\}^n$. We show that this problem is equivalent to QBP. (This equivalence shows that all the results about Max-Cut extend to QBP.) This is done by *homogenizing* QBP and thereby increasing the dimension by one. Indeed, given $v^TQv - 2c^Tv$, define the $(n+1) \times (n+1)$ matrix Q^c obtained by adding a 0^{th} dimension to Q and placing the vector $-c$ in the new row and column, so that

$$Q^c := \begin{bmatrix} 0 & -c^T \\ -c & Q \end{bmatrix}.$$

If we consider the variable $\bar{v} = \begin{pmatrix} v_0 \\ v \end{pmatrix} \in \{\pm 1\}^{n+1}$ and the new quadratic form

$$\bar{v}^T Q^c \bar{v} = v^TQv - 2v_0(c^Tv),$$

then we get an equivalent Max-Cut problem:

$$\begin{array}{rll} \mu^* = & \max & \bar{v}^T Q^c \bar{v} \\ & \text{s.t.} & \bar{v} \in \{\pm 1\}^{n+1}. \end{array}$$

The equivalence to QBP follows from the observation that if $v_0 = 1$ at optimality, then v is optimal for QBP, whereas if $v_0 = -1$ at optimality, then $-v$ if optimal for QBP. A similar homogenization of the linear terms to make them quadratic will be used in Section 2.4.1 to derive the first strengthened SDP relaxation.

We will work extensively with the Max-Cut problem and its relaxations expressed in the space $\mathcal{S}^n$. For this purpose, consider the change of variable $X = vv^T, v \in \{\pm 1\}^n$. Then $v^T Q v = \operatorname{trace} QX$, and a well-known equivalent formulation for Max-Cut is

$$\text{(MC3)} \qquad \begin{array}{rll} \mu^* = & \max & \operatorname{trace} QX \\ & \text{s.t.} & \operatorname{diag}(X) = e \\ & & \operatorname{rank}(X) = 1 \\ & & X \succeq 0, X \in \mathcal{S}^n, \end{array}$$

where diag denotes the linear operator that returns a vector containing the diagonal elements of its matrix argument.

Having derived MC3, we will obtain yet another formulation in $\mathcal{S}^n$ by applying Theorem 2.1.1. This theorem was first proved in Anjos and Wolkowicz [5]. We present a simpler proof communicated to us by Tunçel [92].

Theorem 2.1.1 *Let X be an $n \times n$ symmetric matrix. Then*

$$X \succeq 0, X \in \{\pm 1\}^{n \times n} \quad \textit{if and only if} \quad X = v\,v^T, \ \textit{for some} \ \ v \in \{\pm 1\}^n.$$

Proof: Write $X = \begin{pmatrix} 1 & x^T \\ x & \bar{X} \end{pmatrix}$ and apply the Schur complement to obtain

$$X \succeq 0 \quad \Leftrightarrow \quad \bar{X} - x\,x^T \succeq 0.$$

Now $X \in \{\pm 1\}^{n\times n}$ and $X \succeq 0$ implies $\operatorname{diag}(X) = e$. Therefore $\operatorname{diag}(\bar{X} - x\,x^T) = 0$ and the positive semidefiniteness implies $\bar{X} - x\,x^T = 0$. Thus $\bar{X} = x\,x^T$ and $X = \begin{pmatrix} 1 \\ x \end{pmatrix} \cdot \begin{pmatrix} 1 \\ x \end{pmatrix}^T$, as desired. ■

Theorem 2.1.1 shows that we can replace the requirement that the rank of X be equal to one by the ±1 constraint on the elements of X. This yields our fourth formulation of Max-Cut:

$$\text{(MC4)} \qquad \begin{array}{rrl} \mu^* = & \max & \frac{1}{4}\operatorname{trace} LX \\ & \text{s.t.} & \operatorname{diag}(X) = e \\ & & X \circ X = E \\ & & X \succeq 0, X \in \mathcal{S}^n\,, \end{array}$$

where $\circ$ denotes the Hadamard (elementwise) product of matrices and E denotes the matrix of all ones.

To obtain our next formulation, we add to MC4 the quadratic matrix constraint

$$X^2 - nX = 0.$$

The validity of the constraint $X^2 - nX = 0$ follows from the observation that $X^2 = vv^Tvv^T$ and $v^Tv = n$ for all $v \in \{\pm 1\}^n$.

Our motivation for adding this constraint is that we will be using Lagrangian duality to obtain new SDP relaxations. To obtain tighter bounds we therefore want the duality gap incurred in the process to be as small as possible. The interest in this quadratic constraint comes from the results in Anstreicher et al. [6, 7] where the addition of such (redundant) constraints was shown to guarantee strong duality for certain problems where duality gaps can exist.

Furthermore, we now show that after adding this constraint, we can remove certain other constraints and obtain a formulation with only linear and quadratic equality constraints. Indeed, since we can simultaneously diagonalize X and X^2, the eigenvalues of X must satisfy $\lambda^2 - n\lambda = 0$, which implies that the only eigenvalues of X are 0 and n. This shows that the constraint $X \succeq 0$ becomes redundant and may be removed. Moreover, since the diagonal constraint implies that the trace of X is n, we conclude that X must be rank-one and the rank constraint can also be removed.

The resulting problem MC5 is thus another formulation of Max-Cut:

$$\text{(MC5)} \qquad \begin{array}{rcl} \mu^* = & \max & \operatorname{trace} QX \\ & \text{s.t.} & \operatorname{diag}(X) = e \\ & & X \circ X = E \\ & & X^2 - nX = 0. \end{array}$$

We will make use of one more formulation in $\mathcal{S}^n$. This formulation is obtained by adding more quadratic constraints to MC5. Recall the change of variable $X = vv^T$. Since $X_{ij} = v_i v_j$ and $v_k^2 = 1$ for $k = 1, \ldots, n$, the constraints

$$X_{ij} = v_i v_j = v_i v_k^2 v_j = v_i v_k \cdot v_k v_j = X_{ik} \cdot X_{kj}$$

also hold for every rank-one X corresponding to a cut. Adding these constraints to MC5, we obtain the formulation MC6:

$$
\text{(MC6)} \qquad
\begin{array}{rll}
\mu^* = & \max & \operatorname{trace} QX \\
& \text{s.t.} & \operatorname{diag}(X) = e \\
& & X \circ X = E \\
& & X^2 - nX = 0 \\
& & X_{ij} = X_{ik}X_{kj}, \quad \forall\, 1 \le i, j, k \le n.
\end{array}
$$

2.2 Well-Known Relaxations of Max-Cut

In this section we introduce some notation for well-known relaxations of the feasible set of the Max-Cut problem.

The smallest convex set containing all the matrices X which are feasible for MC5 is their convex hull, called the *cut polytope*:

$$C_n := \operatorname{conv}\{X : X = vv^T, v \in \{\pm 1\}^n\}.$$

Optimizing trace QX over C_n would yield exactly μ^*, but an efficient description of the cut polytope is not known.

A well-known relaxation of the cut polytope is the *metric polytope* M_n, defined as the set of all matrices satisfying the triangle inequalities:

$$
\begin{aligned}
M_n := \{X \in \mathcal{S}^n \ &: \operatorname{diag}(X) = e, \text{ and} \\
& X_{ij} + X_{ik} + X_{jk} \ge -1, X_{ij} - X_{ik} - X_{jk} \ge -1, \\
& - X_{ij} + X_{ik} - X_{jk} \ge -1, -X_{ij} - X_{ik} + X_{jk} \ge -1, \\
& \forall\, 1 \le i < j < k \le n\}.
\end{aligned}
$$

The triangle inequalities model the following constraints: for any three mutually connected nodes in the graph, it is only possible to cut either zero or two of the edges joining them. In fact, the triangle inequalities are sufficient to describe the cut polytope for graphs with less than five nodes, i.e. $C_n = M_n$ for $n \leq 4$; however, $C_n \subsetneq M_n$ for $n \geq 5$, see for example Deza and Laurent [30]. Furthermore, if no subgraph of G induced by edges with non-zero weights is contractible to K_5, the complete graph on 5 nodes, then Barahona [11] proved that the linear programming problem

$$\begin{array}{ll} \max & \operatorname{trace} QX \\ \text{s.t.} & X \in M_n \end{array}$$

has optimal value equal to μ^*.

Denote the feasible set of MC3, less the rank constraint, by

$$\mathcal{E}_n := \{X \in \mathcal{S}^n \ : \operatorname{diag}(X) = e, X \succeq 0\}.$$

The set $\mathcal{E}_n$ is the *elliptope* studied in Laurent and Poljak [64, 65] and is a convex relaxation of the feasible set of formulation MC5 since we dropped the rank constraint. With this notation, let us define the semidefinite programming relaxation SDP1:

$$(\text{SDP1}) \qquad \begin{array}{rcll} \nu_1^* & = & \max & \operatorname{trace} QX \\ & & \text{s.t.} & X \in \mathcal{E}_n. \end{array}$$

This SDP relaxation is well known and has been studied by Delorme and Poljak [29], Goemans and Williamson [41], Laurent et al. [66], and Nesterov [76] among others.

Goemans and Williamson [41] provided estimates for the quality of the SDP1 bound for Max-Cut. They proved that the optimal value of this relaxation is at most 14% above the value of the maximum cut, provided there are no negative edge weights. More precisely,

$$\mu^* \geq \alpha_{GW} \, \nu_1^*,$$

where $\alpha_{GW} := \min_{0 \leq \theta \leq \pi} \frac{2}{\pi} \frac{\theta}{1-\cos\theta} \approx 0.87856$, and since $\frac{1}{\alpha_{GW}} \leq 1.14$, we have that $\nu_1^* \leq 1.14\mu^*$. Furthermore, by randomly rounding a solution to the SDP relaxation, they obtain a α_{GW}-approximation algorithm, i.e. an algorithm that produces a cut with value at least α_{GW} times the optimal value. (Note that Håstad [52] proved that it is NP-hard to find a ρ-approximation algorithm for Max-Cut with ρ greater than 0.9412.)

For the general case of Max-Cut with no restriction on the edge weights, Nesterov [76] recently proved constant relative accuracy estimates for the SDP1 bound. To state Nesterov's result, let us define the quantities μ_*, $(\nu_1)_*$ and $s(\beta)$ as follows:

$$\begin{array}{rll} \mu_* = & \min & v^T Q v \\ & \text{s.t.} & v \in \{\pm 1\}^n, \end{array}$$

$$\begin{array}{rll} (\nu_1)_* = & \min & \operatorname{trace} QX \\ & \text{s.t.} & X \in \mathcal{E}_n, \end{array}$$

and

$$s(\beta) := \beta \, \nu_1^* + (1-\beta)\,(\nu_1)_*, \;\; \beta \in [0,1].$$

(Note that $\mu_* = 0$ in the absence of negative edge weights.) Nesterov [76] proved that without any assumption on the matrix Q,

$$\frac{|s(\frac{2}{\pi}) - \mu^*|}{\mu^* - \mu_*} \leq \frac{\pi}{2} - 1 < \frac{4}{7}.$$

Nesterov's analysis is extended further in Nesterov [78] and Ye [97].

2.3 SDP Relaxations via Lagrangian Relaxation

In Section 2.4 we will derive a new strengthened SDP relaxation of Max-Cut by applying the "recipe" presented in Poljak et al. [83]. For our purposes, this recipe can be summarized as:

1. Choose a formulation of Max-Cut and take its Lagrangian dual;

2. Use the hidden semidefinite constraint in the Lagrangian dual to formulate it as an SDP;

3. Take the dual of this SDP, i.e. the dual of the dual, to obtain the SDP relaxation of Max-Cut.

Whenever the objective function or the constraints in the primal problem contain a linear term, negative semidefiniteness of the Hessian (i.e. the hidden semidefinite constraint) is not sufficient for boundedness of the quadratic; feasibility of the stationarity condition is also needed. Alternatively, one can replace the linear constraints by norm constraints, homogenize and use strong duality of the trust region subproblem (see Stern and Wolkowicz [91]). The latter technique is used in Section 2.4.1.

2.3.1 First Lifting via Lagrangian Relaxation

In this section we illustrate the above recipe by sketching the details of the derivation of the relaxation SDP1 starting with the formulation MC2 and using Lagrangian duality.

The Lagrangian dual of MC2 can be written as

$$\mu^* \leq \nu_1^* = \min_y \; \max_v v^T Q v - v^T (\operatorname{Diag} y) v + e^T y.$$

The inner maximization has a hidden constraint, that is, the quadratic is bounded above only if its Hessian is negative semidefinite. This is equivalent to the following SDP:

$$\begin{array}{rcll} \nu_1^* & = & \min & e^T y \\ & & \text{s.t.} & \operatorname{Diag} y \succeq Q. \end{array}$$

Slater's (strict feasibility) constraint qualification holds for this problem. Therefore its Lagrangian dual has the same optimal value and is precisely SDP1:

$$\begin{array}{rcll} \mu^* & \leq & \nu_1^* = & \max & \operatorname{trace} QX \\ & & & \text{s.t.} & \operatorname{diag}(X) = e \\ & & & & X \succeq 0. \end{array}$$

2.4 First Strengthened SDP Relaxation

The derivation of SDP2 starts with the formulation MC5:

$$\text{(MC5)} \qquad \begin{array}{rll} \mu^* = & \max & \operatorname{trace} QX \\ & \text{s.t.} & \operatorname{diag}(X) = e \\ & & X \circ X = E \\ & & X^2 - nX = 0. \end{array}$$

We present two different derivations of the strengthened relaxation SDP2. The first derivation follows the recipe in Poljak et al. [83] which was described and illustrated in Section 2.3 while the second derivation is a direct second lifting using the motivation that cuts correspond to rank-one matrices in the strengthened relaxation. Although the second derivation is simpler and can be done independently, we also include the first one because it gives insight into how the choice of (possibly redundant) constraints determines the SDP relaxation which we obtain from the Lagrangian dual. Furthermore, once the result of the first derivation is obtained, the second derivation of the same SDP becomes obvious. But it is not clear how to directly derive an SDP that has not yet been formulated. (Note that the equivalence of the two derivations follows from Theorem 9 of Poljak et al. [83] and the discussion preceding that theorem.)

2.4.1 Second Lifting via Lagrangian Relaxation

To efficiently apply Lagrangian relaxation and not lose the information from the linear constraint, we need to replace that constraint with the norm constraint

$||\mathrm{diag}\,(X) - e||^2 = 0$ and homogenize (see Poljak et al. [83]). We then lift this matrix problem into a higher dimensional matrix space.

To keep the dimension as low as possible, we take advantage of the symmetry of X. Recall that $t(i) = \frac{i(i+1)}{2}$ and that sMat is the linear operator which, given a vector $x \in \Re^{t(n)}$, returns a matrix $X \in \mathcal{S}^n$ obtained by filling in columnwise the upper triangular part of X with the $t(n)$ components of x and completing the strictly lower triangle by symmetry. Thus we rewrite MC5 as

$$\begin{array}{rll} \mu^* = & \max & \mathrm{trace}\,(Q\,\mathrm{sMat}\,(x))\,y_0 \\ & \text{s.t.} & \mathrm{diag}\,(\mathrm{sMat}\,(x))^T\mathrm{diag}\,(\mathrm{sMat}\,(x)) \\ & & \qquad - 2e^T\mathrm{diag}\,(\mathrm{sMat}\,(x))y_0 + n = 0 \\ \text{(MC5)} & & E - \mathrm{sMat}\,(x) \circ \mathrm{sMat}\,(x) = 0 \\ & & \mathrm{sMat}\,(x)^2 - n\,\mathrm{sMat}\,(x)y_0 = 0 \\ & & 1 - y_0^2 = 0 \\ & & x \in \Re^{t(n)}, y_0 \in \Re. \end{array}$$

Note that this problem is equivalent to the previous formulation since we can change X to $-X$ if $y_0 = -1$ is optimal.

We now take the Lagrangian dual of MC5. Introducing Lagrange multipliers

$w, t \in \Re$ and $T, S \in \mathcal{S}^n$, the dual is

$$\begin{aligned}\nu_2^* \;=\; & \min_{t,w,T,S} \max_{x,y_0} \{ \text{trace } (Q\text{sMat}\,(x))\, y_0 \\ & + w(\text{diag}\,(\text{sMat}\,(x))^T \text{diag}\,(\text{sMat}\,(x)) - 2e^T \text{diag}\,(\text{sMat}\,(x)) y_0 + n) \\ & + \text{trace}\, T(E - \text{sMat}\,(x) \circ \text{sMat}\,(x)) \\ & + \text{trace}\, S((\text{sMat}\,(x))^2 - n\, \text{sMat}\,(x) y_0) \\ & + t(1 - y_0^2)\}. \end{aligned} \tag{2.1}$$

Note that moving the constraint $1 - y_0^2$ into the Lagrangian does not increase the duality gap, since the Lagrangian relaxation of the trust-region subproblem is tight (see Stern and Wolkowicz [91]).

We now find the hidden semidefinite constraint in the dual problem (2.1). The inner maximization of (2.1) is an unconstrained pure quadratic maximization, therefore its optimal value is infinity unless the Hessian is negative semidefinite in which case $x = 0$, $y_0 = 0$ is optimal. We proceed to calculate the Hessian and thus find the hidden semidefinite constraint.

Recall the following linear operators:

- svec is the inverse of sMat , i.e. it forms a $t(n)$-vector columnwise from an $n \times n$ symmetric matrix while ignoring the strictly lower triangular part of the matrix;
- dsvec acts like svec but multiplies by 2 the off-diagonal entries of its (symmetric) matrix argument;
- Mat forms an $n \times n$ matrix columnwise from an n^2-vector;

- vec is the inverse of Mat ;

- vsMat (x) := vec (sMat (x)).

Using trace QsMat $(x) = x^T$dsvec (Q), and pulling out a 2 for convenience, we get the constant part (no Lagrange multipliers) of the Hessian:

$$2\,H_Q := 2\begin{pmatrix} 0 & \frac{1}{2}\mathrm{dsvec}\,(Q)^T \\ \frac{1}{2}\mathrm{dsvec}\,(Q) & 0 \end{pmatrix}.$$

The nonconstant part of the Hessian is made up of a linear combination of matrices, i.e. it is a linear operator on the Lagrange multipliers.

To make the quadratic forms in (2.1) easier to differentiate we note that

$$\mathrm{dsvec}\,\mathrm{Diag}\,\mathrm{diag}\,\mathrm{sMat} = \mathrm{sdiag}^*\mathrm{sdiag} \qquad (= \mathrm{Diag}\,\mathrm{svec}\,(I))$$

and rewrite the quadratic forms as follows:

$$\begin{aligned} \mathrm{sdiag}\,(x)^T\mathrm{sdiag}\,(x) &= x^T\,(\mathrm{dsvec}\,\mathrm{Diag}\,\mathrm{diag}\,\mathrm{sMat}\,)\,x; \\ e^T\mathrm{sdiag}\,(x) &= (\mathrm{dsvec}\,\mathrm{Diag}\,e)^T\,x; \end{aligned}$$

$$\begin{aligned} \mathrm{trace}\,S(\mathrm{sMat}\,(x))^2 &= \langle \mathrm{sMat}\,(x)\,, S\,\mathrm{sMat}\,(x)\rangle \\ &= \mathrm{vsMat}\,(x)^T\mathrm{vec}\,(S\,\mathrm{sMat}\,(x)) \\ &= x^T\mathrm{vsMat}^*\mathrm{vec}\,\,(S\mathrm{sMat}\,(x)) \\ &= x^T\,[\mathrm{vsMat}^*\mathrm{vec}\,\,(S\mathrm{sMat}\,)]\,x \\ &= x^T\,[\mathrm{vsMat}^*\mathrm{vec}\,S\mathrm{Mat}\,\mathrm{vsMat}\,]\,x \end{aligned}$$

$$= x^T \left[(\mathrm{Mat}\,\mathrm{vsMat}\,)^* S\,(\mathrm{Mat}\,\mathrm{vsMat}\,)\right] x;$$

$$\begin{aligned}
\operatorname{trace} T(\mathrm{sMat}\,(x) \circ \mathrm{sMat}\,(x)) &= x^T \left\{\mathrm{dsvec}\ (T \circ \mathrm{sMat}\,(x))\right\} \\
&= x^T\,(\mathrm{dsvec}\ (T \circ \mathrm{sMat}\,))\,x,
\end{aligned}$$

where the expression with S involves $\mathrm{vsMat}\,^*\mathrm{vec}$ instead of dsvec because $S\mathrm{sMat}\,(x)$ may not be symmetric. (It is easy to verify that $\mathrm{vsMat}\,^*\mathrm{vec}$ reduces to dsvec if S is symmetric.) However, the expression is still a congruence of S. The last expression follows from the following easy result: If $A, B \in \mathcal{S}^n$, then $\operatorname{trace}\,(A(B \circ C)) = \operatorname{trace}\,(B(A \circ C))$.

For notational convenience, we let $\mathcal{H}(w,T,S,t)$ denote the *negative* of the non-constant part of the Hessian, and we split it into four linear operators with the factor 2:

$$\begin{aligned}
2\mathcal{H}(w,T,S,t) &:= 2\mathcal{H}_1(w) + 2\mathcal{H}_2(T) + 2\mathcal{H}_3(S) + 2\mathcal{H}_4(t) \\
&= 2w \begin{pmatrix} 0 & (\mathrm{dsvec}\,\mathrm{Diag}\,e)^T \\ (\mathrm{dsvec}\,\mathrm{Diag}\,e) & -\mathrm{sdiag}\,^*\mathrm{sdiag} \end{pmatrix} \\
&\quad + 2 \begin{pmatrix} 0 & 0 \\ 0 & \mathrm{dsvec}\ (T \circ \mathrm{sMat}\,) \end{pmatrix} \\
&\quad + 2 \begin{pmatrix} 0 & \frac{n}{2}\mathrm{dsvec}\,(S)^T \\ \frac{n}{2}\mathrm{dsvec}\,(S) & (\mathrm{Mat}\,\mathrm{vsMat}\,)^* S\,(\mathrm{Mat}\,\mathrm{vsMat}\,) \end{pmatrix} \\
&\quad + 2t \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.
\end{aligned}$$

The matrix $\mathrm{sdiag}^*\mathrm{sdiag} \in \mathcal{S}^{t(n)}$ is diagonal with elements determined using

$$\begin{aligned} e_i^T\left(\mathrm{sdiag}^*\mathrm{sdiag}\right)e_j &= \mathrm{sdiag}\,(e_i)^T\mathrm{sdiag}\,(e_j) \\ &= \begin{cases} 1, & \text{if } i = j = t(k), \text{ for some } k \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

Similarly, letting $T = \sum\limits_{i,j} t_{ij}E_{ij}$, we have

$$\mathrm{dsvec}\,(T \circ \mathrm{sMat}\,) = \sum_{i,j} t_{ij}\mathrm{dsvec}\,(E_{ij} \circ \mathrm{sMat}\,).$$

Then for fixed i, j, the matrix $\mathrm{dsvec}\,(E_{ij} \circ \mathrm{sMat}\,)$ has element k, l equal to

$$e_k^T\left[\mathrm{dsvec}\,(E_{ij} \circ \mathrm{sMat}\,(e_l))\right].$$

Similarly, we can find the k, l element of $(\mathrm{Mat}\,\mathrm{vsMat}\,)^*\,S\,(\mathrm{Mat}\,\mathrm{vsMat}\,)$ by cancelling vec and Mat and evaluating

$$e_k^T\mathrm{vsMat}^*\mathrm{vec}\,(S\mathrm{sMat}\,(e_l)).$$

Upon cancelling the factor of 2 on both sides of the constraint, we get the SDP

$$\begin{array}{rll} \nu_2^* = & \min & nw + \mathrm{trace}\,ET + \mathrm{trace}\,0S + t \\ & \text{s.t.} & \mathcal{H}(w, T, S, t) \succeq H_Q. \end{array}$$

If we take T sufficiently positive definite and t sufficiently large, then we can guarantee Slater's constraint qualification. Therefore the dual of this SDP has the same optimal value ν_2^* and it provides a strengthened SDP relaxation of Max-Cut:

$$
\text{(SDP2)} \qquad
\begin{array}{rll}
\nu_2^* = & \max & \operatorname{trace} H_Q Y \\
& \text{s.t.} & \mathcal{H}_1^*(Y) = n \\
& & \mathcal{H}_2^*(Y) = E \\
& & \mathcal{H}_3^*(Y) = 0 \\
& & \mathcal{H}_4^*(Y) = 1 \\
& & Y \succeq 0, Y \in \mathcal{S}^{t(n)+1}.
\end{array}
$$

To express the linear operators $\mathcal{H}_i^*(Y)$, $i = 1, 2, 3, 4$, let us index the rows of Y by $0, 1, \ldots, t(n)$ and partition it as

$$
Y = \begin{pmatrix} Y_{00} & x^T \\ x & \bar{Y} \end{pmatrix},
$$

where $\bar{Y} \in \mathcal{S}^{t(n)}$.

It is straightforward to check that

$$
\mathcal{H}_2^*(Y) = \operatorname{sMat} \operatorname{diag}(\bar{Y}) \quad \text{and} \quad \mathcal{H}_4^*(Y) = Y_{00},
$$

so the constraints $\mathcal{H}_2^*(Y) = E$ and $\mathcal{H}_4^*(Y) = 1$ are equivalent to $\operatorname{diag}(Y) = e$. Also, $\mathcal{H}_1^*(Y)$ is twice the sum of the elements in the first row of Y corresponding to the positions of the diagonal of $\operatorname{sMat}(x)$ minus the sum of the same elements in the diagonal of $\bar{Y}$, i.e.

$$
\mathcal{H}_1^*(Y) = 2\operatorname{svec}(I_n)^T x - \operatorname{trace} \operatorname{Diag}(\operatorname{svec}(I_n))\bar{Y}.
$$

The constraint $\mathcal{H}_1^*(Y) = n$ effectively requires that $Y_{0,t(i)} = 1, \forall\, i = 1, \ldots, n$, as shown in the proof of Lemma 2.4.1 below, and thus $\operatorname{diag}(\operatorname{sMat}(x)) = e$ holds. Finally, to find $\mathcal{H}_3^*(Y)$, recall that by definition

$$\langle \mathcal{H}_3(S), Y \rangle \quad = \quad n \operatorname{dsvec}(S)^T x - \left\langle (\operatorname{Mat} \operatorname{vsMat})^* \, S \, (\operatorname{Mat} \operatorname{vsMat}), \bar{Y} \right\rangle.$$

Taking adjoints,

$$\begin{aligned} \langle S, \mathcal{H}_3^*(Y) \rangle \quad &= \quad \operatorname{trace} S n \operatorname{sMat}(x) - \left\langle S, (\operatorname{Mat} \operatorname{vsMat}) \, \bar{Y} \, (\operatorname{Mat} \operatorname{vsMat})^* \right\rangle \\ &= \quad \left\langle S, n \operatorname{sMat}(x) - (\operatorname{Mat} \operatorname{vsMat}) \, \bar{Y} \, (\operatorname{Mat} \operatorname{vsMat})^* \right\rangle. \end{aligned}$$

Recall that $(\operatorname{Mat} \operatorname{vsMat})^* = \operatorname{vsMat}^* \operatorname{vec}$ is essentially (and in the symmetric case reduces to) dsvec.

The constraint $\mathcal{H}_3^*(Y) = 0$ implies immediately that if Y is feasible for SDP2, then $\operatorname{sMat}(x)$ is positive semidefinite (and in fact feasible for SDP1). This is proved in Lemma 2.4.3.

We now prove that the feasible set of SDP2 has no strictly feasible points.

Lemma 2.4.1 *If Y is feasible for SDP2, then $Y_{0,t(i)} = 1, i = 1, \ldots, n$. Hence Y is not positive definite.*

Proof: Let Y be feasible for SDP2. The constraints $\mathcal{H}_2^*(Y) = E$ and $\mathcal{H}_4^*(Y) = 1$ together imply that $\operatorname{diag}(Y) = e$. The constraint $\mathcal{H}_1^*(Y) = n$ can be written as

$$2 \operatorname{svec}(I_n)^T x - \operatorname{trace} \operatorname{Diag}(\operatorname{svec}(I_n)) \bar{Y} = n,$$

with $Y = \begin{pmatrix} 1 & x^T \\ x & \bar{Y} \end{pmatrix}$. Since $\operatorname{diag}(Y) = e$, $\operatorname{trace}\operatorname{Diag}(\operatorname{svec}(I_n))\bar{Y} = n$ and so $\operatorname{svec}(I_n)^T x = n$, or equivalently $\sum_{i=1}^{n} Y_{0,t(i)} = n$. Now $Y \succeq 0$ implies every principal minor of Y is nonnegative, so $|Y_{0,t(i)}| \leq 1$ must hold (again because $\operatorname{diag}(Y) = e$). So $\sum_{i=1}^{n} Y_{0,t(i)} = n \Rightarrow Y_{0,t(i)} = 1, i = 1, \ldots, n$. Hence each of the 2×2 principal minors obtained from the subsets of rows and columns $\{0, t(i)\}, i = 1, \ldots, n$ equals zero. Hence Y is not positive definite. ■

2.4.2 Direct Second Lifting

We can derive the SDP2 relaxation directly from MC5 using the rank-one relationship

$$Y \cong \begin{pmatrix} y_0 \\ x \end{pmatrix} \begin{pmatrix} y_0 & x^T \end{pmatrix}, \quad X = \operatorname{sMat}(x).$$

Using this approach we express the constraints that the elements of X are ±1 and $\operatorname{diag}(X) = e$ as

$$\operatorname{diag}(Y) = e \quad \text{and} \quad Y_{0,t(i)} = 1, i = 1, \ldots, n.$$

We also express the $t(n+1)$ constraints from $X^2 - nX = 0$. The constraints corresponding to equating the diagonal entries are clearly redundant. After they are removed, the result is the SDP relaxation:

$$
\text{(SDP2)} \qquad
\begin{array}{rll}
\nu_2^* = & \max & \operatorname{trace} H_Q Y \\
& \text{s.t.} & \operatorname{diag}(Y) = e \\
& & Y_{0,t(i)} = 1, i = 1, \ldots, n \\
& & Y_{0,T(i,j)} = \frac{1}{n} \sum_{k=1}^{n} Y_{T(i,k),T(k,j)}, \ \ \forall\, 1 \leq i < j \leq n \\
& & Y \succeq 0, Y \in \mathcal{S}^{t(n)+1},
\end{array}
$$

where

$$
T(i,j) := \begin{cases} t(j-1)+i, \text{ if } i \leq j \\ t(i-1)+j, \ \text{otherwise.} \end{cases}
$$

Recall that $t(i) = \frac{i(i+1)}{2}$, so $T(i,i) = t(i)$.

Remark 2.4.2 *The indices for the linear constraints in SDP2 may be thought of as the entries of a matrix T constructed in the following way. Expanding the relationship $X = \operatorname{sMat}(x)$ we have:*

$$
X = \begin{pmatrix} x_1 & x_2 & x_4 & \ldots \\ x_2 & x_3 & x_5 & \ldots \\ & \ldots & & x_{t(n)} \end{pmatrix}.
$$

We can now define a matrix T by keeping only the indices *of the entries of x:*

$$
T := \begin{pmatrix} 1 & 2 & 4 & \ldots \\ 2 & 3 & 5 & \ldots \\ & \ldots & & t(n) \end{pmatrix}.
$$

The first two sets of constraints imply that the 2×2 leading principal minor of any Y feasible for SDP2 is all ones. This gives us an alternative proof that every feasible Y for SDP2 is singular. It also means that there is still some redundancy in the constraints of SDP2, and in Chapter 3 we will make use of this fact to project the feasible set of SDP2 onto a lower dimensional face of the positive semidefinite cone where Slater's constraint qualification holds. In doing so we will reduce the dimension of the matrix variable, and hence the number of variables in the SDP relaxation.

2.4.3 Properties of the First Strengthened Relaxation

We first note that the relaxation SDP2 can be solved in polynomial-time. If we consider, say, the formulation MC1, we see that the original Max-Cut problem is specified by the elements of the Laplacian matrix L (which is defined in terms of the original edge weights w_{ij}) and by n, the number of nodes in the graph. We first note that the matrix variable has $t(t(n)+1) = O(n^4)$ scalar variables within it, and that the number of linear equality constraints is

$$(t(n)+1) + n + \frac{n(n-1)}{2} = O(n^2).$$

Hence both the number of variables and the number of constraints are polynomial in n. Secondly, we consider the magnitude of the coefficients in the objective function H_Q and the linear constraints. The elements of H_Q are all zeros or elements of $Q = \frac{1}{4}L$, some unchanged and some divided by 2 (due to the action of the operator $\frac{1}{2}\text{dsvec}\,(\cdot)$). Hence if we use, say, rational number representation, then the size of the elements of H_Q is linear with respect to the size of the elements of L. Finally, all the coefficients of the linear constraints are zeros, ones, or $\frac{1}{n}$. Since the latter can be represented with size proportional to n (again using rational representation), we conclude that the data required to represent the relaxation SDP2 is polynomial with respect to the data required to represent the original problem MC1. By the work of Nesterov and Nemirovskii [77], it follows that, given a prescribed accuracy $\epsilon > 0$, we can use interior-point methods to find in polynomial-time an approximate solution Y_ϵ satisfying

$$\operatorname{trace} H_Q Y_\epsilon \leq \nu_2^* + \epsilon.$$

(Polynomial-time means that the total number of arithmetic operations required to compute Y_ϵ is bounded above by a polynomial function of the number of constraints, the number of variables, and the size of the data required to represent SDP2.)

Our next result is that the matrix obtained by applying sMat to the first row of a feasible Y is positive semidefinite, even though this nonlinear constraint was not explicitly included in MC5. The use of operator notation yields a succinct proof of this lemma.

Lemma 2.4.3 *Suppose that Y is feasible for SDP2. Then*

$$\text{sMat}\,\left(Y_{1:t(n),0}\right) \succeq 0$$

and so is feasible for SDP1.

Proof: Using the partition $Y = \begin{pmatrix} 1 & x^T \\ x & \bar{Y} \end{pmatrix}$, we see that $\bar{Y}$ is positive semidefinite. Rewriting the constraint $\mathcal{H}_3^*(Y) = 0$ as

$$\mathrm{sMat}\,(x) = \frac{1}{n}\,(\mathrm{Mat}\,\mathrm{vsMat}\,)\,\bar{Y}\,(\mathrm{Mat}\,\mathrm{vsMat}\,)^*\,,$$

we see that sMat (x) is a congruence of $\bar{Y}$. The result follows. ■

Consequently, the relaxation SDP2 is a strengthening of SDP1.

Theorem 2.4.4 *The optimal values of SDP1 and SDP2 satisfy*

$$\nu_2^* \leq \nu_1^*.$$

Proof: Suppose that

$$Y^* = \begin{pmatrix} 1 & x^{*T} \\ x^* & \bar{Y}^* \end{pmatrix}$$

solves SDP2. From Lemma 2.4.3, $X^* := \mathrm{sMat}\,(x^*)$ is feasible for SDP1, therefore

$$\begin{array}{rcl} \nu_2^* & = & \mathrm{trace}\, H_Q Y^* \\ & = & (\mathrm{dsvec}\, Q)^T x^* \\ & = & \mathrm{trace}\, Q X^* \\ & \leq & \nu_1^*. \end{array}$$

■

2.5 Second Strengthened SDP Relaxation

The derivation of SDP3, the second strengthened SDP relaxation, starts with the formulation MC6:

$$\text{(MC6)} \qquad \begin{array}{rll} \mu^* = & \max & \operatorname{trace} QX \\ & \text{s.t.} & \operatorname{diag}(X) = e \\ & & X \circ X = E \\ & & X^2 - nX = 0 \\ & & X_{ij} = X_{ik}X_{kj}, \quad \forall\, 1 \leq i,j,k \leq n. \end{array}$$

The dual of MC6 will yield a relaxation tighter than SDP2 because an increase in the number of constraints in the primal problem implies a greater number of Lagrange multipliers in the dual. This gives us a smaller duality gap and hence a better bound. Furthermore, there is an interesting connection between the constraints

$$X_{ij} = X_{ik}X_{kj}, \quad \forall\, 1 \leq i,j,k \leq n$$

and the metric polytope. This connection is first observed in the statement and proof of Theorem 2.5.4 and will also be exploited in Chapter 4. Taking the dual of the dual of MC6 (and removing redundant constraints in the resulting SDP) yields the strengthened relaxation SDP3 defined below.

Alternatively, we can motivate SDP3 by considering that the rank-one matrices $X = vv^T, v \in \{\pm 1\}^n$ have all their entries equal to ± 1. Hence the corresponding matrices Y feasible for SDP2 have all their entries in the first row and column equal to ±1. Now consider the following constraints from SDP2:

$$Y_{0,T(i,j)} = \frac{1}{n}\sum_{k=1}^{n} Y_{T(i,k),T(k,j)}, \ \ \forall\, 1 \le i < j \le n,$$

for $Y = \begin{pmatrix} 1 & x^T \\ x & \bar{Y} \end{pmatrix}$ and $x = \operatorname{svec}(vv^T)$. The entry $Y_{0,T(i,j)}$ is in the first row of Y and therefore it is equal to 1 in magnitude. The corresponding constraint states that it must be equal to the average of n entries in the block $\bar{Y}$. But each of these n entries has magnitude at most 1, so for equality to hold, they must all have magnitude equal to 1, and in fact they must all equal $Y_{0,T(i,j)}$.

Either approach yields the relaxation SDP3, which after the removal of clearly redundant constraints may be written as:

$$\text{(SDP3)} \qquad \begin{array}{rll} \nu_3^* = & \max & \operatorname{trace} H_Q Y \\ & \text{s.t.} & \operatorname{diag}(Y) = e \\ & & Y_{0,t(i)} = 1, i = 1, \ldots, n \\ & & Y_{0,T(i,j)} = Y_{T(i,k),T(k,j)}, \ \ \forall\, k, \forall\, 1 \le i < j \le n \\ & & Y \succeq 0, Y \in \mathcal{S}^{t(n)+1}. \end{array}$$

We observe that since $Y_{0,t(i)} = 1$, $i = 1, \ldots, n$, it follows (as in Lemma 2.4.1 for SDP2) that SDP3 has no strictly feasible points.

We also observe that SDP3 is solvable in polynomial-time. Indeed, the arguments applied to SDP2 mostly carry through unchanged, except for two aspects related to the modified constraints. The first is the fact that the number of linear constraints is not $O(n^2)$, but rather

$$(t(n)+1)+n+n\cdot\frac{n(n-1)}{2}=O(n^3),$$

which remains polynomial in n. The second is that the coefficients of the linear constraints of SDP3 are all zeros and ones, that is, there are no longer any coefficients equal to $\frac{1}{n}$. Therefore we again conclude that the data required to represent SDP3 is polynomial with respect to the data required to represent the original Max-Cut problem and hence that an approximate solution of SDP3 to within any (fixed) accuracy can be found in polynomial-time using interior-point methods [77].

Remark 2.5.1 *We point out that for* $n=3$*, SDP2 and SDP3 are* equivalent. *Indeed, let us consider SDP2 and recall that if* Y *is feasible for SDP2, then* $Y_{0,t(i)}=1, i=1,2,3$ *(by Lemma 2.4.1). Recall also that* $T(i,i)=t(i)$ *and consider the* 3×3 *principal submatrix of* Y *corresponding to rows and columns* $\{0,T(k,k),T(i,j)\}$*, for fixed* i,j *such that* $1\le i<j\le 3$ *and for any* $k=1,2,3$*. Since* $Y\succeq 0$ *implies this submatrix is also positive semidefinite, we have*

$$\begin{pmatrix} 1 & 1 & Y_{0,T(i,j)} \\ 1 & 1 & Y_{T(k,k),T(i,j)} \\ Y_{0,T(i,j)} & Y_{T(k,k),T(i,j)} & 1 \end{pmatrix}\succeq 0 \Rightarrow Y_{T(k,k),T(i,j)}=Y_{0,T(i,j)}.$$

This is easily verified. The general case of this implication is stated and proved in Chapter 4 as Lemma 4.4.2.

Consider now the pair $i = 1, j = 2$. The implication above shows that

$$Y_{0,T(1,2)} = Y_{T(1,1),T(1,2)} \quad \text{and} \quad Y_{0,T(1,2)} = Y_{T(2,2),T(1,2)} = Y_{T(1,2),T(2,2)}.$$

Hence two of the three constraints of SDP3 for the given i, j also hold for SDP2. Now the sum constraint of SDP2 implies

$$\begin{aligned}
Y_{0,T(1,2)} &= \frac{1}{3}\sum_{k=1}^{3} Y_{T(1,k),T(k,2)} \\
&= \frac{1}{3}\left(Y_{T(1,1),T(1,2)} + Y_{T(1,2),T(2,2)} + Y_{T(1,3),T(3,2)}\right) \\
&= \frac{1}{3}\left(Y_{0,T(1,2)} + Y_{0,T(1,2)} + Y_{T(1,3),T(3,2)}\right)
\end{aligned}$$

and therefore $Y_{0,T(1,2)} = Y_{T(1,3),T(3,2)}$ holds, which is precisely the third constraint of SDP3 corresponding to $i = 1, j = 2$. The constraints for the pairs $i = 1, j = 3$ and $i = 2, j = 3$ can be checked similarly.

Remark 2.5.6 below presents an example showing that this equivalence fails for $n \geq 4$.

2.5.1 Definition and Some Properties of the Set F_n

Let us define the projection of the feasible set of SDP3 onto $\mathcal{S}^n$ as

$$F_n := \{X \in \mathcal{S}^n : X = \text{sMat}\,(Y_{1:t(n),0}), Y \text{ feasible for SDP3}\}.$$

Since the feasible set of SDP3 is convex and compact, and since F_n is its image under a linear transformation, it follows that F_n is also convex and compact. Also, it is straightforward to verify that F_n contains the cut polytope.

Lemma 2.5.2 $C_n \subseteq F_n$.

Proof: Consider $X = vv^T, v \in \{\pm 1\}^n$. Let

$$x = \operatorname{svec}(X) \quad \text{and} \quad Y = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T.$$

We show that Y is feasible for SDP3. Indeed, $Y \succeq 0$ and $Y_{0,0} = 1$. Since $x_{T(i,j)} = v_i v_j$, for $1 \leq i \leq j \leq n$,

$$Y_{T(i,j),T(i,j)} = (x_{T(i,j)})^2 = {v_i}^2 {v_j}^2 = 1.$$

Therefore $\operatorname{diag}(Y) = e$. Also, $Y_{0,t(i)} = Y_{0,T(i,i)} = x_{T(i,i)} = {v_i}^2 = 1$.
Finally, for $1 \leq i < j \leq n$,

$$\begin{aligned} Y_{T(i,k),T(k,j)} &= x_{T(i,k)} x_{T(k,j)} \\ &= v_i v_k v_k v_j \\ &= v_i v_j \\ &= x_{T(i,j)} \\ &= Y_{0,T(i,j)}. \end{aligned}$$

Hence, each $X = vv^T, v \in \{\pm 1\}^n$ has a corresponding Y feasible for SDP3 and so $X \in F_n$.

The result now follows from the convexity of C_n and F_n. ■

Since every Y feasible for SDP3 is feasible for SDP2, by Lemma 2.4.3 we have:

Corollary 2.5.3 $F_n \subseteq \mathcal{E}_n$. ■

By Lemma 2.5.2, we observe that $\mu^* \leq \nu_3^* \leq \nu_2^* \leq \nu_1^*$. We now prove an additional property of SDP3 that is not inherited from SDP2, namely that the matrices in F_n also satisfy all the triangle inequalities.

Theorem 2.5.4 $F_n \subseteq M_n$.

Proof: Suppose $X \in F_n$, then $X = \mathrm{sMat}\,(Y_{1:t(n),0})$ for some Y feasible for SDP3. Since $Y_{0,t(i)} = 1 \ \forall\, i$, it follows that $\mathrm{diag}\,(X) = e$ holds.

Given i, j, k such that $1 \leq i < j < k \leq n$, let $Y_{i,j,k}$ denote the 4×4 principal submatrix of Y corresponding to the indices $\{0, T(i,j), T(i,k), T(j,k)\}$. Let $a = X_{ij} = Y_{0,T(i,j)}, b = X_{ik} = Y_{0,T(i,k)}, c = X_{jk} = Y_{0,T(j,k)}$. Then

$$Y_{i,j,k} = \begin{pmatrix} 1 & a & b & c \\ a & 1 & c & b \\ b & c & 1 & a \\ c & b & a & 1 \end{pmatrix},$$

since $\mathrm{diag}\,(Y) = e$ and

$$Y_{0,T(i,j)} = Y_{T(i,k),T(k,j)}, \quad Y_{0,T(i,k)} = Y_{T(i,j),T(j,k)}, \quad Y_{0,T(j,k)} = Y_{T(j,i),T(i,k)}.$$

Applying the Schur complement,

$$\begin{aligned} Y_{i,j,k} \succeq 0 \quad &\Leftrightarrow \quad \begin{pmatrix} 1 & c & b \\ c & 1 & a \\ b & a & 1 \end{pmatrix} - \begin{pmatrix} a \\ b \\ c \end{pmatrix} \begin{pmatrix} a & b & c \end{pmatrix} \succeq 0 \\ &\Leftrightarrow \quad \begin{pmatrix} 1-a^2 & c-ab & b-ac \\ c-ab & 1-b^2 & a-bc \\ b-ac & a-bc & 1-c^2 \end{pmatrix} \succeq 0 \end{aligned}$$

which implies

$$e^T \begin{pmatrix} 1-a^2 & c-ab & b-ac \\ c-ab & 1-b^2 & a-bc \\ b-ac & a-bc & 1-c^2 \end{pmatrix} e \geq 0.$$

Hence,

$$\begin{aligned} Y_{i,j,k} \succeq 0 \quad &\Rightarrow \quad 3 - (a+b+c)^2 + 2(a+b+c) \geq 0 \\ &\Leftrightarrow \quad \gamma^2 - 2\gamma - 3 \leq 0, \text{where}\, \gamma := a+b+c \\ &\Leftrightarrow \quad (\gamma - 3)(\gamma + 1) \leq 0 \\ &\Leftrightarrow \quad -1 \leq \gamma \leq 3 \\ &\Rightarrow \quad a+b+c \geq -1. \end{aligned}$$

Therefore, $X_{ij} + X_{ik} + X_{jk} \geq -1$ holds for X.

Because multiplication of row and column i of $Y_{i,j,k}$ by -1 will not affect the positive semidefiniteness of $Y_{i,j,k}$, multiplying the two rows and two columns of $Y_{i,j,k}$ with indices $T(i,k)$ and $T(j,k)$ and applying the same argument to the resulting matrix, we obtain $X_{ij} - X_{ik} - X_{jk} \geq -1$. Similarly, the inequalities $-X_{ij} + X_{ik} - X_{jk} \geq -1$ and $-X_{ij} - X_{ik} + X_{jk} \geq -1$ also hold. ■

We have thus proved:

Corollary 2.5.5 $C_n \subseteq F_n \subseteq \mathcal{E}_n \cap M_n$. ■

Remark 2.5.6 *Remark 2.5.1 pointed out that for $n = 3$, the relaxations SDP2 and SDP3 are equivalent. This remark shows explicitly that this equivalence fails for $n \geq 4$, and furthermore that the triangle inequalities are not enforced by SDP2.*

Consider the weighted adjacency matrix

$$A = \begin{pmatrix} 0 & 4 & 1 & 1 \\ 4 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

for a graph with $n = 4$ nodes. Then the maximum cut has a weight of 6 and is attained by the cut with node sets $\{1, 3\}, \{2, 4\}$ and by the cut with node sets $\{1\}, \{2, 3, 4\}$.

The computed SDP1 bound is 6.0625, that for SDP2 is 6.0112, and that for SDP3 is 6.0000.

An optimal matrix X for SDP2, obtained by taking the sMat *of the first column, is*

$$\begin{pmatrix} 1.0000e+00 & -9.9485e-01 & -2.1943e-02 & -9.9747e-01 \\ -9.9485e-01 & 1.0000e+00 & -2.3659e-02 & 9.9715e-01 \\ -2.1943e-02 & -2.3659e-02 & 1.0000e+00 & -1.2873e-03 \\ -9.9747e-01 & 9.9715e-01 & -1.2873e-03 & 1.0000e+00 \end{pmatrix}$$

which violates a triangle inequality, since

$$X_{12} + X_{13} + X_{23} \leq -1.0404 < -1,$$

and the triangle inequality requires that $X_{12} + X_{13} + X_{23} \geq -1$.

Hence, we conclude that the relaxation SDP2 does not automatically enforce the triangle inequalities when $n \geq 4$.

In Section 2.5.3, we will prove that the inclusions in Corollary 2.5.5 are in fact strict for $n \geq 5$. However, because we do not have an explicit description of F_n, first we need to address the issue of testing for membership in F_n. This is the focus of the next section.

2.5.2 Testing for Membership in F_n

The set F_n is defined as the image of the feasible set of SDP3 under the linear mapping sMat applied to the first row of every feasible matrix in SDP3. It is not clear how to give an explicit description of F_n, but given $X \in \mathcal{S}^n$, the question of determining whether $X \in F_n$ can be expressed as:

Given $X \in \mathcal{S}^n$ satisfying $\operatorname{diag}(X) = e$, does there exist a matrix Y feasible for SDP3 such that $\operatorname{sMat}(Y_{1:t(n),0}) = X$?

In this question, only some of the elements of Y are specified, namely the diagonal, the first row and column, and the elements fixed by the rank-two constraints. The remaining elements are considered "free" and we ask whether it is possible to choose them in such a way that the resulting matrix Y is positive semidefinite.

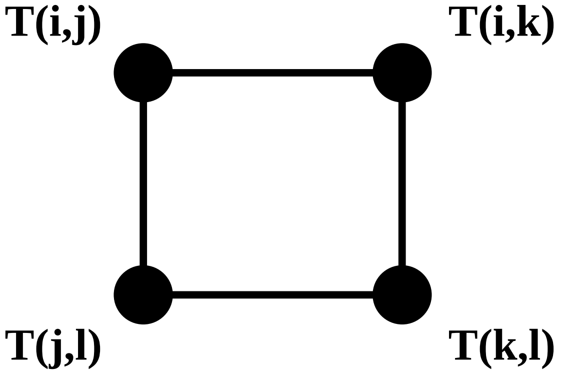


Figure 2.1: A chordless cycle of length 4 in the graph G_Y of Y

This problem is an instance of the positive semidefinite matrix completion problem, which has been extensively studied (see for example [42, 63, 54]).

We can associate with the partial matrix Y a finite undirected graph $G_Y = (V_Y, E_Y)$ as follows: let the node set be $V_Y := \{0, 1, \ldots, t(n)\}$ and let the edge set E_Y contain the edge (i, j) if and only if the entry $Y_{i,j}$ is fixed. Then G_Y is said to be *chordal* if every cycle of length ≥ 4 has a chord, i.e. an edge between two non-consecutive nodes. Grone et al. [42] showed that if the diagonal entries of Y are specified and the principal minors composed of fixed entries are all non-negative, then, if the graph G_Y is chordal, a positive semidefinite completion necessarily exists. In our case, however, it is easy to see that the graph G_Y is not chordal for $n \geq 4$. It suffices to consider the cycle of length 4 depicted in Figure 2.1; since $(T(i,j), T(k,l)) \notin E_Y$ and $(T(i,k), T(j,l)) \notin E_Y$, we see that the cycle has no chords. Moreover, chordless cycles of lengths $4, 5, \ldots, n$ can be found in the graph G_Y. Indeed, for every subset of c indices $\{i_1, i_2, \ldots, i_c\} \subseteq \{1, \ldots, n\}$, $c = 4, 5, \ldots, n$, the graph contains the chordless cycle depicted in Figure 2.2. So we must use a different approach.

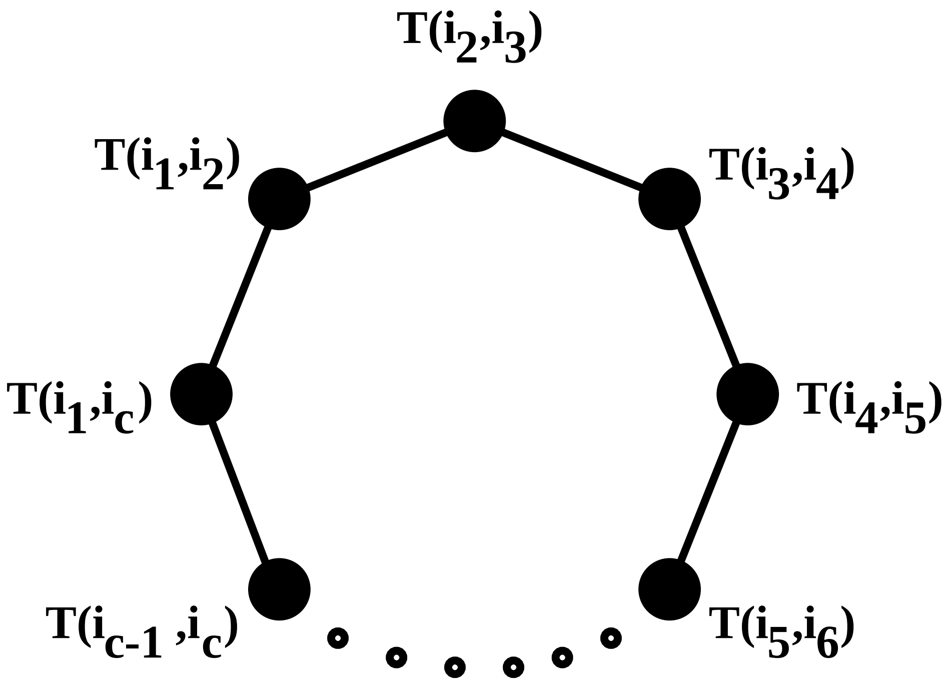


Figure 2.2: A chordless cycle of length c in the graph G_Y of Y

Johnson et al. [55] present an interior-point method for finding an approximate completion, if a completion exists. We use this approach to test membership in F_n. Specifically, we proceed as follows: Given $X \in \mathcal{S}^n$ with $\operatorname{diag}(X) = e$, let $x = \operatorname{svec}(X)$ and let $A \in \mathcal{S}^{t(n)+1}$ be some matrix which satisfies $\operatorname{sMat}(A_{1:t(n),0}) = X$ and furthermore satisfies all the constraints of SDP3 except (possibly) for the positive semidefiniteness constraint. Define $H \in \mathcal{S}^{t(n)+1}$ to be the $\{0,1\}$-matrix satisfying $H_{ij} = 0$ if A_{ij} is "free", and $H_{ij} = 1$ otherwise.

For example, if $X = (X_{ij})$ is 3×3, one possible choice of A is:

$$A = \begin{pmatrix} 1 & 1 & X_{12} & 1 & X_{13} & X_{23} & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ X_{12} & 0 & 1 & 0 & X_{23} & X_{13} & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ X_{13} & 0 & X_{23} & 0 & 1 & X_{12} & 0 \\ X_{23} & 0 & X_{13} & 0 & X_{12} & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

where the "free" entries are filled with zeros. The corresponding matrix H is:

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

To check whether A has a positive semidefinite completion, we consider the problem:

$$\begin{array}{rcl} c^* & = & \min \quad \|H \circ (A - B)\|_F^2 \\ & & \text{s.t.} \qquad B \succeq 0 \end{array}$$

and its dual:

$$\begin{array}{rcl} d^* & = & \max \quad \|H \circ (A-B)\|_F^2 - \operatorname{trace} \Lambda B \\ & & \text{s.t.} \qquad 2\ H \circ H \circ (B-A) = \Lambda \\ & & \qquad\qquad\qquad \Lambda \succeq 0, \end{array}$$

where $\|\cdot\|_F$ denotes the Frobenius matrix norm (see Johnson et al. [55] for more details). Clearly if $c^* = 0$, then the corresponding primal optimal solution B^* is an exact positive semidefinite completion of A. On the other hand, if we find a pair $(\bar{B}, \bar{\Lambda})$ such that $\|H \circ (A-\bar{B})\|_F^2 - \operatorname{trace} \bar{\Lambda}\bar{B} > 0$, then because $c^* \geq d^*$ (by weak duality), it follows that $c^* > 0$ and hence A has no positive semidefinite completion.

Using this approach, we can find examples which prove that the inclusions in Corollary 2.5.5 are in fact strict for $n = 5$, and hence for all $n \geq 5$.

2.5.3 Examples Proving Strict Inclusions

In this section we prove that the inclusions in Corollary 2.5.5 are strict.

Example 2.5.7 *Consider the matrix*

$$X = \begin{pmatrix} 1 & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & 1 & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{4} & 1 & -\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & 1 & -\frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & 1 \end{pmatrix}.$$

It is known that $X \notin C_5$ (see Laurent et al. [66]). Applying the algorithm described in the previous section, we found a 16×16 matrix B^* which is feasible for SDP3 *and such that* $\operatorname{sMat}(B^*_{0,1:15}) = X$*. The matrix* B^* *is defined as*

$$B^*_{T(i,j),0} = \begin{cases} 1, \text{ if } i = j \\ -\frac{1}{4}, \text{ otherwise} \end{cases}$$

$$B^*_{T(i,j),T(k,l)} = \begin{cases} 1, \text{ if } (i,j) = (k,l) \\ \frac{3}{8}, \text{ if } (i,j) \text{ and } (k,l) \text{ are disjoint} \\ -\frac{1}{4}, \text{ otherwise} \end{cases}$$

Hence $X \in F_5$*.*

Example 2.5.8 *Consider the matrix*

$$X = \begin{pmatrix} 1 & -0.65 & -0.65 & -0.65 & 0.93 \\ -0.65 & 1 & 0.3 & 0.3 & -0.65 \\ -0.65 & 0.3 & 1 & 0.3 & -0.65 \\ -0.65 & 0.3 & 0.3 & 1 & -0.65 \\ 0.93 & -0.65 & -0.65 & -0.65 & 1 \end{pmatrix}$$

It is easy to check that $X \in \mathcal{E}_5 \cap M_5$*. Applying the algorithm described in the previous section, we found feasible matrices* $\bar{B}$ *and* $\bar{\Lambda}$ *for which the dual objective value is equal to* $2.81 \times 10^{-4} > 0$*. Hence* $c^* > 0$ *and there is no matrix* B *feasible for SDP3 such that* $\operatorname{sMat}(B_{0,1:15}) = X$*. Thus* $X \notin F_5$*. The matrices* $\bar{B}$ *and* $\bar{\Lambda}$ *are given in Appendix A.*

Hence we have proved that

Theorem 2.5.9 $C_n \subsetneq F_n \subsetneq \mathcal{E}_n \cap M_n$ *for* $n \geq 5$. ■

Theorem 2.5.9 shows that SDP3 is a strict improvement over the addition of all the triangle inequalities to SDP1. In Chapter 3, we use the fact that every feasible matrix for SDP3 is singular to project the feasible set of SDP3 onto a lower dimensional face of the positive semidefinite cone where Slater's constraint qualification holds. In doing so we also reduce the dimension of the matrix variable, and hence the number of variables in SDP3. We then present computational results that demonstrate the strength of SDP3, illustrate the strict inequalities in Theorem 2.5.9, and motivate our further study of the geometry of F_n in Chapters 4 and 5.

Chapter 3

Projection onto the Minimal Face and Computational Results

In this chapter we continue the study of the geometrical structure of the feasible sets of our relaxations, and also present computational results that demonstrate their strength. We will focus our attention on the relaxation SDP3 but we observe at the end of Section 3.1 that the theoretical results in this chapter easily extend to SDP2.

Let $\mathcal{F}_n$ denote the set of $Y \in \mathcal{S}^{t(n)+1}$ that are feasible for SDP3. Since $\mathcal{F}_n$ has no strictly feasible points, we seek to express $\mathcal{F}_n$ in a lower dimensional space. We now show that this can be done without losing the sparsity of the constraints. The approach we follow is along the lines of the work in Zhao et al. [99, 98] and the general framework presented in Tunçel [93]. Our treatment here is specific to the strengthened relaxations SDP2 and SDP3 and provides further insight into the structure of these relaxations.

3.1 The Barycenter and a Projection onto the Minimal Face

In this section we study the structure of the barycenter of the feasible sets of SDP2 and SDP3. From the direct second lifting in Section 2.4.2, it is clear that the 2^{n-1} matrices

$$Y_v := \begin{pmatrix} 1 \\ x_v \end{pmatrix} \begin{pmatrix} 1 \\ x_v \end{pmatrix}^T, \quad x_v := \text{svec}\,(vv^T), v \in \mathcal{V} := \{\pm 1\}^n$$

all belong to $\mathcal{F}_n$. Furthermore, since these are the points we are interested in, we want to project the feasible set onto Φ, the minimal face of the positive semidefinite cone (in $\mathcal{S}^{t(n)+1}$) such that $Y_v \in \Phi \;\forall\, v \in \mathcal{V}$.

Consider the barycenter of the set of points Y_v:

$$\hat{Y} := 2^{-n} \sum_{v \in \mathcal{V}} Y_v.$$

By definition of Φ, $\hat{Y} \in \operatorname{relint} \Phi$. Since Φ is a proper face, we can find a mapping from a lower dimensional positive semidefinite cone to Φ. We construct this mapping using the results of the next theorem, which describes the structure of the barycenter $\hat{Y}$.

Let $P_{i,j}$ denote the $(t(n)+1) \times (t(n)+1)$ permutation matrix equal to the identity matrix with the i^{th} and j^{th} columns permuted. We define the (permutation) matrix

P as the following product of $(n-1)$ permutation matrices:

$$P := P_{2,t(2)}\, P_{3,t(3)}\, \dots\, P_{n,t(n)}.$$

Theorem 3.1.1 *The following statements hold for the barycenter $\hat{Y}$:*

1. *$\hat{Y}$ is a $\{0,1\}$-matrix and*

$$\hat{Y}_{ij} = \begin{cases} 1, & \text{if} \quad i = t(k), j = t(l), k, l \in \{1, \dots, n\}, k \neq l \\ 1, & \text{if} \quad i = j \in \{0, 1, \dots, t(n)\} \\ 0, & \text{otherwise.} \end{cases}$$

2. *The rank of $\hat{Y}$ is $t(n-1)+1$ and the eigenvalues are $(n+1, 1, 0)$ with multiplicities $(1, t(n-1), n)$ respectively.*

3. *The null space and range space of $\hat{Y}$ are*

$$\mathcal{N}(\hat{Y}) = \mathcal{R}\left(P \begin{bmatrix} V \\ 0 \end{bmatrix} \right)$$

and

$$\mathcal{R}(\hat{Y}) = \mathcal{R}\left(P \begin{bmatrix} e & 0 \\ 0 & I_{t(n-1)} \end{bmatrix} \right)$$

respectively, where $V \in \Re^{(n+1)\times n}$ is any matrix such that $\begin{bmatrix} e & V \end{bmatrix}$ is an orthogonal matrix.

Proof:

1. Let $v \in \mathcal{V}$ and consider Y_v. The elements of x_v have the form $(x_v)_j = v_\alpha v_\beta$, where $j = t(\beta - 1) + \alpha$ for $\alpha, \beta \in \{1, \dots, n\}, \alpha \neq \beta$, and furthermore

$$(x_v)_j = \begin{cases} 1, & \text{if} \quad v_\alpha = v_\beta \\ -1, & \text{otherwise.} \end{cases}$$

First consider the case where $\alpha = \beta = k$; here $j = t(k)$ and it is clear that $(x_v)_{t(k)} = 1, k = 1, \dots, n$. This holds independently of the choice of v so we may conclude that

$$\begin{pmatrix} 1 \\ x_v \end{pmatrix}_{t(k)} = 1,\, k = 0, \dots, n, \quad \forall\, v \in \mathcal{V}. \tag{3.1}$$

Now suppose $\alpha \neq \beta$; then $v_\alpha = v_\beta$ for exactly 2^{n-1} elements of $\mathcal{V}$ and $v_\alpha \neq v_\beta$ for the other 2^{n-1} choices of v. Hence,

$$\sum_{v \in \mathcal{V}} (x_v)_j = 0, \forall\, j \notin \{t(0), \dots, t(n)\}. \tag{3.2}$$

Equations (3.1) and (3.2) together imply that the 0^{th} column of $\hat{Y}$ equals $\sum_{k=0}^n e_{t(k)}$, i.e.

$$\hat{Y}_{i,0} = \begin{cases} 1, & \text{if} \quad i \in \{t(0), \dots, t(n)\} \\ 0, & \text{otherwise.} \end{cases}$$

By symmetry of $\hat{Y}$, $\hat{Y}_{0,j} = \hat{Y}_{j,0}$, so it remains to examine $\hat{Y}_{i,j}$ for $i, j = 1, \dots, t(n)$.

The remaining $t(n)$ columns of $\hat{Y}$ are:

$$\hat{Y}_{:,j} = 2^{-n} \sum_{v \in \mathcal{V}} (x_v)_j \begin{pmatrix} 1 \\ x_v \end{pmatrix},$$

for $j = 1, \ldots, t(n)$. If $i = j$ then $\hat{Y}_{i,i} = 2^{-n} \sum_{v \in \mathcal{V}} (x_v)_i^2 = 1$, so we now suppose $i \neq j$.

If $i = t(k)$ and $j = t(l)$ for some $k, l \in \{1, \ldots, n\}, k \neq l$, then

$$\hat{Y}_{i,j} = 2^{-n} \sum_{v \in \mathcal{V}} (x_v)_{t(k)} (x_v)_{t(l)} = 1,$$

using (3.1).

If $i \neq t(k)$, $\forall k$ but $j = t(l)$, then

$$\hat{Y}_{i,j} = 2^{-n} \sum_{v \in \mathcal{V}} (x_v)_i = 1,$$

using (3.1) and (3.2). The case $i = t(k)$ but $j \neq t(l)$, $\forall l$ is handled similarly.

Finally, if $i \neq t(k)$, $\forall k$, and $j \neq t(l)$, $\forall l$, then we need only observe that $((x_v)_i, (x_v)_j) = (1, 1)$ in exactly 2^{n-2} elements of $\mathcal{V}$, and the same count also holds for each of the combinations $(1, -1)$, $(-1, 1)$, and $(-1, -1)$. Thus

$$\sum_{v \in \mathcal{V}} (x_v)_i (x_v)_j = 0$$

and hence $\hat{Y}_{i,j} = 0$.

2. Define

$$\hat{Y}_P := P^T \hat{Y} P = \begin{bmatrix} E & 0 \\ 0 & I_{t(n)-n} \end{bmatrix} \in \mathcal{S}^{(t(n)+1)\times(t(n)+1)}.$$

   Since this is a similarity transformation, $\hat{Y}$ and $\hat{Y}_P$ have exactly the same eigenvalues and it suffices to prove the result for $\hat{Y}_P$. Also, $\hat{Y}_P$ is block diagonal, so its eigenvalues are those of the blocks. The lower block has the eigenvalue 1 with multiplicity $t(n)-n = t(n-1)$ (we have the set of standard eigenvectors $e_{n+1}, \ldots, e_{t(n)}$). The upper block is clearly rank-one; since

$$\hat{Y}_P \begin{pmatrix} e \\ 0 \end{pmatrix} = (n+1) \begin{pmatrix} e \\ 0 \end{pmatrix},$$

   $n+1$ is its only non-zero eigenvalue. For V as in the statement of the theorem, $\hat{Y}_P \begin{pmatrix} V \\ 0 \end{pmatrix} = 0$. So the columns of V (extended with zeros) give a set of eigenvectors for the zero eigenvalue, which has multiplicity n.

3. The result follows by the similarity of $\hat{Y}$ and $\hat{Y}_P$ and the proof of the previous part of the theorem.

■

Now define the matrix

$$W := \ P \ \begin{bmatrix} e & 0 \\ 0 & I_{t(n-1)} \end{bmatrix} \in \Re^{(t(n)+1)\times(t(n-1)+1)},$$

with $e \in \Re^{n+1}$. Then $\mathcal{R}(\hat{Y}) = \mathcal{R}(W)$ and W provides a mapping from $\mathcal{S}^{t(n)+1}$ to the minimal face Φ: if $Y \in \Phi$ then $Y = W\,Y_P\,W^T$ for $Y_P \in \mathcal{S}^{t(n-1)+1}$, and we require $Y_P \succeq 0$ to stay in the positive semidefinite cone of the lower dimensional space.

The projected version of SDP3 is thus:

$$\begin{array}{rll} \nu_3^* = & \max & \text{trace}\,(W^T H_Q W) Y_P \\ & \text{s.t.} & \text{trace}\,(W^T E_{ii} W) Y_P = 1, i = 0, \ldots, t(n) \\ & & \text{trace}\,(W^T E_{0,t(i)} W) Y_P = 1, i = 1, \ldots, n \\ & & \text{trace}\,(W^T (E_{0,T(i,j)} - E_{T(i,k),T(k,j)}) W) Y_P = 0, \\ & & \qquad\qquad \forall\, k = 1, \ldots, n, \forall\, 1 \le i < j \le n \\ & & Y_P \succeq 0, Y_P \in \mathcal{S}^{t(n-1)+1}, \end{array}$$

where $E_{ij} := \frac{1}{2}(e_i e_j^T + e_j e_i^T)$. It remains to remove all the redundant constraints in this projected problem.

Let w_i denote the i^{th} column of W^T. The construction of W implies $w_0^T = w_{t(i)}^T = e_0^T$, $\forall i \in \{1, \ldots, n\}$, and the remaining columns of W, i.e.

$$\{w_{T(i,j)}^T : i, j \in \{1, \ldots, n\}, i < j\} = \{e_1^T, e_2^T, \ldots, e_{t(n-1)}^T\}$$

form a linearly independent set. (Together with w_0^T, they form a basis for $\Re^{t(n-1)+1}$.) Now, since $W^T E_{ii} W = w_i w_i^T$ and $W^T E_{0,t(i)} W = \frac{1}{2}(w_0 w_{t(i)}^T + w_{t(i)}^T w_0^T)$, we have

$$W^T E_{t(i),t(i)} W = w_0 w_0^T = W^T E_{00} W, \forall\, i \in \{1, \ldots, n\}$$

and

$$W^T E_{0,t(i)} W = w_0 w_0^T = W^T E_{00} W, \forall\, i \in \{1, \ldots, n\}.$$

Furthermore,

$$W^T(E_{0,T(i,j)} - E_{T(i,k),T(k,j)})W = \\ \tfrac{1}{2}\{w_0 w_{T(i,j)}^T + w_{T(i,j)} w_0^T - w_{T(i,k)} w_{T(k,j)}^T - w_{T(k,j)} w_{T(i,k)}^T\},$$

therefore if $k = i$ or $k = j$ then $w_{T(i,k)} = w_0$ or $w_{T(k,j)} = w_0$ respectively. Hence,

$$W^T(E_{0,T(i,j)} - E_{T(i,k),T(k,j)})W = 0 \quad \text{if } \ k = i \ \text{ or } \ k = j, \tag{3.3}$$

and the corresponding constraint is redundant. Removing all these redundant constraints, we obtain SDP3$_{\mathrm{P}}$:

$$(\mathrm{SDP3_P}) \qquad \begin{array}{rll} \nu_3^* = & \max & \operatorname{trace}(W^T H_Q W) Y_P \\ & \text{s.t.} & \operatorname{trace}(W^T E_{ii} W) Y_P = 1, \\ & & \qquad\qquad i \in \{0, 1, \ldots, t(n)\} \backslash \{t(1), \ldots, t(n)\} \\ & & \operatorname{trace}(W^T(E_{0,T(i,j)} - E_{T(i,k),T(k,j)})W) Y_P = 0, \\ & & \qquad\qquad \forall\, k \notin \{i,j\}, \forall\, 1 \le i < j \le n \\ & & Y_P \succeq 0, Y_P \in \mathcal{S}^{t(n-1)+1}. \end{array}$$

It is straightforward to check that all the remaining constraints are linearly independent. Moreover, we prove that Slater's constraint qualification holds for SDP3$_{\mathrm{P}}$. This implies that the optimal values of SDP3$_{\mathrm{P}}$ and its dual are equal and we can use a primal-dual interior-point algorithm.

First we simplify our notation. We have the following primal-dual pair:

$$
(\mathrm{SDP3_P}) \quad \begin{array}{ll}
\max & \operatorname{trace} C\,Y \\
\text{s.t.} & \operatorname{diag} Y = e \\
 & \operatorname{trace} A_{ijk} Y = 0, \quad \forall\,(i,j,k) \in \mathcal{J} \\
 & Y \succeq 0,\ \ Y \in \mathcal{S}^{t(n-1)+1}
\end{array}
$$

$$
(\mathrm{DSDP3_P}) \quad \begin{array}{ll}
\min & \sum\limits_{i=1}^{t(n-1)+1} x_i \\
\text{s.t.} & S = \operatorname{Diag}(x) + \sum\limits_{(i,j,k)\in\mathcal{J}} y_{ijk} A_{ijk} - C \\
 & S \succeq 0, \\
 & x \in \Re^{t(n-1)+1},\ \ y \in \Re^{(n-2)\cdot t(n-1)},
\end{array}
$$

where

$$\mathcal{J} := \{(i,j,k) : i,j \in \{1,\ldots,n\}, i<j, k \notin \{i,j\}\},$$

$$A_{ijk} := W^T(E_{0,T(i,j)} - E_{T(i,k),T(k,j)})W \quad \forall\,(i,j,k) \in \mathcal{J}$$

and

$$C := W^T H_Q W.$$

Lemma 3.1.2 *Slater's constraint qualification holds for* SDP3$_\mathrm{P}$.

Proof: We consider the matrix $\tilde{Y} := I_{t(n-1)+1}$. Since $\tilde{Y} \succ 0$, we simply need to verify that it satisfies the equality constraints.

Clearly, $\operatorname{diag} \tilde{Y} = e$. Now observe that

$$\begin{array}{rcl} WW^T & = & P \begin{bmatrix} e & 0 \\ 0 & I_{t(n-1)} \end{bmatrix} \begin{bmatrix} e^T & 0 \\ 0 & I_{t(n-1)} \end{bmatrix} P^T \\ & = & P \begin{bmatrix} E & 0 \\ 0 & I_{t(n-1)} \end{bmatrix} P^T \\ & = & P \, \hat{Y}_P \, P^T \\ & = & \hat{Y}, \end{array}$$

where $\hat{Y}_P$ is the matrix defined in the proof of Theorem 3.1.1.

Using this observation, the second set of equality constraints for $\tilde{Y}$ may be written as

$$\operatorname{trace}\,(E_{0,T(i,j)} - E_{T(i,k),T(k,j)})\hat{Y} = 0, \ \forall\,(i,j,k) \in \mathcal{J},$$

and these equalities hold because

$$\operatorname{trace}\,(E_{0,T(i,j)} - E_{T(i,k),T(k,j)})\hat{Y} = 0 \Leftrightarrow \hat{Y}_{0,T(i,j)} = \hat{Y}_{T(i,k),T(k,j)},$$

and by Theorem 3.1.1(1) both these entries of $\hat{Y}$ are zero. ■

It is straightforward to prove that the same is true for the dual problem.

Lemma 3.1.3 *Slater's constraint qualification holds for* DSDP3$_{\text{P}}$.

Proof: Choosing $\tilde{y}_{ijk} := 0 \ \forall\,(i,j,k) \in \mathcal{J}$ and $\tilde{x}_i := \|\operatorname{dsvec}(Q)\|_1 + 1 \ \forall\, i =$

$1, \ldots, t(n-1)+1$, the corresponding dual (slack) variable is

$$\tilde{S} = (\|\mathrm{dsvec}\,(Q)\|_1 + 1) I_{t(n-1)+1} - C$$

which is strictly diagonally dominant and has all its diagonal entries positive. Hence $\tilde{S}$ is positive definite. ■

All the results in this section extend to SDP2. The corresponding projected problem is

$$(\mathrm{SDP2_P}) \qquad \begin{array}{rll} \nu_2^* = & \max & \mathrm{trace}\, C\, Y_P \\ & \text{s.t.} & \mathrm{diag}\, Y_P = e, \\ & & \mathrm{trace}\,(W^T R_{ij}\, W) Y_P = 0, \;\; \forall\, 1 \le i < j \le n \\ & & Y_P \succeq 0, Y_P \in \mathcal{S}^{t(n-1)+1}, \end{array}$$

where

$$R_{ij} := n\, E_{0,T(i,j)} \quad - \quad \sum_{k=1}^{n} E_{T(i,k),T(k,j)}.$$

Using equation (3.3), it is straightforward to check that

$$W^T R_{ij}\, W = (n-2)\, W^T E_{0,T(i,j)}\, W \quad - \quad \sum_{k=1,\; k \notin \{i,j\}}^{n} W^T E_{T(i,k),T(k,j)\, W}.$$

3.2 Comparison of the Relaxations for Selected Graphs

The relaxations SDP1, SDP2$_\mathrm{P}$, and SDP3$_\mathrm{P}$ were compared for several interesting problems using the software package SDPpack (version 0.9 Beta) [4]. For completeness we also solved the metric polytope relaxation:

$$\begin{array}{ll} \max & \operatorname{trace} QX \\ \text{s.t.} & X \in M_n, \end{array}$$

and the relaxation obtained by adding all the triangle inequalities to SDP1:

$$\begin{array}{ll} \max & \operatorname{trace} QX \\ \text{s.t.} & X \in \mathcal{E}_n \cap M_n. \end{array}$$

The metric polytope relaxation is easily formulated as an LP and we solved it using the Matlab solver LINPROG. The results are summarized in Table 3.1. The value ρ equals the value of the optimal cut divided by the bound, and R.E. denotes the relative error with respect to the optimal cut. A relative error equal to zero means that the relative error was below 10^{-6}.

The test problems in Table 3.1 are as follows:

- The first line of results corresponds to solving the relaxations for a 5-cycle, denoted C_5, with unit edge weights. This example is well known for almost achieving the worst-case of the analysis of Goemans and Williamson [41]. Observe that SDP3$_\mathrm{P}$ solves it exactly.

Graph	μ^*	SDP1 bound	SDP2$_\mathrm{P}$ bound	M_n bound	$\mathcal{E}_n \cap M_n$ bound	SDP3$_\mathrm{P}$ bound
C_5	4	4.5225 $\rho = 0.8845$ R.E.: 13.06%	4.2889 $\rho = 0.9326$ R.E.: 7.22%	4.0000 $\rho = 1.0000$ R.E.: 0%	4.0000 $\rho = 1.0000$ R.E.: 0%	4.0000 $\rho = 1.0000$ R.E.: 0%
K_5	6	6.2500 $\rho = 0.9600$ R.E.: 4.17%	6.2500 $\rho = 0.9600$ R.E.: 4.17%	6.6667 $\rho = 0.9000$ R.E.: 11.11%	6.2500 $\rho = 0.9600$ R.E.: 4.17%	6.2500 $\rho = 0.9600$ R.E.: 4.17%
$K_5 \backslash e$	6	6.2500 $\rho = 0.9600$ R.E.: 4.17%	6.1160 $\rho = 0.9810$ R.E.: 1.93%	6.0000 $\rho = 1.0000$ R.E.: 0%	6.0000 $\rho = 1.0000$ R.E.: 0%	6.0000 $\rho = 1.0000$ R.E.: 0%
K_5 given by $A(G)$	9.28	9.6040 $\rho = 0.9663$ R.E.: 3.49%	9.4056 $\rho = 0.9866$ R.E.: 1.35%	9.3867 $\rho = 0.9886$ R.E.: 1.15%	9.2961 $\rho = 0.9983$ R.E.: 0.17%	9.2800 $\rho = 1.0000$ R.E.: 0%
AW_9^2	12	13.5 $\rho = 0.8889$ R.E.: 12.50%	12.9827 $\rho = 0.9243$ R.E.: 8.19%	12.8571 $\rho = 0.9333$ R.E.: 7.14%	12.6114 $\rho = 0.9515$ R.E.: 5.10%	12.4967 $\rho = 0.9603$ R.E.: 4.14%
Pet. $(n = 10)$	12	12.5 $\rho = 0.9600$ R.E.: 4.17%	12.3781 $\rho = 0.9695$ R.E.: 3.15%	12.0000 $\rho = 1.0000$ R.E.: 0%	12.0000 $\rho = 1.0000$ R.E.: 0%	12.0000 $\rho = 1.0000$ R.E.: 0%
Given in App.B $(n = 12)$	88	90.3919 $\rho = 0.9735$ R.E.: 2.72%	89.5733 $\rho = 0.9824$ R.E.: 1.79%	89.3333 $\rho = 0.9851$ R.E.: 1.52%	88.0029 $\rho = 1.0000$ R.E.: $3.3E-5$	88.0000 $\rho = 1.0000$ R.E.: 0%

Table 3.1: Computational comparison of all Max-Cut relaxations for selected test problems

- The next three examples are for the complete graph on 5 nodes, denoted K_5, with different choices of edge weights.

  - The second line in the table corresponds to K_5 with unit edge weights on all edges. In this example, none of the four SDP relaxations attains the Max-Cut optimal value, and in fact they are not distinguishable. Only the polyhedral relaxation M_5 gives a noticeably weaker bound.

  - The third line corresponds to K_5 with unit edge weights on all but one edge, which is assigned weight zero. For this example, the relaxation M_5 attains the exact Max-Cut optimal value (q.v. Barahona's result quoted in Section 2.2), hence SDP3_P is also exact, by Theorem 2.5.4.

  - The fourth line corresponds to the K_5 defined by the weighted adjacency matrix

$$A(G) = \begin{pmatrix} 0 & 1.52 & 1.52 & 1.52 & 0.16 \\ 1.52 & 0 & 1.60 & 1.60 & 1.52 \\ 1.52 & 1.60 & 0 & 1.60 & 1.52 \\ 1.52 & 1.60 & 1.60 & 0 & 1.52 \\ 0.16 & 1.52 & 1.52 & 1.52 & 0 \end{pmatrix}.$$

    This problem is interesting because the optimal value is different for each relaxation. Hence this example illustrates the second strict inclusion in Theorem 2.5.9. Furthermore, SDP3_P is the only relaxation that attains the Max-Cut optimal value.

- The fifth line corresponds to the antiweb AW_9^2 [30, Definition 29.1.1] with unit edge weights[1]. The graph is depicted in Figure 3.1. This example also gives a different optimal value for each relaxation; however, unlike the previous example, $\mathrm{SDP3_P}$ is *not* exact for this graph. Hence this example illustrates both strict inclusions in Theorem 2.5.9.

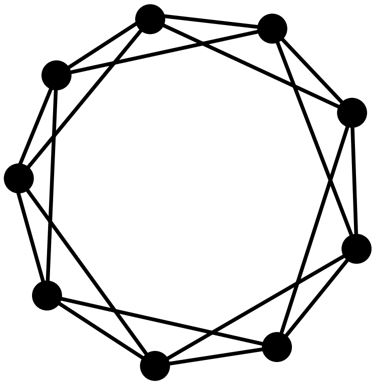


Figure 3.1: Antiweb AW_9^2

- The last two lines correspond to slightly larger graphs. The first one has 10 nodes; it is the well-known Petersen graph with unit edge weights. The second one is a graph with 12 nodes that also gives a different result for each relaxation; its weighted adjacency matrix is given in Appendix B.

The results of Table 3.1 cover only small problems. This is because solving the relaxations $\mathrm{SDP2_P}$ and $\mathrm{SDP3_P}$ using SDPpack becomes extremely time-consuming and requires large amounts of memory even for moderate values of n. To verify the behaviour of the relaxations on larger problems, we considered two other SDP packages. One was CSDP (version 2.3), a C implementation of an interior-point method developed by Brian Borchers [16, 17] and accessible via the NEOS Server for

[1] Thanks to Franz Rendl for suggesting this example.

Optimization [33] at `http://www-neos.mcs.anl.gov/neos/`. The second package was SBmethod, a C++ implementation of the spectral bundle method developed by Helmberg et al. [49, 46, 45]. Using these packages, we obtained the results presented in Table 3.2. Again, a relative error equal to zero means that the relative error was below 10^{-6}.

We point out that for SBmethod, the termination precision parameter was set (by default) to 10^{-5}. This is the value used to determine whether the maximal progress of the next step is small in comparison to the absolute value of the function (see Helmberg [45] for more details). This setting explains the relative errors greater than 10^{-6} in the SDP3$_{\mathrm{P}}$ bound for $n \geq 30$.

The bound SDP1 is known to be excellent both theoretically ($\rho \geq .878$ by Goemans and Williamson [41]) and empirically ($\rho \cong .97$, see for example Helmberg et al. [50]). Nonetheless, these computational results show that SDP2$_{\mathrm{P}}$ and SDP3$_{\mathrm{P}}$ frequently yield a *strict* improvement over SDP1, and that on randomly generated test problems (with non-negative integer weights), the SDP3$_{\mathrm{P}}$ relaxation often yields the *optimal* value of Max-Cut. This motivates our study (in the next section) of the application of SDP3$_{\mathrm{P}}$ to spin glass problems.

For simplicity of notation, in the remainder of this chapter we drop the subscript P but note that all computation is carried out using SDP3$_{\mathrm{P}}$.

Number of nodes	μ^*	SDP1 bound	SDP2$_\text{P}$ bound	SDP3$_\text{P}$ bound
10	648	666.428 $\rho = 0.9723$ R.E.: 2.84%	656.8020 $\rho = 0.9866$ R.E.: 1.36%	648.000 $\rho = 1.0000$ R.E.: 0%
11	1060	1084.345 $\rho = 0.9775$ R.E.: 2.30%	1072.352 $\rho = 0.9885$ R.E.: 1.17%	1060.000 $\rho = 1.0000$ R.E.: 0%
15	2290	2317.354 $\rho = 0.9882$ R.E.: 1.19%	2301.634 $\rho = 0.9949$ R.E.: 0.51%	2290.000 $\rho = 1.0000$ R.E.: 0%
16	2270	2318.867 $\rho = 0.9789$ R.E.: 2.15%	2300.354 $\rho = 0.9868$ R.E.: 1.34%	2270.000 $\rho = 1.0000$ R.E.: 0%
25	380	385.4737 $\rho = 0.9858$ R.E.: 1.44%	383.6503 $\rho = 0.9905$ R.E.: 0.96%	380.000 $\rho = 1.0000$ R.E.: 0%
30	1705.5	1751.600 $\rho = 0.9737$ R.E.: 2.70%	1743.205 $\rho = 0.9784$ R.E.: 2.21%	1705.578 $\rho = 1.0000$ R.E.: $4.6E - 5$
33	1888.5	1932.968 $\rho = 0.9770$ R.E.: 2.35%	1926.119 $\rho = 0.9805$ R.E.: 1.99%	1888.564 $\rho = 1.0000$ R.E.: $3.4E - 5$
36	27108.55	28305.28 $\rho = 0.9577$ R.E.: 4.41%	27944.30 $\rho = 0.9701$ R.E.: 3.08%	27108.81 $\rho = 1.0000$ R.E.: $9.8E - 6$

Table 3.2: Computational comparison of SDP1, SDP2$_\text{P}$, and SDP3$_\text{P}$ on randomly generated graphs with non-negative edge weights

3.3 Application of SDP3 to Spin Glass Problems

An interesting application of the Max-Cut problem arises in the area of statistical physics. We briefly sketch the context and relevance of Max-Cut to this field and then present computational results. The background material presented here is drawn from Fischer and Hertz [34], Jünger and Rinaldi [57], and Section 4.5 of Deza and Laurent [30].

One of the models developed by physicists in their quest to understand condensed states of matter is the so-called *spin glass* model. Spin glasses are one of the most complex kinds of condensed states studied in solid state physics and are not hard to find experimentally. A spin glass is any system consisting of a collection of magnetic moments (spins) whose low-temperature behaviour is disordered, in contrast to the uniform or periodic behaviour found in magnets, for example. To visualize one simple instance of such a system, consider a host non-magnetic metal (e.g. gold) and a multitude of small magnetic particles diluted in the host metal.

The magnetic particles have a magnetic orientation (spin) which can be described by a unit vector in $\Re^3$. Since each spin generates a magnetic field, the interactions between this field and every other spin will lead pairs of particles to interact with each other and orient themselves in identical or opposite directions. This interaction between pairs of spins is most evident at low temperatures and yields a certain ordered configuration for the entire system. As a result, most of the interesting behaviours of spin glasses correspond to states of minimal energy, commonly called *ground states*.

To observe ground states experimentally requires the system to be cooled towards 0°K very slowly and is thus quite time-consuming. Therefore mathematical models have been developed in order to conduct such experiments using computer simulations. Finding the ground state of a spin glass system is thus reduced to the optimization problem of *minimizing* the Hamiltonian representing the total energy of the system. We now show how, under certain simplifying assumptions, this optimization problem is in fact a Max-Cut problem.

3.3.1 The Ising Model

An important simplification of the spin glass model arises by assuming that each spin can take only one of two possible orientations ("north" and "south") and that all the spins are parallel to each other. The three-dimensional vector describing the spin of each particle can thus be reduced to a single scalar variable restricted to the values $+1$ or -1. This simplification, first proposed by in 1925 by Ernst Ising [53], leads to the well-known Ising model. This model also allows for the presence of a constant external magnetic field.

Let us suppose that the spin glass has n particles and let $v_i \in \{\pm 1\}$ denote the magnetic orientation of particle i and let v_0 denote the magnetic orientation of the exterior magnetic field which has strength h. For any pair i, j of particles, let J_{ij} denote the magnitude of the interaction between them. For example

$$J_{ij} = A \frac{\cos(D r_{ij})}{B^3 r_{ij}^3},$$

where r_{ij} is the distance between the particles, and A, B, and D are given constants which depend on the materials in the system. Then the energy of the system is modelled by the Hamiltonian

$$H := -\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} J_{ij}v_iv_j - h\sum_{j=1}^{n} v_0v_j.$$

To find the ground state, we want to minimize H over the 2^n possible configurations of the Ising spins (we may assume $v_0 = +1$ without loss of generality). Equivalently, we can maximize $-H$.

3.3.2 Max-Cut Formulation

Let G be a complete graph with node set $V = \{0, 1, \dots, n\}$ and with edge weights equal to

$$w_{ij} = -J_{ij}, \text{ for } i, j = 1, \dots, n,\ i < j,$$

$$w_{0j} = -h, \text{ for } j = 1, \dots, n.$$

Next rewrite the Hamiltonian H as

$$\begin{aligned} H &= \sum_{i=0}^{n-1}\sum_{j=i+1}^{n} w_{ij}v_iv_j \\ &= \sum_{i=0}^{n-1}\sum_{j=i+1:v_i=v_j}^{n} w_{ij}v_iv_j + \sum_{i=0}^{n-1}\sum_{j=i+1:v_i\neq v_j}^{n} w_{ij}v_iv_j \\ &= \sum_{ij\notin\delta(S)} w_{ij} \quad - \sum_{ij\in\delta(S)} w_{ij}, \end{aligned}$$

where $S = \{i \in V : v_i = +1\}$ defines a cut in the graph and

$$\delta(S) := \{ij : i \in S, j \in V \backslash S\}$$

equals the set of edges that are cut. Note that $v_i = v_j$ if and only if the edge $ij \notin \delta(S)$.

Now define the constant T to equal the sum of all the edge weights:

$$T := \sum_{i=0}^{n-1} \sum_{j=i+1}^{n} w_{ij}.$$

Then

$$H - T = -2 \sum_{ij \in \delta(S)} w_{ij}$$

and so $-H = 2 \sum_{ij \in \delta(S)} w_{ij} - T$. Therefore minimizing the Hamiltonian is equivalent to the optimization problem:

$$\begin{array}{ll} \max & 2 \sum_{ij \in \delta(S)} w_{ij} - T \\ \text{s.t.} & v \in \{\pm 1\}^n, \end{array}$$

which is a Max-Cut problem.

3.3.3 Further Modelling Issues

The pairwise interactions J_{ij} in a spin glass may be non-zero for every pair of particles. Because these interactions decrease very rapidly as the distance between the two particles increases, it is often assumed that $J_{ij} = 0$ for pairs of particles "sufficiently" far apart (the exception is the external field which interacts with every particle in the system). This idea leads to the so-called *short range interaction models*, where the particles are placed at regular positions in a two- or three-dimensional grid and the spin of each particle is assumed to interact only with its nearest neighbours on the grid. The resulting arrangement corresponds to a grid graph.

However, if the grid is not sufficiently large, the behaviour of spins at the boundary of the grid may lead to erroneous computational results. One way to alleviate this problem in the simulation is to allow wrap-around and let the boundary spins interact with their "neighbours" on the opposite grid boundary. The result is a system with periodic boundary conditions and a toroidal graph structure. Because of the limitations on the size of the SDP3 relaxations that we can solve, our computational study will be limited to such toroidal graphs.

There are several models in the literature for generating the magnitudes and signs of the pairwise interactions. We consider only two such models. The Gaussian model draws the interactions from a Gaussian distribution, whereas the $\pm J$ model draws the interactions from a binary distribution that takes on only ± 1 values. (The actual constant J can always be factored out in front of the Hamiltonian.)

All the Ising spin glass problems in our study were generated using the `torusgen` codes from Jünger and Liers [56]. The generator of Gaussian instances uses mean 0 and variance 1. The generator of ± 1 instances includes a parameter that specifies the percentage of interactions valued at -1. The case where this is equal to 50% is of particular interest in this context, therefore we experimented with values in the range 40% to 60%.

3.3.4 Computational Study of the Semidefinite Relaxations SDP1 and SDP3 for Ising Spin Glass Problems

Our first two tables of computational results illustrate the behaviour of the relaxations SDP1 and SDP3 on 20 Ising spin glass problems. All the problems have periodic boundary conditions. The numbering scheme for the problems is straightforward:

- G2-xx are two-dimensional Ising problems with pairwise interactions drawn from a Gaussian distribution with mean 0 and variance 1;
- G3-xx are three-dimensional Ising problems with pairwise interactions drawn from a Gaussian distribution with mean 0 and variance 1;
- PM2-xx and PM3-xx are (respectively) two- and three-dimensional Ising problems with ± 1 pairwise interactions. For each problem, the percentage of interactions valued at -1 is specified.

Table 3.3 gives the details for the problems we solved. For the case of problems with ± 1 interactions, we picked one configuration of the 2-D ± 1 problems, namely the 4×4, and one configuration of the 3-D ± 1 problems, namely the $2 \times 3 \times 3$, and varied the percentage of -1's away from 50%.

Both SDP relaxations were solved using the CSDP package. The times reported were calculated from the NEOS output and give an indication of the magnitude of computational effort required for each bound. In particular, we see that problems with 20 or more spins lead to an instance of SDP3 that typically takes over an hour to solve using CSDP.

Table 3.4 shows how the bounds SDP1 and SDP3 performed on each problem. For both bounds, the value of ρ equals the ratio of the optimal Max-Cut value to the SDP bound. (For all our test problems the dimensions are small so the optimal Max-Cut value is easy to calculate). We also checked whether the optimal matrices (in the space $\mathcal{S}^n$) computed by CSDP for each problem were rank-one in the following sense: a matrix was considered to be rank-one if the second largest eigenvalue λ_2 satisfied $\lambda_2 \leq 10^{-6}$. In these cases the $n \times n$ matrix had entries equal to ± 1 to within the first several decimals and so by simple rounding of the entries it was straightforward to read off the corresponding cut. All the cuts that we extracted in this way achieved the value of the corresponding SDP (upper) bound and thus were indeed optimal.

Problem number	Spatial dim. and type of interactions	# rows	# cols	# layers	SDP1 time	SDP3 time
G2-01	2-D Gaussian	3	4	-	0:0:13	0:0:53
G2-02	2-D Gaussian	4	5	-	0:0:09	1:28:21
G2-03	2-D Gaussian	5	4	-	0:0:11	1:28:21
G2-04	2-D Gaussian	6	4	-	0:0:08	14:56:55
G2-05	2-D Gaussian	4	4	-	0:0:10	0:11:29
G3-01	3-D Gaussian	2	2	4	0:0:07	0:26:50
G3-02	3-D Gaussian	3	3	2	0:0:08	1:1:32
G3-03	3-D Gaussian	2	3	2	0:0:07	0:0:52
G3-04	3-D Gaussian	2	4	2	0:0:11	0:10:51
G3-05	3-D Gaussian	5	2	2	0:0:10	1:28:21
PM2-01	2-D ± 1 (50% are -1)	3	4	-	0:0:08	0:0:52
PM2-02	2-D ± 1 (40% are -1)	4	4	-	0:0:10	0:11:20
PM2-03	2-D ± 1 (50% are -1)	4	4	-	0:0:08	0:10:49
PM2-04	2-D ± 1 (60% are -1)	4	4	-	0:0:14	0:10:50
PM2-05	2-D ± 1 (50% are -1)	5	4	-	0:0:09	2:53:06
PM3-01	3-D ± 1 (40% are -1)	2	3	3	0:0:13	0:33:24
PM3-02	3-D ± 1 (45% are -1)	2	3	3	0:0:13	0:33:21
PM3-03	3-D ± 1 (50% are -1)	2	3	3	0:0:08	0:59:36
PM3-04	3-D ± 1 (55% are -1)	2	3	3	0:0:15	0:31:49
PM3-05	3-D ± 1 (60% are -1)	2	3	3	0:0:14	0:34:54

Table 3.3: Problem descriptions and time to compute the SDP bounds

Problem number	SDP1 bound	SDP1 ρ	SDP3 bound	SDP3 ρ	optimal X for SDP1 had $\lambda_2 \leq 10^{-6}$	optimal X for SDP3 had $\lambda_2 \leq 10^{-6}$
G2-01	9.3403	0.9825	9.1764	1.0000	No	Yes
G2-02	16.2890	0.9775	15.9219	1.0000	No	Yes
G2-03	11.1454	0.9666	10.7731	1.0000	No	Yes
G2-04	11.8862	0.9288	11.0394	1.0000	No	Yes
G2-05	9.6370	0.9985	9.6222	1.0000	No	Yes
G3-01	13.9879	0.9999	13.9878	1.0000	No	Yes
G3-02	18.1701	0.9720	17.6606	1.0000	No	Yes
G3-03	4.7052	0.9730	4.5780	1.0000	No	Yes
G3-04	8.0473	0.9192	7.3973	1.0000	No	Yes
G3-05	11.7420	0.9942	11.6744	1.0000	No	Yes
PM2-01	7.2087	0.8323	6.0000	1.0000	No	No
PM2-02	15.2087	0.9205	14.0000	1.0000	No	No
PM2-03	10.8416	0.7379	8.0000	1.0000	No	No
PM2-04	10.1431	0.9859	10.0000	1.0000	No	Yes
PM2-05	15.7962	0.8863	14.0000	1.0000	No	No
PM3-01	20.0623	0.9470	19.0000	1.0000	No	Yes
PM3-02	17.2188	0.8711	15.0000	1.0000	No	No
PM3-03	15.4263	0.9075	14.0000	1.0000	No	No
PM3-04	14.5197	0.9642	14.0000	1.0000	No	Yes
PM3-05	9.4926	0.8428	8.0000	1.0000	No	Yes

Table 3.4: Computed SDP bounds and their optimality

Next, we fixed one set of parameters and generated a batch of problems with that set of parameters. We made the choices:

- ±1 pairwise interactions: 5 rows, 4 columns, 50% negative interactions (see Table 3.5)
- Gaussian pairwise interactions: 5 rows, 4 columns (see Table 3.6)

For a closer look at the behaviour of the ranks of the optimal matrices, we also kept track of how the ranks of the computed optimal 20×20 matrices compared for both relaxations. The ranks were computed using a tolerance of 10^{-6}, which means that any eigenvalue with absolute value smaller than 10^{-6} was considered to be zero for the purpose of rank determination. An asterisk (*) indicates that an optimal cut could be read directly from the optimal matrix X by simple rounding of the entries.

Problem number	SDP1 bound	SDP1 ρ	SDP3 bound	SDP3 ρ	Rank of SDP1 opt. matrix X	Rank of SDP3 opt. matrix X
PM2-B01	14.2787	0.8404	12.0000	1.0000	3	13
PM2-B02	14.8775	0.9410	14.0000	1.0000	3	3
PM2-B03	14.8098	0.9453	14.0000	1.0000	3	2
PM2-B04	13.7604	0.8721	12.0000	1.0000	4	17
PM2-B05	15.5298	0.9015	14.0000	1.0000	3	4
PM2-B06	14.8426	0.9432	14.0000	1.0000	3	1 (*)
PM2-B07	15.5846	0.8983	14.0000	1.0000	3	5
PM2-B08	16.0907	0.8701	14.0000	1.0000	3	10
PM2-B09	14.1352	0.8489	12.0000	1.0000	3	15
PM2-B10	13.8398	0.8671	12.0000	1.0000	3	17
PM2-B11	14.7213	0.8151	12.1798	0.9852	3	17
PM2-B12	14.1478	0.8482	12.0000	1.0000	3	14
PM2-B13	14.6379	0.9564	14.0000	1.0000	4	2
PM2-B14	14.5681	0.8237	12.0746	0.9938	3	20
PM2-B15	14.0210	0.8559	12.0000	1.0000	3	17

Table 3.5: SDP bounds for 15 two-dimensional instances with ±1 pairwise interactions, 5 rows, 4 columns, and 50% negative interactions

Problem number	SDP1 bound	SDP1 ρ	SDP3 bound	SDP3 ρ	Rank of SDP1 opt. matrix X	Rank of SDP3 opt. matrix X
G2-B01	18.8588	0.9977	18.8155	1.0000	2	1 (*)
G2-B02	14.7421	0.9731	14.3457	1.0000	2	1 (*)
G2-B03	13.2008	0.9157	12.0874	1.0000	3	1 (*)
G2-B04	12.5660	0.9911	12.4538	1.0000	2	1 (*)
G2-B05	9.8149	0.9483	9.3077	1.0000	2	1 (*)
G2-B06	17.2963	0.9742	16.8504	1.0000	2	1 (*)
G2-B07	10.5096	0.9452	9.9339	1.0000	3	1 (*)
G2-B08	11.4777	0.9340	10.7200	1.0000	4	1 (*)
G2-B09	19.4582	0.9894	19.2519	1.0000	3	1 (*)
G2-B10	13.8900	0.9456	13.1340	1.0000	2	1 (*)
G2-B11	16.4840	0.9770	16.1048	1.0000	2	1 (*)
G2-B12	22.7340	0.9569	21.7535	1.0000	2	1 (*)
G2-B13	13.9023	0.9675	13.4504	1.0000	2	1 (*)
G2-B14	11.1122	0.9848	10.9436	1.0000	2	1 (*)
G2-B15	15.9040	0.9628	15.3119	1.0000	3	1 (*)

Table 3.6: SDP bounds for 15 two-dimensional instances with Gaussian pairwise interactions, 5 rows, 4 columns

To solve problems larger than $n = 20$, we used SBmethod [45]. We set the termination precision parameter (`-te`) to 10^{-4} but even so solving SDP3 often took several hours: in fact the time varied from a few minutes to over four hours depending on the problem.

This weaker precision requirement meant that the eigenvalues of the optimal matrices were not as clearly separated between zeros and non-zeros. Therefore the simple rank-determination procedure used previously did not work as well. Nonetheless, it was often possible to get an optimal cut from the optimal matrix computed by SBmethod for SDP3 by applying the following simple heuristic: Take the optimal matrix computed by SBmethod for SDP3 and round all its entries to ± 1 according to sign. Of course, zeros may occur, but we do not concern ourselves with this; we just wish to check how often this heuristic can extract an optimal cut. Then check whether the resulting matrix is positive semidefinite (there is no reason why this should be preserved after the rounding is done). If it is, then check its rank and, if the rank equals one, then extract the corresponding cut and check its optimality by comparing its value with the bound computed by SDP3. The rightmost column in the next two tables indicates whether this heuristic actually yielded an optimal cut.

For comparison purposes, we applied the same heuristic to SDP1. The results are presented in Tables 3.7 and 3.8.

Problem number	SDP1 bound	SDP1 ρ	SDP3 bound	SDP3 ρ	Does SDP1 *rounded* opt. matrix X remain $\succeq 0$ & yield opt. cut?	Does SDP3 *rounded* opt. matrix X remain $\succeq 0$ & yield opt. cut?
PM2-2B01	14.1385	0.8487	12.0033	0.9997	No	No
PM2-2B02	14.3700	0.9743	14.0002	1.0000	No	Yes
PM2-2B03	15.2119	0.9203	14.0011	0.9999	No	No
PM2-2B04	13.4624	0.8914	12.0013	0.9999	No	No
PM2-2B05	13.6693	0.8779	12.0003	1.0000	No	No
PM2-2B06	13.9242	0.8618	12.0011	0.9999	No	No
PM2-2B07	14.4899	0.9662	14.0010	0.9999	No	Yes
PM2-2B08	13.5102	0.8882	12.0002	1.0000	No	Yes
PM2-2B09	13.2993	0.9023	12.0007	0.9999	No	No
PM2-2B10	17.4753	0.9156	16.0015	0.9999	No	Yes
PM2-2B11	13.2813	0.9035	12.0015	0.9999	No	No
PM2-2B12	13.7794	0.8709	12.0010	0.9999	No	No
PM2-2B13	13.4571	0.8917	12.0001	1.0000	No	Yes
PM2-2B14	14.2686	0.9812	14.0001	1.0000	No	Yes
PM2-2B15	14.6953	0.9527	14.0002	1.0000	No	Yes
PM2-2B16	21.4629	0.9318	20.0022	0.9999	No	No
PM2-2B17	21.3496	0.9368	20.0012	0.9999	No	No
PM2-2B18	23.6477	0.9303	22.0056	0.9997	No	No
PM2-2B19	21.4248	0.9335	20.0021	0.9999	No	No
PM2-2B20	23.6510	0.9302	22.0016	0.9999	No	No
PM2-2B21	23.3862	0.9407	22.0015	0.9999	No	No
PM2-2B22	23.4821	0.9369	22.0020	0.9999	No	No
PM2-2B23	23.3058	0.9440	22.0043	0.9998	No	No
PM2-2B24	21.5622	0.9275	20.0056	0.9997	No	No
PM2-2B25	24.0359	0.9153	22.0018	0.9999	No	No
PM2-2B26	23.7228	0.9274	22.0025	0.9999	No	No
PM2-2B27	23.4800	0.9370	22.0038	0.9998	No	No
PM2-2B28	23.1365	0.9509	22.0037	0.9998	No	No
PM2-2B29	21.9249	0.9122	20.0039	0.9998	No	No
PM2-2B30	23.1911	0.9486	22.0036	0.9998	No	No

Table 3.7: Results for 30 two-dimensional instances with ±1 pairwise interactions, 4 rows, 6 columns; instances B01 to B15 have 60% negative interactions, and instances B16-B30 have 40% negative interactions

Problem	SDP1	SDP1	SDP3	SDP3	Does SDP1	Does SDP3
number	bound	ρ	bound	ρ	*rounded*	*rounded*
					opt. matrix X	opt. matrix X
					remain $\succeq 0$	remain $\succeq 0$
					& yield opt. cut?	& yield opt. cut?
G3-B01	20.7215	0.9220	19.1064	1.0000	No	Yes
G3-B02	20.6533	0.9081	18.7550	0.9999	No	Yes
G3-B03	23.2266	0.9407	21.8492	1.0000	No	Yes
G3-B04	27.5300	0.9160	25.2183	1.0000	No	Yes
G3-B05	30.4599	0.9401	28.6357	1.0000	No	Yes
G3-B06	27.5609	0.9592	26.4379	1.0000	No	Yes
G3-B07	20.6411	0.9311	19.2200	0.9999	No	Yes
G3-B08	16.3548	0.8731	14.2789	0.9999	No	Yes
G3-B09	23.3815	0.9066	21.1967	1.0000	No	Yes
G3-B10	26.6095	0.9458	25.1687	1.0000	No	Yes
G3-B11	26.9040	0.9757	26.2517	1.0000	No	Yes
G3-B12	16.6815	0.9461	15.7827	0.9999	No	Yes
G3-B13	22.6882	0.9459	21.4621	1.0000	No	Yes
G3-B14	27.1578	0.9513	25.8363	1.0000	No	Yes
G3-B15	25.7163	0.9935	25.5507	0.9999	No	Yes
G3-B16	27.3973	0.9859	27.0125	0.9999	No	Yes
G3-B17	19.9860	0.8980	17.9476	0.9999	No	Yes
G3-B18	25.8479	0.9677	25.0144	1.0000	No	Yes
G3-B19	25.4209	0.9339	23.7411	1.0000	No	Yes
G3-B20	20.5478	0.8931	18.3532	0.9999	No	Yes
G3-B21	20.1629	0.9719	19.5975	0.9999	No	Yes
G3-B22	20.7638	0.9910	20.5795	0.9999	No	Yes
G3-B23	27.5138	0.9273	25.5140	1.0000	No	Yes
G3-B24	18.0188	0.9242	16.6527	1.0000	No	Yes
G3-B25	20.7169	0.9399	19.4721	1.0000	No	Yes
G3-B26	23.4681	0.9694	22.7506	1.0000	No	Yes
G3-B27	26.9078	0.9495	25.5491	1.0000	No	Yes
G3-B28	22.3647	0.9263	20.7178	1.0000	No	Yes
G3-B29	19.2096	0.9626	18.4930	0.9999	No	Yes
G3-B30	31.2686	0.9629	30.1105	1.0000	No	Yes

Table 3.8: Results for 30 three-dimensional instances with Gaussian pairwise interactions, 3 rows, 3 columns, 3 layers

Note that for every problem in Table 3.7 where this simple heuristic failed, the SDP3 bound is very close to the optimal value of Max-Cut, i.e. $\rho \approx 1$. Therefore the computed optimal matrix $X^* \in F_n$ lies in, or very close to, an optimal face of the positive semidefinite cone in $\mathcal{S}^n$, and so a purification procedure similar to those for linear programming (see for example Lewis [69], Kortanek and Zhu [60]) might be able to find an optimal cut starting from X^* and confirm optimality with the value of the SDP3 bound.

One final point about the bounds provided by SDP3 is that if we solve SDP3 using an algorithm that solves the dual problem, such as the bundle method implemented in SBmethod, then it is possible to stop the algorithm before completion and obtain an upper bound on μ^*, the value of the maximum cut. Because we stop the algorithm before completion, such bounds can be computed for much larger problems than those presented so far and can improve on the bound yielded by SDP1, even when the latter is solved to optimality. Table 3.9 presents some evidence of this for problems with n varying between 80 and 108.

When computing provably optimal solutions for NP-hard problems such as Max-Cut, some kind of enumerative technique must usually be applied. In this context, it is often worthwhile to compute good bounds at the current node of the enumeration tree before branching. It may be worthwhile for such methods to invest the time required to get a very tight bound using SDP3 at the root node of the enumeration tree, and perhaps also at other branching nodes, to help reduce the combinatorial explosion in the enumerative process. Current and future research efforts on solving sparse structured SDPs (see [14, 27, 22, 23, 24, 25, 35, 36, 74]) may lead to efficient methods for computing the SDP3 bound for large instances of Max-Cut.

Dimension and type of interactions	# rows	# cols	# layers	Exact Max-Cut value μ^*	Exact SDP1 bound (time)	SDP3 bound after 2 hours	SDP3 bound after 4 hours	SDP3 bound after 6 hours
3-D ±1 (50% -1) $n = 80$	4	4	5	70	80.1634 (1 sec.)	79.1668	79.0373	78.8684
3-D ±1 (50% -1) $n = 90$	3	5	6	80	89.8366 (1 sec.)	89.6088	89.0331	88.8184
2-D ±1 (50% -1) $n = 95$	5	19	-	66	74.4966 (2 sec.)	74.6724	74.3339	74.1664
3-D ±1 (50% -1) $n = 105$	3	5	7	92	104.4611 (2 sec.)	112.3705	104.5821	103.8961
3-D Gaussian $n = 96$	4	4	6	82.89442	90.0685 (3 sec.)	87.4646	86.2858	85.8233
2-D Gaussian $n = 99$	9	11	-	55.34969	59.8389 (2 sec.)	60.5334	58.4295	57.8747
2-D Gaussian $n = 100$	10	10	-	55.59722	58.8150 (3 sec.)	59.2651	57.4884	56.9097
3-D Gaussian $n = 108$	3	4	9	71.94114	80.1510 (4 sec.)	83.2906	77.6150	76.5602

Table 3.9: Computational comparison of the bounds SDP1 and SDP3 on toroidal graphs for larger values of n (all times are as reported by SBmethod); the exact Max-Cut values were computed by Liers [70]

3.3.5 Summary of Empirical Observations

The computational results which we have presented for Ising spin glass problems lead to the following empirical observations:

1. SDP3 *always* found the value of the optimal cut for the problems with Gaussian distributed interactions.

2. SDP3 almost always found the value of the optimal cut for the problems with ± 1 interactions.

3. Furthermore, a very simple rounding procedure applied to the matrix X obtained by solving SDP3 seems to often yield an optimal rank-one matrix and hence an optimal cut: it found an optimal cut for *all* problems with Gaussian distributed interactions, and also for several of the problems with ± 1 interactions.

4. The remarkable tightness of the bounds obtained with SDP3 may make it worthwhile for enumerative techniques to invest the computational effort required to compute the bound at the root node of the enumeration tree, and perhaps also at other branching nodes, in order to reduce the combinatorial explosion in the enumerative process.

One interesting question is how to design a more sophisticated rounding procedure than the one used here. The rounding procedure should perhaps make use of all the information contained in the optimal matrix Y computed by SBmethod, instead of using only the matrix $X \in \mathcal{S}^n$ obtained from the first column of Y. Such a procedure might be related to the well-known randomized algorithm of Goemans and Williamson [41], although it is worth noting that their analysis does not apply to Ising spin glass problems which have negative pairwise interactions (edge weights). This question is the focus of continuing research.

The computational results presented in this chapter demonstrate the remarkably good bounds computed by the relaxation SDP3. The strength of SDP3 thus motivates our focus on this tighter relaxation in Chapters 4 and 5 where we study the geometry of the set F_n and connections to the ranks of the matrix Y feasible for SDP3.

Chapter 4

Rank Characterization of the Faces of C_3

In Section 2.5 we proved that $F_n \subseteq M_n$ for $n \geq 3$. Using well-known results of polyhedral theory, it immediately follows that $F_3 = C_3$, since the triangle inequalities define all the facets of C_3 (see for example Deza and Laurent [30, page 503]). The polytope C_3 is three-dimensional and, beside the face of dimension 3 that is C_3 itself, it has faces of dimensions 0 (vertices), 1, and 2 (facets).

In this chapter we prove that given $Y \in \mathcal{F}_3$, where $\mathcal{F}_3$ is the feasible set of SDP3 for the complete graph on 3 nodes, the rank of Y characterizes the dimension of the face of C_3 where $X = \mathrm{sMat}\,(Y_{1:6,0})$ lies. This shows explicitly the connection between the rank of the variable Y of the second lifting and the possible locations of the projected matrix X within C_3. The results we prove for $n = 3$ cast further light on how SDP3 captures all the structure of C_3, and furthermore they are stepping stones for studying similar rank relationships for $n \geq 4$ in Chapter 5.

4.1 Ranks of Matrices in Low Dimensional Faces of C_n

In this section we present some elementary facts about the cut polytope C_n. First we recall from Deza and Laurent [30] the definition of k-neighbourliness: Given $k \geq 1$, a polytope is said to be k-neighbourly if, for every subset of at most k vertices of the polytope, the convex hull of this subset of vertices is a face of the polytope. This implies, in particular, that every subset of k vertices of the polytope is affinely independent. In the proof of Lemma 4.1.1, we will make use of the fact that the cut polytope is 2-neighbourly. The 2-neighbourliness property follows from the fact that the cut polytope is, in fact, 3-neighbourly (see Deza and Laurent [30, page 540]), but was first proved independently by Barahona and Mahjoub [13].

The results in Lemma 4.1.1 are elementary and reflect well-known results about the cut polytope. However, because we study the cut polytope in the matrix space $\mathcal{S}^n$ rather than in the vector space $\Re^{t(n-1)}$, these results are not explicitly stated in the literature to the best of our knowledge. Hence we state and prove them for completeness.

Lemma 4.1.1 *Let $v_1, \ldots, v_{2^{n-1}}$ be elements of $\{\pm 1\}^n$ that represent the 2^{n-1} distinct cuts possible for the complete graph on n nodes, $n \geq 3$. By definition,*

$$C_n = conv\{v_i v_i^T, \quad i = 1, \ldots, 2^{n-1}\}.$$

Then

1. *X is a vertex of C_n if and only if $X = v_i\, v_i^T$ for some $i \in \{1, \ldots, 2^{n-1}\}$.*

2. *X is in the relative interior of a face of C_n of dimension 1 if and only if $X = \alpha\, v_i\, v_i^T + (1-\alpha)\, v_j\, v_j^T$ for some $\alpha \in (0,1)$ and $\{i,j\} \subseteq \{1, \ldots, 2^{n-1}\}$, $i \neq j$.*

Proof: To prove the first claim, note that if X is a vertex of C_n, then X must be a generator of the convex hull, and thus $X = v_i\, v_i^T$ for some $i \in \{1, \ldots, 2^{n-1}\}$.

The converse is also straightforward to prove. For completeness, and because the proof illustrates some interesting aspects of the structure of C_n, we present a proof that follows the approach employed in Laurent and Poljak [64] to prove Theorem 4.2.1 below.

Suppose $\bar{X} = v_i\, v_i^T$. It suffices to prove that the normal cone $\mathcal{N}(C_n, \bar{X})$ is full dimensional, i.e. has dimension $t(n-1) = \begin{pmatrix} n \\ 2 \end{pmatrix}$. For each pair of indices i, j, $1 \leq i < j \leq n$, define the matrix

$$B_{ij} := \begin{cases} E_{ij}, & \text{if } \bar{X}_{ij} = 1 \\ -E_{ij}, & \text{if } \bar{X}_{ij} = -1. \end{cases}$$

Then, for any $X \in C_n$,

$$\begin{aligned} \left\langle B_{ij}, X - \bar{X} \right\rangle \leq 0 \;\; &\Leftrightarrow \;\; \langle B_{ij}, X \rangle \leq \left\langle B_{ij}, \bar{X} \right\rangle \\ &\Leftrightarrow \;\; \bar{X}_{ij}\, X_{ij} \leq 1, \end{aligned}$$

which holds since every element of X has absolute value at most 1. Hence, $B_{ij} \in \mathcal{N}(C_n, \bar{X})$.

Since the set $\{B_{ij} \mid 1 \leq i < j \leq n\}$ is linearly independent and has dimension $t(n-1)$, it follows that $\mathcal{N}(C_n, \bar{X})$ is full dimensional. Hence $\bar{X}$ is a vertex of C_n.

To prove the second claim, first assume that X is in the relative interior of a face of dimension 1. Then by definition of such a face, X is the strict convex combination of two vertices, i.e. X is formed exactly as claimed.

Conversely, if $X = \alpha\, v_i\, v_i^T + (1-\alpha)\, v_j\, v_j^T$, then the fact that the cut polytope is 2-neighbourly implies that the convex hull of the vertices $v_i\, v_i^T$ and $v_j\, v_j^T$ is a (one-dimensional) face of C_n, and hence X is in the relative interior of a face of dimension 1. ■

4.2 Ranks of Matrices in $\mathcal{E}_n$

For the elliptope, only the vertices of C_n are completely characterized by the rank of X. Laurent and Poljak proved the following theorem (expressed here using our notation):

Theorem 4.2.1 *(Laurent and Poljak, Theorem 2.5 in Laurent and Poljak [64])* *The elliptope $\mathcal{E}_n$ has precisely 2^{n-1} vertices, each of the form $v_i\, v_i^T$ for some $i \in \{1, \ldots, 2^{n-1}\}$.*

By Lemma 4.1.1, the vertices of $\mathcal{E}_n$ are thus exactly the vertices of C_n. However, the correspondence between rank and geometric structure breaks down as soon as we consider matrices with rank equal to 2. Indeed, for $n = 3$ consider the matrix

$$\tilde{X} = \begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 1 & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & 1 \end{pmatrix}.$$

It can easily be checked that $\tilde{X} \in \mathcal{E}_3$ and has rank equal to two. However,

$$\tilde{X}_{12} + \tilde{X}_{13} + \tilde{X}_{23} = -\frac{3}{2} < -1$$

and so $\tilde{X}$ violates the triangle inequality $X_{12} + X_{13} + X_{23} \geq -1$. Hence $\tilde{X} \notin C_3$.

In this chapter we prove Theorem 4.4.7 which shows that, in contrast to the elliptope case, whenever $Y \in \mathcal{F}_3$ and $X = \mathrm{sMat}\,(Y_{1:6,0})$, the rank of Y completely characterizes the dimension of the face of C_3 in which X lies. In the next section, we prove some results about the ranks of the matrices in C_3. These results will be used to prove the aforementioned characterization.

4.3 Ranks of Matrices in C_3

For the rest of this chapter, we focus our attention on the polytope C_3 and the ranks of the matrices therein. First observe that there are four distinct cuts for the complete graph K_3, which we represent by the vectors

$$v_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, v_2 = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}, v_3 = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}, \quad \text{and} \quad v_4 = \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix}.$$

It is straightforward to check that every subset of three (or fewer) of these vectors is linearly independent. Recall that $X \in C_3$ if and only if

$$X = \sum_{i=1}^{4} \alpha_i \, v_i \, v_i^T, \sum_{i=1}^{4} \alpha_i = 1, \alpha_i \geq 0, i = 1, 2, 3, 4. \tag{4.1}$$

Lemma 4.3.1 *$X \in C_3$ is positive definite if and only if every convex combination in (4.1) that equals X has at least three α_i positive.*

Proof: Every $X \in C_3$ is equal to a sum of matrices each with rank (at most) one, therefore

$$\operatorname{rank}(X) = \operatorname{rank}\left(\sum_{i=1}^{4} \alpha_i \, v_i \, v_i^T\right) \leq \sum_{i=1}^{4} \operatorname{rank}\left(\alpha_i \, v_i \, v_i^T\right).$$

If X is positive definite then $\operatorname{rank}(X) = 3$, therefore at least three of the matrices $\alpha_i \, v_i \, v_i^T$ must be non-zero. Hence at least three α_i are positive in any convex combination of vertices equal to X.

To prove sufficiency, suppose that every convex combination of vertices equal to X has at least three α_i positive. We prove that X is positive definite by contradiction.

Suppose there exists $w \in \Re^3$ such that $w \neq 0$ and $X\,w = 0$. Because every subset of three v_i's forms a basis for $\Re^3$, we may assume without loss of generality that $v_1^T w \neq 0$ and $v_2^T w \neq 0$ (that is, at least two v_i's are not orthogonal to w). Then

$$X\,w = \alpha_1\,v_1\,(v_1^T w) + \alpha_2\,v_2\,(v_2^T w) + \alpha_3\,v_3\,(v_3^T w) + \alpha_4\,v_4\,(v_4^T w) = 0 \tag{4.2}$$

therefore

$$\alpha_1\,(v_1^T w)\,v_1^T = -\left[\alpha_2\,(v_2^T w)\,v_2^T + \alpha_3\,(v_3^T w)\,v_3^T + \alpha_4\,(v_4^T w)\,v_4^T\right]$$

and, multiplying by w on the right, we obtain that

$$\alpha_1\,(v_1^T w)^2 = -\left[\alpha_2\,(v_2^T w)^2 + \alpha_3\,(v_3^T w)^2 + \alpha_4\,(v_4^T w)^2\right].$$

Now observe that each term $\alpha_i\,(v_i^T w)^2$ is non-negative. Hence both sides of the equation must equal zero. Since $v_1^T w \neq 0$, $\alpha_1 = 0$ must hold, and similarly $\alpha_2 = 0$ must also hold. This contradicts our assumption that every convex combination of vertices equal to X has at least three α_i positive. ■

Lemma 4.3.2 *$X \in C_3$ has rank equal to 1 if and only if $X = v_i\,v_i^T$ for some $i \in \{1, 2, 3, 4\}$.*

Proof: Sufficiency is immediate. So suppose $X \in C_3$ has rank equal to 1. By Lemma 4.3.1, every linear combination of vertices equal to X has at most two α_i

non-zero. To get a contradiction, suppose that some linear combination equal to X has two α_i non-zero. Further suppose $\alpha_3 = 0$ and $\alpha_4 = 0$ without loss of generality. Now proceed as follows:

1. Choose $w_1 \in \Re^3$ such that $v_2^T\, w_1 = 0$ and $v_1^T\, w_1 \neq 0$ (this is possible because v_1 and v_2 are linearly independent). Then

   $$X\, w_1 = \alpha_1\, (v_1^T w_1)\, v_1 + \alpha_2\, (v_2^T w_1)\, v_2 = \alpha_1\, (v_1^T w_1)\, v_1,$$

   where $\alpha_1\, (v_1^T w_1) \neq 0$. Hence v_1 is in the range of X.

2. Similarly, choose $w_2 \in \Re^3$ such that $v_1^T\, w_2 = 0$ and $v_2^T\, w_2 \neq 0$, and conclude that v_2 is in the range of X.

Since $\{v_1, v_2\}$ form a linearly independent set, the range of X has dimension at least 2. But the rank of X equals the dimension of its range, hence $\operatorname{rank} X \geq 2$, which is a contradiction. This concludes the proof. ■

Corollary 4.3.3 *$X \in C_3$ has rank equal to 2 if and only if there exists $\alpha \in (0, 1)$ and $i \neq j$, $i, j \in \{1, 2, 3, 4\}$ such that $X = \alpha\, v_i\, v_i^T + (1 - \alpha)\, v_j\, v_j^T$.*

Proof: Again sufficiency is immediate by the linear independence of $\{v_i, v_j\}$ for $i \neq j$, $i, j \in \{1, 2, 3, 4\}$. And, if $\operatorname{rank} X = 2$ then

- by Lemma 4.3.1, every convex combination equal to X has at most two α_i non-zero;

- by Lemma 4.3.2, every convex combination equal to X has at least two α_i non-zero.

Hence the result. ■

We summarize our results on the ranks of matrices $X \in C_3$ in the following theorem:

Theorem 4.3.4 *Let $X \in C_3$, i.e.*

$$X = \sum_{i=1}^{4} \alpha_i \, v_i \, v_i^T, \;\; \sum_{i=1}^{4} \alpha_i = 1, \; \alpha_i \geq 0, i = 1, 2, 3, 4.$$

Then

- $\operatorname{rank} X = 1 \Leftrightarrow X$ *is a vertex of C_3;*
- $\operatorname{rank} X = 2 \Leftrightarrow X$ *is in the relative interior of a face of C_3 of dimension 1.*
- $\operatorname{rank} X = 3 \Leftrightarrow$ *every convex combination of vertices equal to X has at least three α_i positive.*

Proof: This result follows immediately from Lemmas 4.1.1, 4.3.1, 4.3.2, and Corollary 4.3.3. ■

We see that while the vertices and one-dimensional faces of C_3 are completely characterized by the rank of X, knowledge of the rank of X is insufficient to distinguish whether X lies in the relative interior of a facet (face of dimension 2) or in the interior of the cut polytope. We now prove that the rank of the variable Y in the second lifting achieves this purpose.

4.4 Rank Connections Between $\mathcal{F}_3$ and C_3

In order to simplify the proofs, for the rest of this chapter and in Chapter 5 we will work with the matrix Y projected onto the minimal face. Note that the columns of Y that are not explicitly considered in the minimal face, in this case columns 1,3, and 6, are all equal to column 0. In general, whenever $Y \in \mathcal{F}_n$, the columns indexed by $T(k,k)$ are all equal to column 0, since positive semidefiniteness of the 3×3 principal submatrix of Y corresponding to the rows and columns $\{0, T(k,k), T(i,j)\}$ gives

$$\begin{pmatrix} 1 & 1 & Y_{0,T(i,j)} \\ 1 & 1 & Y_{T(k,k),T(i,j)} \\ Y_{0,T(i,j)} & Y_{T(k,k),T(i,j)} & 1 \end{pmatrix} \succeq 0 \Rightarrow Y_{T(i,j),T(k,k)} = Y_{T(i,j),0}.$$

The implication follows by symmetry and by Lemma 4.4.2 proved below. Hence, the rank of $Y \in \mathcal{F}_n$ always equals the rank of its projection onto the minimal face, and all the upcoming results in this chapter and the next hold identically for the unprojected matrix Y.

Let us therefore set the notation (for the case $n = 3$):

$$X = \begin{pmatrix} 1 & X_{12} & X_{13} \\ X_{12} & 1 & X_{23} \\ X_{13} & X_{23} & 1 \end{pmatrix} \quad \text{and} \quad Y = \begin{pmatrix} 1 & X_{12} & X_{13} & X_{23} \\ X_{12} & 1 & X_{23} & X_{13} \\ X_{13} & X_{23} & 1 & X_{12} \\ X_{23} & X_{13} & X_{12} & 1 \end{pmatrix}.$$

We will make use of the following well-known result from matrix theory (see for example Theorem 4.3.8 in Horn and Johnson [51]):

Let A be a given $n \times n$ Hermitian matrix, let y be a given n-vector, and let a be a given real number. Let $\hat{A}$ be the $(n+1) \times (n+1)$ Hermitian matrix obtained by bordering A with y and a as follows:

$$\hat{A} \equiv \begin{pmatrix} A & y \\ y^T & a \end{pmatrix}.$$

Let the eigenvalues of A and $\hat{A}$ be denoted by $\{\lambda_i(A)\}$ and $\{\lambda_i(\hat{A})\}$ respectively, and assume that they have been arranged in increasing order $\lambda_1(A) \leq \ldots \leq \lambda_n(A)$ and $\lambda_1(\hat{A}) \leq \ldots \leq \lambda_{n+1}(\hat{A})$. Then

$$\lambda_1(\hat{A}) \leq \lambda_1(A) \leq \lambda_2(\hat{A}) \leq \lambda_2(A) \leq \ldots \leq \lambda_{n-1}(A) \leq \lambda_n(\hat{A}) \leq \lambda_n(A) \leq \lambda_{n+1}(\hat{A}).$$

The first lemma of this section establishes that the matrices Y with rank equal to 1 map to the vertices of C_3.

Lemma 4.4.1 *Let $Y \in \mathcal{F}_3$ and $X = \operatorname{sMat}(Y_{1:6,0})$. Then*

$$\operatorname{rank} Y = 1 \Leftrightarrow \operatorname{rank} X = 1.$$

Proof: First suppose $\operatorname{rank} Y = 1$. Then $\operatorname{rank} X \leq 1$, since X is a principal submatrix of Y. But $X \neq 0$, hence $\operatorname{rank} X = 1$.

Now suppose $\operatorname{rank} X = 1$. Since $X \succeq 0$, Theorem 2.1.1 implies that $X \in \{\pm 1\}^{3\times 3}$. Hence, $Y \in \{\pm 1\}^{4\times 4}$ and since $Y \in \mathcal{F}_3$ implies $Y \succeq 0$, Theorem 2.1.1 implies $\operatorname{rank} Y = 1$. ■

Having settled the vertex case, we proceed to prove some simple but very useful facts about the higher rank occurrences of Y. First we state and prove a simple technical lemma that will be useful in this chapter and the next.

Lemma 4.4.2 *Suppose*

$$\begin{pmatrix} 1 & a & b \\ a & 1 & c \\ b & c & 1 \end{pmatrix} \succeq 0.$$

Then

1. *If $a^2 = 1$ then $b = a\,c$;*

2. *If $b^2 = 1$ then $a = b\,c$;*

3. *If $c^2 = 1$ then $a = c\,b$.*

Proof: We prove only the case $b^2 = 1$. The other cases are proved similarly, or may be reduced to this case by symmetrically permuting rows and columns of the matrix.

Positive semidefiniteness of the matrix implies that its determinant is nonnegative, therefore

$$(1 - c^2) - a(a - bc) + b(ac - b) \geq 0$$

which implies

$$-(b^2\, c^2 + a^2 - 2\, a\, b\, c) \geq 0,$$

since $b^2 = 1$. Equivalently,

$$-(a - b\, c)^2 \geq 0$$

holds, and hence $a = b\, c$. ■

We note that all three implications of Lemma 4.4.2 may hold simultaneously. This is exemplified by the matrix

$$\begin{pmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}^T .$$

The next lemma establishes the possible ranks of Y whenever one or more of the elements of X have absolute value equal to 1.

Lemma 4.4.3 *Let $Y \in \mathcal{F}_3$ and $X = \operatorname{sMat}(Y_{1:6,0})$. Then the following hold:*

1. *If $|X_{ij}| = 1$ for some pair $i, j, i \neq j$, then $\operatorname{rank} Y \leq 2$.*

2. *If exactly one X_{ij} has absolute value equal to 1, then $\operatorname{rank} Y = 2$.*

3. *If two or three off-diagonal elements of X have absolute value equal to 1, then $\operatorname{rank} Y = 1$.*

Proof: The proof considers the possible cases with at least one element $X_{ij} \in \{\pm 1\}$. If all three of X_{12}, X_{13}, and X_{23} are equal to ±1 then by Theorem 2.1.1, $\operatorname{rank} Y = 1$. If two of them are equal to ±1, let us suppose without loss of generality that $|X_{12}| = 1$ and $|X_{13}| = 1$. If $X_{12} = 1$ then the leading 3×3 principal submatrix of Y is positive semidefinite, so applying Lemma 4.4.2,

$$\begin{pmatrix} 1 & 1 & X_{13} \\ 1 & 1 & X_{23} \\ X_{13} & X_{23} & 1 \end{pmatrix} \succeq 0 \Rightarrow X_{23} = X_{13},$$

and thus X_{23} also equals ±1, therefore $\operatorname{rank} Y = 1$ again. Similarly, $X_{12} = -1 \Rightarrow X_{23} = -X_{13} \in \{\pm 1\}$ and $\operatorname{rank} Y = 1$.

Finally, if exactly one of them equals ±1, suppose again without loss of generality that $|X_{12}| = 1$. If $X_{12} = 1$ then $X_{23} = X_{13}$ but neither equals ± 1 and

$$Y = \begin{pmatrix} 1 & 1 & X_{13} & X_{13} \\ 1 & 1 & X_{13} & X_{13} \\ X_{13} & X_{13} & 1 & 1 \\ X_{13} & X_{13} & 1 & 1 \end{pmatrix}$$

which has rank equal to two. Similarly, $X_{12} = -1 \Rightarrow X_{23} = -X_{13}$ and again $\operatorname{rank} Y = 2$. ■

Lemma 4.4.4, which we now prove, is perhaps a surprising result. It shows that if $X \in F_3$ has rank equal to two, then one of its elements has absolute value equal to one. Such a statement does not hold for the elliptope $\mathcal{E}_3$, as is demonstrated for example by the matrix $\tilde{X}$ presented in Section 4.2. Lemma 4.4.4 will also be very useful in the sequel.

Lemma 4.4.4 *If $X \in F_3$ and $\operatorname{rank} X = 2$ then $|X_{ij}| = 1$ for some $i \neq j$.*

Proof: If X has rank equal to 2 then X is singular. Clearly not all three off-diagonal entries of X can equal zero. Suppose two elements of X equal zero, and without loss of generality, that $X_{12} = 0$ and $X_{13} = 0$, so that

$$X = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & X_{23} \\ 0 & X_{23} & 1 \end{pmatrix}.$$

Since X is singular, $\det X = 0$ holds and so

$$1 - X_{23}^2 = 0,$$

which implies $X_{23} = \pm 1$.

Now suppose at most one element of X equals zero. Since X is singular, there is a linear dependency between its three columns and therefore there exist two scalars α and β, not both zero, such that

$$1 = \alpha\, X_{12} + \beta\, X_{13} \tag{4.3}$$

$$X_{12} = \alpha + \beta\, X_{23} \tag{4.4}$$

$$X_{13} = \alpha\, X_{23} + \beta \tag{4.5}$$

If $\alpha = 0$ then equation (4.5) implies $X_{13} = \beta$, therefore equation (4.3) implies $X_{13}^2 = 1$ and we are done. Similarly, if $\beta = 0$ then equation (4.4) implies $X_{12} = \alpha$, therefore equation (4.3) implies $X_{12}^2 = 1$ and we are again done.

Therefore we may suppose that $\alpha \neq 0$ and $\beta \neq 0$. Using equations (4.4) and (4.5) to substitute for X_{12} and X_{13} in equation (4.3) and solving for X_{23}, we obtain the equation

$$X_{23} = \frac{1 - \alpha^2 - \beta^2}{2\, \alpha\, \beta}. \tag{4.6}$$

Now, $\operatorname{rank} X = 2$ implies $\lambda_1(X) = 0$ and therefore, by the interlacing of eigenvalues, $\lambda_1(Y) = 0$. So $\det Y = 0$ and substituting for X_{12} and X_{13} in Y using (4.4) and (4.5), we have

$$\det \begin{pmatrix} 1 & \alpha+\beta\,X_{23} & \alpha\,X_{23}+\beta & X_{23} \\ \alpha+\beta\,X_{23} & 1 & X_{23} & \alpha\,X_{23}+\beta \\ \alpha\,X_{23}+\beta & X_{23} & 1 & \alpha+\beta\,X_{23} \\ X_{23} & \alpha\,X_{23}+\beta & \alpha+\beta\,X_{23} & 1 \end{pmatrix} = 0$$

which, upon expanding the determinant, is equivalent to

$$(1-2\,\alpha^2-2\,\beta^2+\alpha^4+\beta^4-2\,\alpha^2\,\beta^2)\,(1-2\,X_{23}^2+X_{23}^4)=0. \tag{4.7}$$

Equation (4.6) implies

$$\begin{aligned} X_{23}^2 &= \frac{(1-\alpha^2-\beta^2)^2}{4\,\alpha^2\,\beta^2} \\ &= \frac{1-2\,\alpha^2-2\,\beta^2+\alpha^4+\beta^4+2\,\alpha^2\,\beta^2}{4\,\alpha^2\,\beta^2} \end{aligned}$$

and therefore we have the equation

$$1-X_{23}^2=\frac{4\,\alpha^2\,\beta^2-(1-2\,\alpha^2-2\,\beta^2+\alpha^4+\beta^4+2\,\alpha^2\,\beta^2)}{4\,\alpha^2\,\beta^2}$$

from which it follows that

$$-4\,\alpha^2\,\beta^2\,(1-X_{23}^2)=1-2\,\alpha^2-2\,\beta^2+\alpha^4+\beta^4-2\,\alpha^2\,\beta^2. \tag{4.8}$$

Substituting for the first term on the left-hand side of equation (4.7) using equation (4.8), we conclude that

$$(-4\,\alpha^2\,\beta^2\,(1 - X_{23}^2))\,(1 - 2\,X_{23}^2 + X_{23}^4) = 0$$

or, equivalently,

$$-4\,\alpha^2\,\beta^2\,(1 - X_{23}^2)^3 = 0.$$

Since $\alpha \neq 0$ and $\beta \neq 0$, we must have $1 - X_{23}^2 = 0 \Leftrightarrow X_{23} = \pm 1$. This concludes the proof. ■

The next lemma shows that the rank correspondence between X and Y also holds for the rank-two case.

Lemma 4.4.5 *Let $Y \in \mathcal{F}_3$ and $X = \mathrm{sMat}\,(Y_{1:6,0})$. Then*

$$\operatorname{rank} Y = 2 \Leftrightarrow \operatorname{rank} X = 2.$$

Proof: If $\operatorname{rank} Y = 2$ then

$$\lambda_1(Y) \leq \lambda_1(X) \leq \lambda_2(Y) \leq \lambda_2(X) \leq \lambda_3(Y) \leq \lambda_3(X) \leq \lambda_4(Y),$$

and $\lambda_1(Y) \geq 0$ by positive semidefiniteness.

Hence,

$$\operatorname{rank} Y = 2 \quad \Rightarrow \quad \lambda_2(Y) = 0$$

$$\begin{aligned} &\Rightarrow \quad \lambda_1(X) = 0 \\ &\Rightarrow \quad \operatorname{rank} X \leq 2. \end{aligned}$$

But by Lemma 4.4.1, $\operatorname{rank} X = 1 \Leftrightarrow \operatorname{rank} Y = 1$. Hence $\operatorname{rank} X = 2$.

Conversely, suppose that $\operatorname{rank} X = 2$. By Lemma 4.4.4, $X_{ij} = \pm 1$ for some pair $i, j, i \neq j$, and by Lemma 4.4.3 this implies $\operatorname{rank} Y \leq 2$. Finally, by Lemma 4.4.1, the rank of Y cannot equal 1 if $\operatorname{rank} X = 2$. Hence $\operatorname{rank} Y = 2$. ■

The only remaining possible value for $\operatorname{rank} X$ is 3, i.e. X is full-rank and thus lies in the relative interior of $\mathcal{E}_3$. This corresponds to Y having rank either 3 or 4. The next lemma shows that the rank of Y being equal to 3 characterizes the cases where one of the triangle inequalities holds tightly, and thus X lies in the relative interior of a facet of C_3.

Lemma 4.4.6 *Let $Y \in \mathcal{F}_3$ and $X = \operatorname{sMat}(Y_{1:6,0})$. Then*

$\operatorname{rank} Y = 3 \quad \Leftrightarrow \quad \operatorname{rank} X = 3$ *and one triangle inequality holds with equality.*

Proof: First suppose that $\operatorname{rank} X = 3$ and that

$$\alpha X_{12} + \beta X_{13} + \gamma X_{23} + 1 = 0$$

holds for given $\alpha, \beta, \gamma \in \{\pm 1\}$ such that $\alpha\,\beta\,\gamma = 1$. Then

$$\begin{aligned} Y \begin{pmatrix} 1 \\ \alpha \\ \beta \\ \gamma \end{pmatrix} &= \begin{pmatrix} 1 + \alpha\,X_{12} + \beta\,X_{13} + \gamma\,X_{23} \\ X_{12} + \alpha + \beta\,X_{23} + \gamma\,X_{13} \\ X_{13} + \alpha\,X_{23} + \beta + \gamma\,X_{12} \\ X_{23} + \alpha\,X_{13} + \beta\,X_{12} + \gamma \end{pmatrix} \\ &= \begin{pmatrix} 1 + \alpha\,X_{12} + \beta\,X_{13} + \gamma\,X_{23} \\ \frac{1}{\alpha}(\alpha\,X_{12} + \alpha^2 + \alpha\,\beta\,X_{23} + \alpha\,\gamma\,X_{13}) \\ \frac{1}{\beta}(\beta\,X_{13} + \alpha\,\beta\,X_{23} + \beta^2 + \beta\,\gamma\,X_{12}) \\ \frac{1}{\gamma}(\gamma\,X_{23} + \alpha\,\gamma\,X_{13} + \beta\,\gamma\,X_{12} + \gamma^2) \end{pmatrix} \\ &= 0, \end{aligned}$$

since $\alpha^2 = 1$, $\beta^2 = 1$, $\gamma^2 = 1$, and $\alpha\,\beta = \frac{1}{\gamma} = \gamma$, $\alpha\,\gamma = \frac{1}{\beta} = \beta$, $\beta\,\gamma = \frac{1}{\alpha} = \alpha$.

Since

$$\begin{pmatrix} 1 \\ \alpha \\ \beta \\ \gamma \end{pmatrix} \neq 0,$$

it follows that Y is singular, and so $\operatorname{rank} Y \leq 3$. Since $\operatorname{rank} X = 3$, by Lemmas 4.4.1 and 4.4.5, we conclude that $\operatorname{rank} Y = 3$.

Conversely, suppose that $\operatorname{rank} Y = 3$. By Lemmas 4.4.1 and 4.4.5, it follows that $\operatorname{rank} X = 3$. It remains to prove that one triangle inequality holds with equality.

Since $\operatorname{rank} Y = 3$, performing Gaussian elimination on Y will yield a matrix with the elements of the last row all equal to zeros. We will perform Gaussian elimination on Y, but before proceeding, we first observe that $\operatorname{rank} X = 3$ implies X is positive definite and thus all its principal submatrices are nonsingular. Specifically considering the top left 2×2 principal submatrix implies that $1 - X_{12}^2 \neq 0$. Furthermore,

$$\det X = 1 - X_{12}^2 - X_{13}^2 - X_{23}^2 + 2\, X_{12}\, X_{13}\, X_{23} > 0$$

since X is positive definite. Because we also have

$$\begin{aligned} (1 - X_{13}^2) - \frac{(X_{23} - X_{12}\, X_{13})^2}{1 - X_{12}^2} = 0 \quad &\Leftrightarrow \quad 1 - X_{12}^2 - X_{13}^2 - X_{23}^2 + 2\, X_{12}\, X_{13}\, X_{23} = 0 \\ &\Leftrightarrow \quad \det X = 0, \end{aligned}$$

and the latter does not hold, we conclude that

$$(1 - X_{13}^2) - \frac{(X_{23} - X_{12}\, X_{13})^2}{1 - X_{12}^2} \neq 0. \tag{4.9}$$

We can now perform Gaussian elimination on Y. Applying elementary row operations, we reduce Y to the form

$$\begin{pmatrix} 1 & X_{12} & X_{13} & X_{23} \\ 0 & 1-X_{12}^2 & \Delta_{231} & \Delta_{132} \\ 0 & 0 & (1-X_{13}^2)-\frac{\Delta_{231}^2}{1-X_{12}^2} & \Delta_{123}-\frac{\Delta_{231}\,\Delta_{132}}{1-X_{12}^2} \\ 0 & 0 & 0 & \left((1-X_{23}^2)-\frac{\Delta_{132}^2}{1-X_{12}^2}\right)-\frac{\left(\Delta_{123}-\frac{\Delta_{132}\,\Delta_{231}}{1-X_{12}^2}\right)^2}{\left((1-X_{13}^2)-\frac{\Delta_{231}^2}{1-X_{12}^2}\right)} \end{pmatrix}$$

where we use the notation

$$\Delta_{ijk} := X_{ij} - X_{ik}\,X_{kj}$$

for convenience.

So, $\operatorname{rank} Y = 3$ implies

$$\left((1-X_{23}^2)-\frac{\Delta_{132}^2}{1-X_{12}^2}\right)-\frac{\left(\Delta_{123}-\frac{\Delta_{132}\,\Delta_{231}}{1-X_{12}^2}\right)^2}{\left((1-X_{13}^2)-\frac{\Delta_{231}^2}{1-X_{12}^2}\right)}=0$$

$\Leftrightarrow$

$$\left((1-X_{23}^2)-\frac{\Delta_{132}^2}{1-X_{12}^2}\right)\left((1-X_{13}^2)-\frac{\Delta_{231}^2}{1-X_{12}^2}\right)=\left(\Delta_{123}-\frac{\Delta_{132}\,\Delta_{231}}{1-X_{12}^2}\right)^2$$

$\Leftrightarrow$

$$\frac{[(1-X_{23}^2)(1-X_{12}^2)-(X_{13}-X_{12}\,X_{23})^2][(1-X_{13}^2)(1-X_{12}^2)-(X_{23}-X_{12}\,X_{13})^2]}{(1-X_{12}^2)^2}$$

$$= \frac{[(1 - X_{12}^2)(X_{12} - X_{13}\,X_{23}) - (X_{13} - X_{12}\,X_{23})(X_{23} - X_{12}\,X_{13})]^2}{(1 - X_{12}^2)^2}$$

$\Leftrightarrow$

$$(1 - X_{12}^2 - X_{13}^2 - X_{23}^2 + 2\,X_{12}\,X_{13}\,X_{23})^2 = (X_{12} - 2\,X_{13}\,X_{23} - X_{12}^3 + X_{12}\,X_{13}^2 + X_{12}\,X_{23}^2)^2$$

and therefore *exactly* one of the following equalities holds:

$$(1 - X_{12}^2 - X_{13}^2 - X_{23}^2 + 2\,X_{12}\,X_{13}\,X_{23}) = (X_{12} - 2\,X_{13}\,X_{23} - X_{12}^3 + X_{12}\,X_{13}^2 + X_{12}\,X_{23}^2) \tag{4.10}$$

or

$$(1 - X_{12}^2 - X_{13}^2 - X_{23}^2 + 2\,X_{12}\,X_{13}\,X_{23}) = -(X_{12} - 2\,X_{13}\,X_{23} - X_{12}^3 + X_{12}\,X_{13}^2 + X_{12}\,X_{23}^2) \tag{4.11}$$

Both cannot hold because the left-hand side equals $\det X$ and is therefore non-zero.

Case 1: Equation (4.10) holds:

In this case, we have that equation (4.10) holds if and only if

$$1 - X_{12}^2 - X_{12} + X_{12}^3 - (X_{13}^2 + X_{23}^2 - 2\,X_{13}\,X_{23}) + X_{12}\,(2\,X_{13}\,X_{23} - X_{13}^2 - X_{23}^2) = 0$$

$\Leftrightarrow$

$$(1 - X_{12}^2)(1 - X_{12}) - (X_{13} - X_{23})^2 - X_{12}\,(X_{13} - X_{23})^2 = 0$$

$\Leftrightarrow$

$$(1-X_{12})^2(1+X_{12})-(X_{13}-X_{23})^2(1+X_{12})=0.$$

Since we know $X_{12}\neq -1$, the last equation is equivalent to

$$(1-X_{12})^2=(X_{13}-X_{23})^2.$$

Therefore we have two subcases: either

$$1-X_{12}=X_{13}-X_{23} \tag{4.12}$$

or

$$1-X_{12}=-(X_{13}-X_{23}). \tag{4.13}$$

If (4.12) holds, then the triangle inequality $-X_{12}-X_{13}+X_{23}\geq -1$ holds with equality. If (4.13) holds, then the triangle inequality $-X_{12}+X_{13}-X_{23}\geq -1$ holds with equality. Hence if equation (4.10) holds, we have indeed one triangle inequality that holds with equality.

Case 2: Equation (4.11) holds:

In this case, we have that equation (4.11) holds if and only if

$$1-X_{12}^2+X_{12}-X_{12}^3-(X_{13}^2+X_{23}^2+2\,X_{13}\,X_{23})+X_{12}\,(2\,X_{13}\,X_{23}+X_{13}^2+X_{23}^2)=0$$

$\Leftrightarrow$

$$(1+X_{12}^2)(1-X_{12})-(1-X_{12})(X_{13}+X_{23})^2=0,$$

by an argument similar to the one in Case 1. This equation is equivalent to

$$(1+X_{12})^2=(X_{13}+X_{23})^2$$

therefore we again have two subcases: either

$$1+X_{12}=X_{13}+X_{23} \tag{4.14}$$

or

$$1+X_{12}=-(X_{13}+X_{23}). \tag{4.15}$$

If (4.14) holds, then the triangle inequality $X_{12}-X_{13}-X_{23}\geq -1$ holds with equality. If (4.15) holds, then the triangle inequality $X_{12}+X_{13}+X_{23}\geq -1$ holds with equality. So if equation (4.11) holds, we also have one triangle inequality that holds with equality.

This completes the proof. ■

Theorem 4.4.7 summarizes the case $n=3$. It shows how the rank of Y characterizes the dimension of the face of C_3 where X lies.

Theorem 4.4.7 *Let $Y \in \mathcal{F}_3$ and $X = \operatorname{sMat}(Y_{1:6,0})$. Then $X \in C_3$ and*

- $\operatorname{rank} Y = 1$ *iff X is a vertex of C_3;*
- $\operatorname{rank} Y = 2$ *iff X lies in the relative interior of a face of C_3 of dimension 1;*
- $\operatorname{rank} Y = 3$ *iff X lies in the relative interior of a facet (face of dimension 2) of C_3;*
- $\operatorname{rank} Y = 4$ *iff X lies in the relative interior of C_3.*

Proof: Since $F_3 = C_3$, the first two statements follow by Theorem 4.3.4 and Lemmas 4.4.1 and 4.4.5. The third statement then follows from Lemma 4.4.6 and the fact that the triangle inequalities define all the facets of C_3. Finally, if no triangle inequality holds with equality, then this last fact also implies that X must lie in the interior of the cut polytope. ■

The following corollary is immediate.

Corollary 4.4.8 *Let $Y^* \in \mathcal{F}_3$ be optimal for SDP3. Then $\nu_3^* = \mu^*$, and if* $\operatorname{rank} Y = 3$, *then*

$$\nu_3^* < \nu_1^*.$$

Proof: The equality follows from $F_3 = C_3$, and the strict inequality from the fact that if $\operatorname{rank} Y = 3$, then $X = \operatorname{sMat}(Y_{1:6,0}) \in \operatorname{relint} \mathcal{E}_3$ and hence cannot be optimal for SDP1. ■

This concludes our study of the case $n = 3$. In the next chapter, we show that the characterization of the vertices of the cut polytope via $\operatorname{rank} Y = 1$ extends to all $n \geq 4$. More interestingly, we show that the characterization of the one-dimensional faces via $\operatorname{rank} Y = 2$ also holds for $n \geq 4$. Furthermore, we will prove that if $\operatorname{rank} Y = 2$ for $n \geq 3$, then it is possible to extract the two rank-one matrices (corresponding to cuts) which are the vertices of the one-dimensional face of the cut polytope where X lies. Indeed, we will present an algorithm that does precisely this.

Chapter 5

Ranks and Low Dimensional Faces of the Cut Polytope

Theorems 4.2.1 and 2.1.1 show that SDP1 characterizes the vertices of C_n as exactly those matrices in the elliptope that have rank equal to one. This implies that if X^* is optimal for SDP1 and $\operatorname{rank} X^* = 1$ then $\nu_1^* = \mu^*$, i.e. SDP1 yields the optimal value of Max-Cut. Furthermore, since X^* is rank-one, SDP1 also yields an optimal cut.

In this chapter we show that this characterization extends to SDP3, i.e. $Y \in \mathcal{F}_n$ is rank-one precisely when $X = \operatorname{sMat}(Y_{1:t(n),0})$ is a vertex of the cut polytope. Thus, if Y^* is optimal for SDP3 and $\operatorname{rank} Y^* = 1$ then $X^* = \operatorname{sMat}(Y^*_{1:t(n),0})$ is also rank-one, $\nu_3^* = \mu^*$ holds, and SDP3 yields an optimal cut.

We then proceed to prove a stronger property for SDP3, namely that $Y \in \mathcal{F}_n$ is rank-two if and only if $X = \operatorname{sMat}(Y_{1:t(n),0})$ is rank-two. In fact, this matrix X has a very special structure which allows us to design a simple algorithm that extracts from X two rank-one matrices (corresponding to cuts) which are the vertices of a one-dimensional face of the cut polytope. Thus we prove that X lies in the relative interior of a face of C_n of dimension 1. Hence, if Y^* is optimal for SDP3 and $\operatorname{rank} Y^* \leq 2$ then $\nu_3^* = \mu^*$ holds, and SDP3 yields both the optimal value and an optimal cut for Max-Cut (see Corollary 5.2.10).

For simplicity of the proofs, in this chapter we continue to work with the matrix Y projected onto the minimal face. Recall that because the rank of $Y \in \mathcal{F}_n$ always equals the rank of its projection onto the minimal face, all our results hold identically for the unprojected matrix Y.

The study of the geometry of F_n for $n \geq 4$ is complicated by the fact that given $X \in F_n$, we do not immediately have a full description of the corresponding variable Y feasible for SDP3. Indeed, as we observed in Section 2.5.2, we have only partial knowledge about Y. Hence we have to address two different, but related, issues:

1. Determine the possible value(s) that the unspecified elements of Y can take, for a given $X \in F_n$;
2. Hence determine the possible rank(s) that the matrix Y can have, given the rank of X.

We begin by settling the rank-one case. There are no surprises there, as the matrix Y is unique for every rank-one X and the characterization of Theorem 4.4.7 carries through. We then proceed in Section 5.2 to the rank-two case where we prove that, surprisingly, the characterization from Theorem 4.4.7 also holds.

5.1 Rank Characterization of Vertices

The next theorem gives the complete description of the rank-one case.

Theorem 5.1.1 *For $Y \in \mathcal{F}_n$ and $X = \operatorname{sMat}(Y_{1:t(n),0})$:*

1. *If $\operatorname{rank} X = 1$ then the unspecified elements of Y are uniquely determined by*

$$Y_{T(i,j),T(k,l)} = X_{ij}\,X_{kl}$$

for all 4-tuples (i,j,k,l) of distinct elements from $\{1,\ldots,n\}$. Hence Y is uniquely determined by X.

2. $\operatorname{rank} Y = 1 \quad \Leftrightarrow \quad \operatorname{rank} X = 1 \quad \Leftrightarrow \quad X$ *is a vertex of C_n.*

Proof: If $\operatorname{rank} X = 1$ then Theorem 2.1.1 implies $|X_{ij}| = 1$ for all i, j, since X is positive semidefinite. Given any 4-tuple (i,j,k,l) of distinct elements from $\{1,\ldots,n\}$, consider the 3×3 principal submatrix of Y corresponding to rows and columns $\{0, T(i,j), T(k,l)\}$:

$$\begin{pmatrix} 1 & X_{ij} & X_{kl} \\ X_{ij} & 1 & y \\ X_{kl} & y & 1 \end{pmatrix} \succeq 0,$$

where y denotes the element $Y_{T(i,j),T(k,l)}$. By Lemma 4.4.2, $|X_{ij}| = 1$ implies that $y = X_{ij}\,X_{kl}$. This proves the first claim.

The implication $\operatorname{rank} Y = 1 \Rightarrow \operatorname{rank} X = 1$ is clear, and the implication

$$X \text{ is a vertex of } C_n \Rightarrow \operatorname{rank} X = 1$$

follows by Lemma 4.1.1.

If $\operatorname{rank} X = 1$ then $\operatorname{diag} X = e$ implies $X_{ij} = v_i\, v_j$ for some $v \in \{\pm 1\}^n$ and thus, by Lemma 4.1.1, X is a vertex of C_n. Furthermore, let $x = \operatorname{svec}(v\, v^T) = \operatorname{svec}(X)$, so that $x_{T(i,j)} = v_i\, v_j$, and consider the matrix

$$\hat{Y} = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T = \begin{pmatrix} 1 & x^T \\ x & x\, x^T \end{pmatrix}.$$

It is clear that $\operatorname{sMat}(\hat{Y}_{1:t(n),0})$ equals X. Furthermore,

$$\hat{Y}_{T(i,k),T(k,j)} = x_{T(i,k)}\, x_{T(k,j)} = v_i\, v_k\, v_k\, v_j = X_{ij} = Y_{T(i,k),T(k,j)}$$

and

$$\hat{Y}_{T(i,j),T(k,l)} = x_{T(i,j)}\, x_{T(k,l)} = v_i\, v_j\, v_k\, v_l = X_{ij}\, X_{kl} = Y_{T(i,j),T(k,l)}.$$

Hence, $\hat{Y} = Y$ and so $\operatorname{rank} X = 1 \Rightarrow \operatorname{rank} Y = 1$. This concludes the proof. ■

This settles the rank-one case. We proceed to the case where $\operatorname{rank} X = 2$.

5.2 Rank Characterization of One-Dimensional Faces

5.2.1 Implications of Rank-Two for $X \in F_n$

In this section we prove that if $X \in F_n$ and $\operatorname{rank} X = 2$, then X has a very particular structure. We first prove Lemma 5.2.1 which is a straightforward generalization of previous results.

Lemma 5.2.1 *If $X \in F_n$ and $\operatorname{rank} X = 2$, then for every triple (i, j, k) of distinct indices from $\{1, \ldots, n\}$, at least one of X_{ij}, X_{ik}, X_{jk} has absolute value equal to 1.*

Proof: Given i, j, k, consider the 3×3 principal submatrix of X indexed by rows and columns $\{0, T(i, j), T(i, k)\}$:

$$X_{i,j,k} = \begin{pmatrix} 1 & X_{ij} & X_{ik} \\ X_{ij} & 1 & X_{jk} \\ X_{ik} & X_{jk} & 1 \end{pmatrix}.$$

Since $\operatorname{rank} X = 2$, $\operatorname{rank} X_{i,j,k} \leq 2$, and clearly $X_{i,j,k} \succeq 0$.

We have two cases. If $\operatorname{rank} X_{i,j,k} = 1$, then Theorem 2.1.1 implies $|X_{ij}|$, $|X_{ik}|$, and $|X_{jk}|$ all equal 1, and we are done. If $\operatorname{rank} X_{i,j,k} = 2$, then considering the 4×4

submatrix of Y

$$\begin{pmatrix} 1 & X_{ij} & X_{ik} & X_{jk} \\ X_{ij} & 1 & X_{jk} & X_{ik} \\ X_{ik} & X_{jk} & 1 & X_{ij} \\ X_{jk} & X_{ik} & X_{ij} & 1 \end{pmatrix}$$

and applying Lemma 4.4.4, we conclude that one of X_{ij}, X_{ik}, X_{jk} has absolute value equal to 1. ■

We now apply Lemma 5.2.1 to prove a somewhat surprising result which is essential for proving the results in this chapter about characterizing the one-dimensional faces of C_n.

Lemma 5.2.2 *Suppose $X \in F_n$, $n \geq 4$ and $\operatorname{rank} X = 2$. Then there exists a scalar $\alpha \in [0, 1)$ such that for all off-diagonal entries X_{ij} of X, either*

$$|X_{ij}| = 1 \quad \text{or} \quad |X_{ij}| = \alpha.$$

Furthermore, there is at least one occurrence of $\pm\alpha$ in the first row of X, i.e. there exists $j \in \{2, \ldots, n\}$ such that $|X_{1j}| = \alpha$.

Proof: The proof is by contradiction. Suppose there exist $\alpha \in (-1, 1)$ and $\beta \in (-1, 1)$ such that $|\alpha| \neq |\beta|$ and $X_{ij} = \alpha$, $X_{kl} = \beta$ for indices i, j, k, l from $\{1, \ldots, n\}$. We have two cases.

Case 1: $\{i,j\} \cap \{k,l\} \neq \emptyset$: In this case, suppose $j = l$ without loss of generality, so $X_{ij} = \alpha$ and $X_{jk} = \beta$. Consider the 3×3 principal submatrix of X:

$$X_{i,j,k} = \begin{pmatrix} 1 & X_{ij} & X_{ik} \\ X_{ij} & 1 & X_{jk} \\ X_{ik} & X_{jk} & 1 \end{pmatrix} \succeq 0.$$

Clearly $\operatorname{rank} X_{i,j,k} \leq 2$ and since $|X_{ij}| \neq 1$ and $|X_{jk}| \neq 1$, by Theorem 2.1.1 $\operatorname{rank} X_{i,j,k} \neq 1$. Hence $\operatorname{rank} X_{i,j,k} = 2$ and by Lemma 5.2.1, $|X_{ik}| = 1$ holds. Now, Lemma 4.4.2 implies $X_{ij} = X_{ik}\, X_{jk}$, so $\alpha = \pm\beta$ and $|\alpha| = |\beta|$, which is a contradiction. This completes Case 1.

Case 2: $\{i,j\} \cap \{k,l\} = \emptyset$: As in the first case, we consider the principal submatrix $X_{i,j,k}$ of X which, as in Case 1, has rank equal to 2 (since $|X_{ij}| \neq 1$). Therefore (by Lemma 5.2.1), either $|X_{ik}| = 1$ or $|X_{jk}| = 1$.

If $|X_{ik}| = 1$ then by Lemma 4.4.2, $X_{ij} = X_{ik}\, X_{jk}$, so $|X_{jk}| = |X_{ij}| = |\alpha|$. Now consider another 3×3 principal submatrix of X:

$$X_{j,k,l} = \begin{pmatrix} 1 & X_{jk} & X_{jl} \\ X_{jk} & 1 & X_{kl} \\ X_{jl} & X_{kl} & 1 \end{pmatrix} \succeq 0.$$

Since $|X_{kl}| \neq 1$, $\operatorname{rank} X_{j,k,l} = 2$. Hence one of its other two entries has absolute value equal to one. But $|X_{jk}| = |\alpha| \neq 1$, hence $|X_{jl}| = 1$. By Lemma 4.4.2, $X_{jk} = X_{jl}\, X_{kl}$ and therefore $|X_{jk}| = |X_{kl}|$, or equivalently, $|\alpha| = |\beta|$, which contradicts our assumption that $|\alpha| \neq |\beta|$.

Alternatively, if $|X_{jk}| = 1$ then since the principal submatrix $X_{i,j,k}$ has rank equal to 2, it follows that $X_{ij} = X_{jk}\,X_{ik}$, i.e. $|X_{ij}| = |X_{ik}| = |\alpha|$. Now consider the 3×3 principal submatrix of X:

$$X_{i,k,l} = \begin{pmatrix} 1 & X_{ik} & X_{il} \\ X_{ik} & 1 & X_{kl} \\ X_{il} & X_{kl} & 1 \end{pmatrix} \succeq 0.$$

Since $|X_{ik}| \neq 1$ and $|X_{kl}| = |\beta| \neq 1$, $|X_{il}| = 1$ must hold. But then Lemma 4.4.2 implies $|X_{ik}| = |X_{kl}|$, or equivalently, $|\alpha| = |\beta|$, again a contradiction. This completes Case 2, and the proof of the first claim.

Finally, suppose that $|X_{1j}| = 1$ for all $j = 2, \ldots, n$. Then the 3×3 principal submatrix of X:

$$X_{1,i,j} = \begin{pmatrix} 1 & X_{1i} & X_{1j} \\ X_{1i} & 1 & X_{ij} \\ X_{1j} & X_{ij} & 1 \end{pmatrix} \succeq 0$$

implies, by Lemma 4.4.2, that $|X_{ij}| = 1$ for all i, j. But then, since $X \succeq 0$, Theorem 2.1.1 implies that $\operatorname{rank} X = 1$, a contradiction. Hence, we conclude that there exists $j \in \{2, \ldots, n\}$ such that $|X_{1j}| = \alpha$. ■

5.2.2 Algorithm for Constructing the Vertex Cuts

In this section we present an algorithm which, given $X \in F_n$ satisfying $\operatorname{rank} X = 2$, explicitly constructs two cut vectors v_1 and v_2 in $\{\pm 1\}^n$ and non-negative scalars α_1, α_2 such that

$$X = \alpha_1 \, v_1 \, v_1^T \; + \; \alpha_2 \, v_2 \, v_2^T, \quad \alpha_1 + \alpha_2 = 1.$$

By Lemma 4.1.1, this immediately implies that X is in the relative interior of a one-dimensional face of C_n.

We note that the case $\alpha = 0$, where α is the scalar defined in the statement of Lemma 5.2.2, will require special attention in the algorithm.

Algorithm 5.2.3 *(Construction of Vertices for $X \in F_n$, $\operatorname{rank} X = 2$)*

```
Input: Matrix $X \in F_n$ satisfying $\operatorname{rank} X = 2$, and scalar $\alpha \in [0,1)$ (as defined in
    the statement of Lemma 5.2.2).

If $\alpha \neq 0$ then

        1. Let $\alpha_1 = \frac{1+\alpha}{2}$, $\alpha_2 = \frac{1-\alpha}{2}$, $(v_1)_1 = 1$ and $(v_2)_1 = 1$.

        2. For $j = 2, 3, \ldots, n$

            if $X_{1j} = 1$ then let $(v_1)_j = 1$ and $(v_2)_j = 1$;

            if $X_{1j} = -1$ then let $(v_1)_j = -1$ and $(v_2)_j = -1$;

            if $X_{1j} = \alpha$ then let $(v_1)_j = 1$ and $(v_2)_j = -1$;

            if $X_{1j} = -\alpha$ then let $(v_1)_j = -1$ and $(v_2)_j = 1$.
```

```
If $\alpha = 0$ then by Lemma 5.2.2, and by symmetrically permuting rows and columns
    if necessary, we may assume without loss of generality that $X_{12} = \alpha = 0$. Note
    that, with this assumption, if $X_{1j} = 0$ then $|X_{2j}| = 1$ by Lemma 5.2.1.

    1. Let $\alpha_1 = \frac{1}{2}$, $\alpha_2 = \frac{1}{2}$, $(v_1)_1 = 1$, $(v_2)_1 = 1$, $(v_1)_2 = 1$ and $(v_2)_2 = -1$.

    2. For $j = 3, 4, \dots, n$

       if $X_{1j} = 1$ then let $(v_1)_j = 1$ and $(v_2)_j = 1$;

       if $X_{1j} = -1$ then let $(v_1)_j = -1$ and $(v_2)_j = -1$;

       if $X_{1j} = 0$ then as observed above, we have two subcases:

           if $X_{2j} = 1$ then let $(v_1)_j = 1$ and $(v_2)_j = -1$;

           if $X_{2j} = -1$ then let $(v_1)_j = -1$ and $(v_2)_j = 1$.

Output: $v_1$, $v_2$, $\alpha_1$, and $\alpha_2$.
```

We demonstrate the application of Algorithm 5.2.3 in the following two examples.

Example 5.2.4 *($\alpha \neq 0$) Consider*

$$X = \begin{pmatrix} 1 & \frac{1}{3} & 1 & \frac{1}{3} \\ \frac{1}{3} & 1 & \frac{1}{3} & 1 \\ 1 & \frac{1}{3} & 1 & \frac{1}{3} \\ \frac{1}{3} & 1 & \frac{1}{3} & 1 \end{pmatrix}.$$

It is straightforward to check that $\operatorname{rank} X = 2$*, and* $X \in F_4$ *since*

$$Y = \begin{pmatrix} 1 & 1 & \frac{1}{3} & 1 & 1 & \frac{1}{3} & 1 & \frac{1}{3} & 1 & \frac{1}{3} & 1 \\ 1 & 1 & \frac{1}{3} & 1 & 1 & \frac{1}{3} & 1 & \frac{1}{3} & 1 & \frac{1}{3} & 1 \\ \frac{1}{3} & \frac{1}{3} & 1 & \frac{1}{3} & \frac{1}{3} & 1 & \frac{1}{3} & 1 & \frac{1}{3} & 1 & \frac{1}{3} \\ 1 & 1 & \frac{1}{3} & 1 & 1 & \frac{1}{3} & 1 & \frac{1}{3} & 1 & \frac{1}{3} & 1 \\ 1 & 1 & \frac{1}{3} & 1 & 1 & \frac{1}{3} & 1 & \frac{1}{3} & 1 & \frac{1}{3} & 1 \\ \frac{1}{3} & \frac{1}{3} & 1 & \frac{1}{3} & \frac{1}{3} & 1 & \frac{1}{3} & 1 & \frac{1}{3} & 1 & \frac{1}{3} \\ 1 & 1 & \frac{1}{3} & 1 & 1 & \frac{1}{3} & 1 & \frac{1}{3} & 1 & \frac{1}{3} & 1 \\ \frac{1}{3} & \frac{1}{3} & 1 & \frac{1}{3} & \frac{1}{3} & 1 & \frac{1}{3} & 1 & \frac{1}{3} & 1 & \frac{1}{3} \\ 1 & 1 & \frac{1}{3} & 1 & 1 & \frac{1}{3} & 1 & \frac{1}{3} & 1 & \frac{1}{3} & 1 \\ \frac{1}{3} & \frac{1}{3} & 1 & \frac{1}{3} & \frac{1}{3} & 1 & \frac{1}{3} & 1 & \frac{1}{3} & 1 & \frac{1}{3} \\ 1 & 1 & \frac{1}{3} & 1 & 1 & \frac{1}{3} & 1 & \frac{1}{3} & 1 & \frac{1}{3} & 1 \end{pmatrix} \in \mathcal{F}_4$$

and $\operatorname{sMat}(Y_{1:10,0}) = X$*. Note that Lemma 5.2.2 holds with* $\alpha = \frac{1}{3}$*.*

Let us apply Algorithm 5.2.3:

- $\alpha_1 = \frac{1+\alpha}{2} = \frac{2}{3}$, $\alpha_2 = \frac{1-\alpha}{2} = \frac{1}{3}$, $(v_1)_1 = 1$ *and* $(v_2)_1 = 1$*;*

- $X_{12} = \frac{1}{3} = \alpha$*, so* $(v_1)_2 = 1$ *and* $(v_2)_2 = -1$*;*

- $X_{13} = 1$*, so* $(v_1)_3 = 1$ *and* $(v_2)_3 = 1$*;*

- $X_{14} = \frac{1}{3} = \alpha$*, so* $(v_1)_4 = 1$ *and* $(v_2)_4 = -1$*.*

Hence the algorithm yields $\alpha_1 = \frac{2}{3}$, $\alpha_2 = \frac{1}{3}$, $v_1 = (1,1,1,1)^T$ *and* $v_2 = (1,-1,1,-1)^T$*. It is easy to verify that* $\alpha_1 \, v_1 \, v_1^T \; + \; \alpha_2 \, v_2 \, v_2^T = X$*.*

Example 5.2.5 *($\alpha = 0$) Consider*

$$X = \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}.$$

Again $\operatorname{rank} X = 2$*, and* $X \in F_4$ *because*

$$Y = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & -1 & 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & -1 & 1 & -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & -1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & -1 & 1 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 1 & -1 & 0 \\ -1 & -1 & 0 & -1 & 0 & 1 & -1 & 1 & 0 & 0 & -1 \\ 1 & 1 & 0 & 1 & 0 & -1 & 1 & -1 & 0 & 0 & 1 \\ -1 & -1 & 0 & -1 & 0 & 1 & -1 & 1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & -1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & -1 & 1 & -1 & 0 & 0 & 1 \end{pmatrix} \in \mathcal{F}_4$$

and $\operatorname{sMat}(Y_{1:10,0}) = X$*.*

Note $X_{12} = 0$*, so* $X_{12} = \alpha$ *holds. Applying Algorithm 5.2.3:*

- $\alpha_1 = \frac{1}{2}$, $\alpha_2 = \frac{1}{2}$, $(v_1)_1 = 1$, $(v_2)_1 = 1$, $(v_1)_2 = 1$, *and* $(v_2)_2 = -1$*;*

- $X_{13} = 0$*, so we check* X_{23} *(which must equal* ± 1*):* $X_{23} = -1$*, so* $(v_1)_3 = -1$

and $(v_2)_3 = 1$*;*

- $X_{14} = -1$*, so* $(v_1)_4 = -1$ *and* $(v_2)_4 = -1$*.*

Hence the algorithm yields $\alpha_1 = \frac{1}{2}$*,* $\alpha_2 = \frac{1}{2}$*,* $v_1 = (1, 1, -1, -1)^T$ *and* $v_2 = (1, -1, 1, -1)^T$*. It is again easy to verify that* $\alpha_1 \, v_1 \, v_1^T \; + \; \alpha_2 \, v_2 \, v_2^T = X$*.*

We now prove the correctness of Algorithm 5.2.3. It is straightforward to check that $\alpha_1 \geq 0$, $\alpha_2 \geq 0$, $\alpha_1 + \alpha_2 = 1$, and $\alpha_1 - \alpha_2 = \alpha$. If we define

$$X^{\mathrm{alg}} := \alpha_1 \, v_1 \, v_1^T \; + \; \alpha_2 \, v_2 \, v_2^T ,$$

we have the following result:

Lemma 5.2.6 *Let* $X \in F_n$*,* $\operatorname{rank} X = 2$*, and suppose that* v_1*,* v_2*,* α_1*, and* α_2 *are computed by Algorithm 5.2.3. Then* $X = X^{\mathrm{alg}}$*.*

Proof: The proof has two cases which parallel the two cases in Algorithm 5.2.3. However, first observe that for both cases,

$$X^{\mathrm{alg}}_{ii} = \alpha_1 \, (v_1)_i^2 + \alpha_2 \, (v_2)_i^2 = \alpha_1 \cdot 1 + \alpha_2 \cdot 1 = 1,$$

therefore $\operatorname{diag}(X^{\mathrm{alg}}) = e = \operatorname{diag}(X)$ holds for both cases. Now consider each case in turn.

Case 1: $\alpha \neq 0$**:** First check that the first rows of X and X^{alg} are equal.

For $j = 2, 3, \ldots, n$

$$\begin{aligned} X^{\text{alg}}_{1j} &= \alpha_1\,(v_1)_1\,(v_1)_j + \alpha_2\,(v_2)_1\,(v_2)_j \\ &= \begin{cases} \alpha_1 \cdot (1) + \alpha_2 \cdot (1) = 1, & \text{if } X_{1j} = 1 \\ \alpha_1 \cdot (-1) + \alpha_2 \cdot (-1) = -1, & \text{if } X_{1j} = -1 \\ \alpha_1 \cdot (1) + \alpha_2 \cdot (-1) = \alpha_1 - \alpha_2 = \alpha, & \text{if } X_{1j} = \alpha \\ \alpha_1 \cdot (-1) + \alpha_2 \cdot (1) = \alpha_2 - \alpha_1 = -\alpha, & \text{if } X_{1j} = -\alpha \end{cases} \end{aligned}$$

therefore their first rows are equal.

Now for each pair of indices i, j from $\{2, \ldots, n\}$, $i \neq j$, Lemma 5.2.1 implies that at least one of X_{1i}, X_{1j}, X_{ij} has absolute value equal to 1. We consider three cases:

If $|X_{1i}| = 1$ **then** by Lemma 4.4.2, $X_{ij} = X_{1i}\,X_{1j}$. Let $X_{1i} = \sigma, \sigma^2 = 1$, hence the algorithm sets $(v_1)_i = \sigma$ and $(v_2)_i = \sigma$. We now consider two subcases:

If $X_{1j} = \delta, \delta^2 = 1$ **then** the algorithm sets $(v_1)_j = \delta$ and $(v_2)_j = \delta$ and therefore

$$\begin{aligned} X^{\text{alg}}_{ij} &= \alpha_1\,\sigma\,\delta + \alpha_2\,\sigma\,\delta \\ &= \sigma\,\delta \\ &= X_{1i}\,X_{1j} \\ &= X_{ij}. \end{aligned}$$

If $X_{1j} = \delta\,\alpha, \delta^2 = 1$ **then** the algorithm sets $(v_1)_j = \delta$ and $(v_2)_j = -\delta$

and therefore

$$\begin{aligned}X_{ij}^{\text{alg}} &= \alpha_1\,\sigma\,\delta + \alpha_2\,\sigma\,(-\delta)\\ &= \delta\,(\alpha_1-\alpha_2)\,\sigma\\ &= (\delta\,\alpha)\,\sigma\\ &= X_{1j}\,X_{1i}\\ &= X_{ij}.\end{aligned}$$

Hence, if $|X_{1i}| = 1$, we are done.

If $|X_{1j}| = 1$ **then** again by Lemma 4.4.2, $X_{ij} = X_{1i}\,X_{1j}$. Applying the same argument as for the previous case, we are done.

Otherwise, we have $X_{1i} = \sigma\,\alpha, \sigma^2 = 1$, $X_{1j} = \delta\,\alpha, \delta^2 = 1$, and $|X_{ij}| = 1$. In this case, the algorithm sets

$$(v_1)_i = \sigma, (v_2)_i = -\sigma, (v_1)_j = \delta, \text{ and } (v_2)_j = -\delta.$$

Furthermore, Lemma 4.4.2 implies $X_{1i} = X_{ij}\,X_{1j}$, which can be rewritten as $\sigma\,\alpha = X_{ij}\,\delta\,\alpha$, and since $\alpha \neq 0$, $X_{ij} = \sigma\,\delta$ (because $\delta^2 = 1$). Now,

$$\begin{aligned}X_{ij}^{\text{alg}} &= \alpha_1\,\sigma\,\delta + \alpha_2\,(-\sigma)\,(-\delta)\\ &= (\alpha_1+\alpha_2)\,\sigma\,\delta\\ &= \sigma\,\delta\\ &= X_{ij}.\end{aligned}$$

Therefore, if $\alpha \neq 0$, $X^{\text{alg}} = X$.

Case 2: $\alpha = 0$: As in the description of Algorithm 5.2.3, we suppose without loss of generality that $X_{12} = 0$. Now,

$$X_{12}^{\text{alg}} = \frac{1}{2}\,(v_1)_1\,(v_1)_2 + \frac{1}{2}\,(v_2)_1\,(v_2)_2 = \frac{1}{2} - \frac{1}{2} = 0 = X_{12},$$

as desired.

Next we consider the remaining non-diagonal entries of the first two rows of X^{alg}. For each $j = 3, 4, \ldots, n$, there are two cases:

either $|X_{1j}| = 1$**,** so that $X_{1j} = \sigma, \sigma^2 = 1$ and the algorithm sets $(v_1)_j = \sigma$ and $(v_2)_j = \sigma$. Then Lemma 4.4.2 implies $X_{2j} = X_{1j}\,X_{12}$, and so $X_{2j} = 0$. Now,

$$\begin{array}{rcl} X_{1j}^{\text{alg}} & = & \frac{1}{2}\,(v_1)_j + \frac{1}{2}\,(v_2)_j \\ & = & \sigma \quad = \quad X_{1j}, \end{array}$$

and

$$\begin{array}{rcl} X_{2j}^{\text{alg}} & = & \frac{1}{2}\,(v_1)_2\,(v_1)_j + \frac{1}{2}\,(v_2)_2\,(v_2)_j \\ & = & \frac{1}{2}\,\sigma - \frac{1}{2}\,\sigma \\ & = & 0 \quad = \quad X_{2j}. \end{array}$$

or $X_{1j} = 0$**,** in which case Lemma 5.2.1 implies $|X_{2j}| = 1$ so $X_{2j} = \sigma, \sigma^2 = 1$, and the algorithm sets $(v_1)_j = \sigma$, $(v_2)_j = -\sigma$. Thus,

$$\begin{array}{rcl} X_{1j}^{\text{alg}} & = & \frac{1}{2}\,(v_1)_j + \frac{1}{2}\,(v_2)_j \\ & = & \frac{1}{2}\,\sigma + \frac{1}{2}\,(-\sigma) \\ & = & 0 \quad = \quad X_{1j} \end{array}$$

and

$$\begin{aligned} X^{\text{alg}}_{2j} = &= \tfrac{1}{2}\,(v_1)_2\,(v_1)_j + \tfrac{1}{2}\,(v_2)_2\,(v_2)_j \\ &= \tfrac{1}{2}\,\sigma + \tfrac{1}{2}\,(-1)\,(-\sigma) \\ &= \sigma \quad = \quad X_{2j} \end{aligned}$$

This proves that the first two rows of X^{alg} and X are identical.

Now consider any pair of indices i, j from $\{3, \ldots, n\}$, $i \neq j$ and the 4×4 principal submatrix of X:

$$\begin{pmatrix} 1 & X_{12} & X_{1i} & X_{1j} \\ X_{12} & 1 & X_{2i} & X_{2j} \\ X_{1i} & X_{2i} & 1 & X_{ij} \\ X_{1j} & X_{2j} & X_{ij} & 1 \end{pmatrix} \succeq 0.$$

Since $X_{12} = 0$, the top left 2×2 principal submatrix is invertible, so applying the Schur complement, the above principal submatrix is positive semidefinite if and only if

$$\begin{pmatrix} 1 & X_{ij} \\ X_{ij} & 1 \end{pmatrix} - \begin{pmatrix} X_{1i} & X_{2i} \\ X_{1j} & X_{2j} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} X_{1i} & X_{1j} \\ X_{2i} & X_{2j} \end{pmatrix} \succeq 0$$

$$\Leftrightarrow \begin{pmatrix} 1 - (X_{1i}^2 + X_{2i}^2) & X_{ij} - (X_{1i}\,X_{1j} + X_{2i}\,X_{2j}) \\ X_{ij} - (X_{1i}\,X_{1j} + X_{2i}\,X_{2j}) & 1 - (X_{1j}^2 + X_{2j}^2) \end{pmatrix} \succeq 0. \quad (5.1)$$

The fact that $X_{12} = 0$ also implies:

- by Lemma 5.2.1, that

  - one of X_{1i}, X_{2i} has absolute value equal to 1, and
  - one of X_{1j}, X_{2j} has absolute value equal to 1;

- by Lemma 4.4.2, that the other element of each of the above two pairs must equal 0.

Hence,

$$X_{1i}^2 + X_{2i}^2 = 1 \quad \text{and} \quad X_{1j}^2 + X_{2j}^2 = 1,$$

and therefore the matrix inequality (5.1) implies that

$$X_{ij} = X_{1i}\,X_{1j} + X_{2i}\,X_{2j}.$$

We can now proceed to checking that X^{alg} indeed equals the right-hand side of this equation. There are four subcases:

Case i: $|X_{1i}| = 1$ and $|X_{1j}| = 1$: Let $X_{1i} = \sigma, \sigma^2 = 1$ and $X_{1j} = \delta, \delta^2 = 1$, and we also know $X_{2i} = 0$ and $X_{2j} = 0$. The algorithm sets $(v_1)_i = \sigma$, $(v_2)_i = \sigma$, $(v_1)_j = \delta$ and $(v_2)_j = \delta$. Therefore,

$$\begin{aligned} X_{ij}^{\text{alg}} &= \tfrac{1}{2}\,(v_1)_i\,(v_1)_j + \tfrac{1}{2}\,(v_2)_i\,(v_2)_j \\ &= \sigma\,\delta \\ &= X_{1i}\,X_{1j} \\ &= X_{1i}\,X_{1j} + X_{2i}\,X_{2j}, \quad \text{since } X_{2i}\,X_{2j} = 0 \\ &= X_{ij}. \end{aligned}$$

Case ii: $|X_{1i}| = 1$ and $|X_{2j}| = 1$: Let $X_{1i} = \sigma, \sigma^2 = 1$ and $X_{2j} = \delta, \delta^2 = 1$, and also $X_{2i} = 0$ and $X_{1j} = 0$. The algorithm sets $(v_1)_i = \sigma$, $(v_2)_i = \sigma$, $(v_1)_j = \delta$ and $(v_2)_j = -\delta$. Therefore,

$$\begin{aligned} X_{ij}^{\mathrm{alg}} &= \tfrac{1}{2}\,(v_1)_i\,(v_1)_j + \tfrac{1}{2}\,(v_2)_i\,(v_2)_j \\ &= \tfrac{1}{2}\,\sigma\,\delta + \tfrac{1}{2}\,\sigma\,(-\delta) \\ &= 0 \\ &= X_{1i}\,X_{1j} + X_{2i}\,X_{2j}, \quad \text{since } X_{1j} = 0 \text{ and } X_{2i} = 0 \\ &= X_{ij}. \end{aligned}$$

Case iii: $|X_{2i}| = 1$ and $|X_{1j}| = 1$: Let $X_{2i} = \sigma, \sigma^2 = 1$ and $X_{1j} = \delta, \delta^2 = 1$, and also $X_{2j} = 0$ and $X_{1i} = 0$. The algorithm sets $(v_1)_i = \sigma$, $(v_2)_i = -\sigma$, $(v_1)_j = \delta$ and $(v_2)_j = \delta$, and it is straightforward to check that $X_{ij}^{\mathrm{alg}} = X_{ij}$ by applying the same kind of argument as in the previous cases.

Case iv: $|X_{2i}| = 1$ and $|X_{2j}| = 1$: Let $X_{2i} = \sigma, \sigma^2 = 1$ and $X_{2j} = \delta, \delta^2 = 1$, and also $X_{1i} = 0$ and $X_{1j} = 0$. The algorithm sets $(v_1)_i = \sigma$, $(v_2)_i = -\sigma$, $(v_1)_j = \delta$ and $(v_2)_j = -\delta$, and again it is straightforward to check that $X_{ij}^{\mathrm{alg}} = X_{ij}$.

Therefore, if $\alpha = 0$, $X^{\mathrm{alg}} = X$ also holds. This completes the proof. ■

Corollary 5.2.7 *For $n \geq 3$,*

$$X \in F_n \text{ and } \operatorname{rank} X = 2 \qquad \text{if and only if}$$

$$X \text{ is in the relative interior of a face of } C_n \text{ of dimension 1.}$$

Proof: If $X \in F_n$ and $\operatorname{rank} X = 2$, then Lemma 5.2.6 and Lemma 4.1.1 together imply that X is in the relative interior of a face of dimension 1.

Conversely, suppose that X is in the relative interior of a face of dimension 1. Then by Lemma 4.1.1 again, there exist vectors v_1 and v_2 in $\{\pm 1\}^n$, $v_1 \neq v_2$, and a scalar $\alpha \in (0,1)$ such that $X = \alpha\, v_1\, v_1^T + (1-\alpha)\, v_2\, v_2^T$.

By convexity, $X \in F_n$, and clearly $\operatorname{rank} X \leq 2$. To complete the proof, we suppose $\operatorname{rank} X = 1$ and obtain a contradiction.

Clearly $X \succeq 0$, therefore Theorem 2.1.1 implies $|X_{ij}| = 1$ for all i, j. We may also assume without loss of generality that $(v_1)_1 = 1$ and $(v_2)_1 = 1$. Since $v_1 \neq v_2$, there is an index $j \in \{2, \ldots, n\}$ such that $(v_1)_j = -(v_2)_j$. For such a choice of j, we have

$$\begin{aligned} X_{1j} &= \alpha\,(v_1)_j + (1-\alpha)\,(v_2)_j \\ &= \alpha\,(v_1)_j + (1-\alpha)\,(-(v_1)_j) \\ &= (2\,\alpha - 1)\,(v_1)_j. \end{aligned}$$

But since $\alpha \in (0,1)$, it follows that $|2\,\alpha - 1| < 1$, and therefore

$$|X_{1j}| = |(2\,\alpha - 1)\,(v_1)_j| = |2\,\alpha - 1| < 1,$$

a contradiction. ∎

5.2.3 Uniqueness of Y and Rank Characterization

This section proves the last implication required to obtain Theorem 5.2.9. We show that if $X \in F_n$, i.e. if there exists $Y \in \mathcal{F}_n$ such that $X = \operatorname{sMat}(Y_{1:t(n),0})$, and $\operatorname{rank} X = 2$, then $\operatorname{rank} Y = 2$ holds.

Suppose $X \in F_n$ and $\operatorname{rank} X = 2$. Applying Algorithm 5.2.3, we find v_1, v_2, α_1, and α_2 such that $X = \alpha_1 v_1 v_1^T + \alpha_2 v_2 v_2^T$. Define $x_1 := \operatorname{svec}(v_1 v_1^T)$ and $x_2 := \operatorname{svec}(v_2 v_2^T)$, and consider

$$\begin{aligned} Y &:= \alpha_1 \begin{pmatrix} 1 \\ x_1 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \end{pmatrix}^T + \alpha_2 \begin{pmatrix} 1 \\ x_2 \end{pmatrix} \begin{pmatrix} 1 \\ x_2 \end{pmatrix}^T \\ &= \begin{pmatrix} \alpha_1 + \alpha_2 & (\alpha_1 x_1 + \alpha_2 x_2)^T \\ \alpha_1 x_1 + \alpha_2 x_2 & \alpha_1 x_1 x_1^T + \alpha_2 x_2 x_2^T \end{pmatrix}. \end{aligned}$$

Clearly $\operatorname{rank} Y \leq 2$. Also, $Y \in \mathcal{F}_n$ (by convexity) and furthermore

$$\operatorname{sMat}(Y_{1:t(n),0}) = \operatorname{sMat}(\alpha_1 x_1 + \alpha_2 x_2) = X.$$

Hence, $\operatorname{rank} Y \neq 1$ by Theorem 5.1.1 (since $\operatorname{rank} X = 2$). Thus $\operatorname{rank} Y = 2$ and it suffices to prove that this is the only $Y \in \mathcal{F}_n$ such that $X = \operatorname{sMat}(Y_{1:t(n),0})$. This is the result of the next lemma.

Lemma 5.2.8 *Suppose $X \in F_n$ and $\operatorname{rank} X = 2$. Then the matrix $Y \in \mathcal{F}_n$ such that $X = \operatorname{sMat}(Y_{1:t(n),0})$ is unique. Specifically, let α be the scalar defined in the statement of Lemma 5.2.2; then for each 4-tuple (i, j, k, l) of distinct elements from*

$\{1,\ldots,n\}$, $Y_{T(i,j),T(k,l)}$ *equals exactly one of the values* $\{1,-1,\alpha,-\alpha\}$.

Proof: Given X, the restriction that $Y \in \mathcal{F}_n$ fixes all the elements of Y except those of the form $Y_{T(i,j),T(k,l)}$, with i,j,k,l all distinct. We prove that all these elements are also uniquely determined in terms of the elements of X. This proves the uniqueness of Y.

We consider two cases:

Case 1: at least one of X_{ij},X_{kl} equals ± 1: If so, then the positive semidefiniteness of the 3×3 principal submatrix of Y corresponding to rows and columns $\{0, T(i,j), T(k,l)\}$:

$$\begin{pmatrix} 1 & X_{ij} & X_{kl} \\ X_{ij} & 1 & y \\ X_{kl} & y & 1 \end{pmatrix} \succeq 0,$$

where y denotes $Y_{T(i,j),T(k,l)}$, implies (by Lemma 4.4.2) that $y = X_{ij}\,X_{kl} \in \{\pm 1, \pm\alpha\}$, since at least one of X_{ij},X_{kl} equals ± 1.

Case 2: both X_{ij},X_{kl} equal $\pm\alpha$: In this case, we write $X_{ij} = \sigma\,\alpha, \sigma^2 = 1$, $X_{kl} = \delta\,\alpha, \delta^2 = 1$, and consider the 3×3 principal submatrix of X:

$$X_{i,j,k} = \begin{pmatrix} 1 & X_{ij} & X_{ik} \\ X_{ij} & 1 & X_{jk} \\ X_{ik} & X_{jk} & 1 \end{pmatrix} \succeq 0.$$

By Lemma 5.2.1, either $|X_{ik}| = 1$ or $|X_{jk}| = 1$. Without loss of generality (symmetrically permuting rows and columns if necessary) suppose $|X_{ik}| = 1$. Then let $X_{ik} = \beta, \beta^2 = 1$, and observe that by Lemma 4.4.2, $X_{jk} = X_{ik}\, X_{ij} = \beta\,\sigma\,\alpha$. Now consider the 4×4 principal submatrix of Y indexed by the rows and columns $\{0, T(i,j), T(k,l), T(j,k)\}$:

$$\begin{pmatrix} 1 & \sigma\,\alpha & \delta\,\alpha & \beta\,\sigma\,\alpha \\ \sigma\,\alpha & 1 & y & \beta \\ \delta\,\alpha & y & 1 & X_{jl} \\ \beta\,\sigma\,\alpha & \beta & X_{jl} & 1 \end{pmatrix} \succeq 0.$$

Applying the Schur complement, positive semidefiniteness of this submatrix is equivalent to

$$\begin{pmatrix} 1 & y & \beta \\ y & 1 & X_{jl} \\ \beta & X_{jl} & 1 \end{pmatrix} - \begin{pmatrix} \sigma\,\alpha \\ \delta\,\alpha \\ \beta\,\sigma\,\alpha \end{pmatrix} \begin{pmatrix} \sigma\,\alpha \\ \delta\,\alpha \\ \beta\,\sigma\,\alpha \end{pmatrix}^T \succeq 0$$

$$\Leftrightarrow \begin{pmatrix} 1-\alpha^2 & y-\sigma\,\delta\,\alpha^2 & \beta\,(1-\alpha^2) \\ y-\sigma\,\delta\,\alpha^2 & 1-\alpha^2 & X_{jl}-\beta\,\sigma\,\delta\,\alpha^2 \\ \beta\,(1-\alpha^2) & X_{jl}-\beta\,\sigma\,\delta\,\alpha^2 & 1-\alpha^2 \end{pmatrix} \succeq 0.$$

Since $|\alpha| < 1$, $1-\alpha^2 > 0$ and so, scaling both rows and columns by $\frac{1}{\sqrt{1-\alpha^2}}$ (this is a congruence and therefore it does not affect positive semidefiniteness),

$$\Leftrightarrow \begin{pmatrix} 1 & \frac{y-\sigma\,\delta\,\alpha^2}{1-\alpha^2} & \beta \\ \frac{y-\sigma\,\delta\,\alpha^2}{1-\alpha^2} & 1 & \frac{X_{jl}-\beta\,\sigma\,\delta\,\alpha^2}{1-\alpha^2} \\ \beta & \frac{X_{jl}-\beta\,\sigma\,\delta\,\alpha^2}{1-\alpha^2} & 1 \end{pmatrix} \succeq 0.$$

By Lemma 4.4.2, it follows that

$$\begin{aligned} & \tfrac{y-\sigma\,\delta\,\alpha^2}{1-\alpha^2} = \beta\,\tfrac{X_{jl}-\beta\,\sigma\,\delta\,\alpha^2}{1-\alpha^2} \\ \Rightarrow\quad & y - \sigma\,\delta\,\alpha^2 = \beta\,X_{jl} - \beta^2\,\sigma\,\delta\,\alpha^2 \\ \Rightarrow\quad & \qquad y = \beta\,X_{jl} \in \{\pm 1, \pm\alpha\}. \end{aligned}$$

This completes the proof of the uniqueness of Y. ■

Theorem 5.2.9 *For $Y \in \mathcal{F}_n$ and $X = \mathrm{sMat}\,(Y_{1:t(n),0})$,*

$\operatorname{rank} Y = 2 \quad \Leftrightarrow \quad \operatorname{rank} X = 2 \quad \Leftrightarrow \quad$ *X is in the relative interior of a face of C_n of dimension 1.*

Proof: Clearly $\operatorname{rank} Y = 2 \Rightarrow \operatorname{rank} X \leq 2$, and Theorem 5.1.1 implies $\operatorname{rank} X \geq 2$, hence $\operatorname{rank} X = 2$. We have just shown that if $\operatorname{rank} X = 2$, then Y is unique, and the construction of Y at the beginning of this section shows that $\operatorname{rank} Y = 2$. Finally, the second equivalence is exactly the statement of Corollary 5.2.7. ■

Corollary 5.2.10 *Let $Y^* \in \mathcal{F}_n$ be optimal for SDP3. If* $\operatorname{rank} Y^* \leq 2$ *then*

$$\nu_3^* = \mu^*,$$

and furthermore Y^* yields an optimal cut for Max-Cut.

Proof: Theorems 5.1.1 and 5.2.9 imply that if $\operatorname{rank} Y^* \leq 2$ then the matrix $X^* = \operatorname{sMat}(Y^*_{1:t(n),0})$ lies on the boundary of C_n, and hence $\nu_3^* = \mu^*$. If $\operatorname{rank} Y^* = 1$ then $\operatorname{rank} X^* = 1$ so it is straightforward to extract the optimal cut from X^*. Finally, if $\operatorname{rank} Y^* = 2$ then $\operatorname{rank} X^* = 2$ and Algorithm 5.2.3 yields two optimal cuts. ■

5.3 Counter-Example for the Rank-Three Case

We conclude our study of the strengthened SDP relaxations by showing that the equivalence of the ranks between Y and $X = \operatorname{sMat}(Y_{1:t(n),0})$ cannot be extended beyond rank-two.

Of course, it follows from Theorems 5.1.1 and 5.2.9 that $\operatorname{rank} Y = 3 \Rightarrow \operatorname{rank} X = 3$ for $n \geq 3$. However, the converse fails for $n = 3$, and hence for all $n \geq 3$. Indeed, an easy counterexample is provided by the 3 × 3 identity matrix since

$$Y = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

is feasible for SDP3 and has rank equal to 4, whereas $X = \mathrm{sMat}\left(Y_{1:6,0}\right) = I$ has rank equal to 3.

Chapter 6

Derivation of the AR Model for VLSI Layout

The VLSI layout (or floorplanning) problem consists in finding the optimal positions for a given set of modules of fixed area (but perhaps varying dimensions) within a facility. The objective is to minimize the distances between pairs of modules that have a nonzero connection "cost" while ensuring that no modules overlap. If the modules have varying dimensions, then finding their optimal shapes is also a part of the problem.

Our contribution is the AR (Attractor-Repeller) model which is designed to improve upon the 3-stage approach of van Camp et al.[26]. In this chapter we motivate and derive the AR model which replaces Stages 1 and 2 by a single mathematical model that finds a "good" initial point for the solver of the Stage-3 model. The design of the AR model is inspired by the work of Etawil and Vannelli [32] for the VLSI placement problem. Because the AR model is not a convex optimization problem, in Section 6.3 we isolate the source of the non-convexity and thereby derive a convex version. Numerical results demonstrating the potential of the AR model are presented in Chapter 7.

6.1 Previous Related Methods

The floorplanning problem has been throughly studied in the Operations Research literature. In order to put our contribution in context, we begin by reviewing two previous methods that relate closely to our new approach.

6.1.1 The DISCON Method

In 1980 Drezner introduced the DISCON (DISpersion-CONcentration) method [31]. To describe the DISCON method, let us suppose that the modules are labelled $1, \ldots, N$, where N is the total number of modules, and that:

1. Each module is a circle (or can be approximated by a circle) of given radius $r_i, i = 1, \ldots, N$.

2. The distance between two modules is measured as the Euclidean distance between the centres of the circles.

3. The non-negative costs c_{ij} per unit distance between modules i and j are given.

The DISCON method uses a formulation equivalent to

$$\text{(DISCON)} \quad \begin{array}{ll} \min\limits_{(x_i,y_i)} & \sum\limits_{1 \le i < j \le N} c_{ij}\, d_{ij} \\ \text{s.t.} & \\ & d_{ij} \ge r_i + r_j,\ \forall\ 1 \le i < j \le N, \end{array}$$

where

- (x_i, y_i) denotes the centre of the i^{th} module, and
- $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$.

Drezner solves the DISCON problem using a penalty algorithm and a two-phase approach:

- In Phase-1 (the dispersion phase), all the circles are dispersed far from the origin. This phase provides a starting point for the next phase.
- In Phase-2 (the concentration phase), the circles are concentrated (i.e. they are brought as close together as possible) and the resulting arrangement is the final solution.

We point out that both the objective function and the feasible set of the DISCON problem are non-convex. Furthermore, DISCON gives the user no control over the dimensions of the resulting layout of the circles (modules).

6.1.2 The NLT Method

More recently, van Camp, Carter, and Vannelli [26] have introduced a technique where all the modules as well as the facility are restricted to having fixed (given) areas and rectangular shapes, but the dimensions of the rectangles are optimized by a mathematical model.

Specifically, they introduce the following model, which we denote by vCCV:

$$
\text{(vCCV)} \quad
\begin{array}{ll}
\displaystyle\min_{(x_i,y_i)} & \displaystyle\sum_{1\le i<j\le N} c_{ij}\, d_{ij} \\
\text{s.t.} & \\
 & |x_i - x_j| - \tfrac{1}{2}(w_i + w_j) \ge 0 \quad \text{if} \quad |y_i - y_j| - \tfrac{1}{2}(h_i + h_j) < 0 \\
 & |y_i - y_j| - \tfrac{1}{2}(h_i + h_j) \ge 0 \quad \text{if} \quad |x_i - x_j| - \tfrac{1}{2}(w_i + w_j) < 0 \\
 & \tfrac{1}{2}w_T - (x_i + \tfrac{1}{2}w_i) \ge 0 \quad \text{for} \quad i = 1, \ldots, N \\
 & \tfrac{1}{2}h_T - (y_i + \tfrac{1}{2}h_i) \ge 0 \quad \text{for} \quad i = 1, \ldots, N \\
 & (x_i - \tfrac{1}{2}w_i) + \tfrac{1}{2}w_T \ge 0 \quad \text{for} \quad i = 1, \ldots, N \\
 & (y_i - \tfrac{1}{2}h_i) + \tfrac{1}{2}h_T \ge 0 \quad \text{for} \quad i = 1, \ldots, N \\
 & \min(w_i, h_i) - l_i^{\min} \ge 0 \quad \text{for} \quad i = 1, \ldots, N \\
 & l_i^{\max} - \min(w_i, h_i) \ge 0 \quad \text{for} \quad i = 1, \ldots, N \\
 & \min(w_T, h_T) - l_T^{\min} \ge 0 \\
 & l_T^{\max} - \min(w_T, h_T) \ge 0,
\end{array}
$$

where (x_i, y_i) and d_{ij} are as previously defined, and

- w_i, h_i are the width and height of module i;
- $l_i^{\min}, l_i^{\max}$ are the minimum and maximum allowable lengths for the shortest side of module i;
- w_T, h_T are the width and height of the facility; and
- $l_T^{\min}, l_T^{\max}$ are the minimum and maximum allowable lengths for the shortest

side of the facility.

Therefore, with the vCCV model, the user can input ranges for the shortest sides of the modules and of the resulting facility, and the model optimizes all the dimensions within the given ranges.

The NLT (Non-linear optimization Layout Technique) method is based on the above model and employs a three-stage approach:

1. Stage-1 aims to evenly distribute the centres of the modules inside the facility;
2. Stage-2 aims to reduce the overlap between modules;
3. Stage-3 determines the final solution.

Stage-3 consists of solving the complete vCCV model, whereas the problems solved at Stages 1 and 2 correspond to relaxations of the vCCV model. These relaxations approximate each module by a circle whose radius is proportional to the area of the module. The Stage-2 relaxation is presented in Section 6.2. We also observe that the models for all three stages are solved using a penalty-based method.

6.2 Derivation of the AR Model

6.2.1 The Stage-2 Model of the NLT Method

In order to motivate the AR model, we begin by introducing the Stage-2 model used in the NLT method. (Recall that our objective is to improve on this Stage-2 model in terms of generating an initial point for the Stage-3 vCCV problem above.) This model is a relaxation of vCCV where each module is approximated by a circle whose radius is equal to half the square root of the area of the module.

Given all the radii $r_i, i = 1, \ldots, N$, the Stage-2 model (St-2) is:

$$
\text{(St-2)} \qquad
\begin{array}{ll}
\min\limits_{(x_i,y_i),h_T,w_T} & \sum\limits_{1\le i<j\le N} c_{ij}\, d_{ij} \\
\text{subject to} & \\
& d_{ij} \ge r_i + r_j, \ \forall \ i, j \\
& \frac{1}{2} w_T \ge x_i + r_i \quad \text{for} \quad i = 1, \ldots, N \\
& \frac{1}{2} h_T \ge y_i + r_i \quad \text{for} \quad i = 1, \ldots, N \\
& \frac{1}{2} w_T \ge r_i - x_i \quad \text{for} \quad i = 1, \ldots, N \\
& \frac{1}{2} h_T \ge r_i - y_i \quad \text{for} \quad i = 1, \ldots, N \\
& l_T^{\max} \ \ge \ \min(w_T, h_T) \ \ge \ l_T^{\min}.
\end{array}
$$

The objective function is clear. As for the constraints, the first set of constraints in this model are those used in DISCON:

$$d_{ij} \ge r_i + r_j, \ \forall \ 1 \le i < j \le N. \tag{6.1}$$

These constraints prevent overlapping of any pair of circles.

Recall that in DISCON, the user has no control of the overall dimensions of the final arrangement of circles. The model St-2 addresses this issue via the next four sets of constraints, which require that all circles remain inside the facility, and the last two constraints which ensure that the facility's shortest dimension is within the given bounds.

6.2.2 Target Distance Concept

A significant difficulty with the model St-2 is that the constraints (6.1) make the feasible set of St-2 non-convex. In our new model, we enforce these constraints by drawing on the target distance paradigm introduced by Etawil and Vannelli [32].

Note that if the constraints (6.1) are removed from St-2, then the (global) optimal solution is simply to place all the circles with their centres at the same point (since we assume all the costs c_{ij} are non-negative). One way to understand why this happens is to interpret the objective function

$$\sum_{1 \le i < j \le N} c_{ij} \, d_{ij}$$

as an *attractor*, i.e. a function that seeks to make the values d_{ij} as small as possible by attracting all pairs of circles i, j to each other. To prevent this from happening, we will add to the objective function a *repeller* term that seeks to enforce the constraints (6.1).

For convenience we work with the *squares* of the distances between each pair of circles. It is clear that this does not change the optimal layout. Let us therefore define the variables

$$D_{ij} := {d_{ij}}^2 = (x_i - x_j)^2 + (y_i - y_j)^2$$

and the parameters

$$t_{ij} := \alpha\,(r_i + r_j)^2,\ \forall\ 1 \leq i < j \leq N,\ \text{ where }\ \alpha > 0.$$

The parameters t_{ij} are key to our strategy for enforcing the constraints (6.1), i.e. the separation of the circles representing the modules. The idea is that

the value $\sqrt{t_{ij}}$ is the *target distance* between the pair of circles i, j.

Equivalently, t_{ij} is the target value for D_{ij}, which is the square of the distance between the circles i and j with radii r_i and r_j respectively.

The parameter $\alpha > 0$ is introduced to provide some flexibility as to how tightly the user wishes to enforce the constraints (6.1). In theory, our algorithm will ensure that

$$\frac{D_{ij}}{t_{ij}} = 1$$

at optimality, so choosing $\alpha < 1$ sets a target value t_{ij} that allows some overlap of the areas of the respective circles, which means that a “relaxed” version of the constraints (6.1) on the d_{ij}’s is enforced. Similarly, $\alpha = 1$ means that there should be no overlap and the circles should intersect at exactly one point on their boundaries. This is illustrated in Figure 6.1. Note that the parameters t_{ij} may be interpreted as a generalization of the constant d in Etawil and Vannelli [32] since we need to set a different target for each pair of circles because of the variation among the radii.

In practice, we choose α empirically in such a way that we achieve a reasonable separation between all pairs of circles. The choice of α is explored further in the

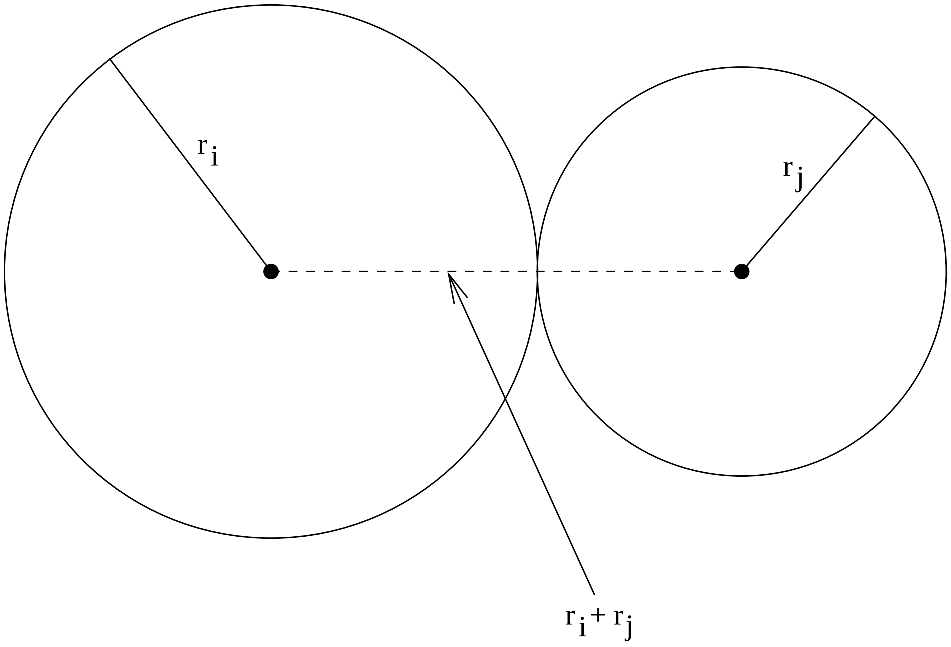


Figure 6.1: Motivation of the concept of target distances

numerical experiments presented in Chapter 7. We now proceed to describing the mathematical model used to enforce the target distances between pairs of circles.

6.2.3 Enforcing the Target Distances

For each pair of circles i and j, if their overlap becomes too large with respect to the target distance t_{ij}, we penalize it by introducing the following term in the objective function of the AR model:

$$f\left(\frac{D_{ij}}{t_{ij}}\right)$$

where the function f is defined by

$$f(z) = \frac{1}{z} - 1, \quad z > 0.$$

The choice of $f(z)$ is inspired by the work of Etawil and Vannelli [32]. The function (of one variable) $f(z)$ is continuously differentiable and strictly decreasing on $(0, \infty)$. We exploit these properties of f for our purposes as follows. We start by placing the circles i and j in some initial configuration where $D_{ij} \gg t_{ij}$, and so $\frac{D_{ij}}{t_{ij}} > 1$. Then, as the ratio $\frac{D_{ij}}{t_{ij}}$ decreases with the effect of the "attractor" component of the objective function, the function f behaves as a monotonically increasing "repeller", and we let them interact until an equilibrium is attained.

By properly adjusting the value of the parameter α, we aim to attain this equilibrium at the point where $\frac{D_{ij}}{t_{ij}} \approx 1$, i.e. where the target distance is attained. In other words, the "attractor" component of our objective function makes the two circles move closer together and pulls them towards a layout where $D_{ij} = 0$, while the "repeller" component prevents the circles from overlapping.

6.2.4 Additional Design Features of the AR Model

Beyond the use of target distances, the AR model offers the user two other features for layout design.

Firstly, the AR model offers the possibility to distinguish between two kinds of modules: those that are mobile and those that are fixed (pads). This is motivated by our intended application of this approach to macro-cell layout problems. We present several such examples in Chapter 7. Another motivation is that it allows for a multiple-round strategy for attacking large problems. The idea is that in each round, the model is solved and some modules are chosen to be fixed before we re-solve in the next round.

Hence, in the AR model, we let M and P denote the set of (mobile) modules and the set of (fixed) pads respectively. Note that target distances are introduced only for pairs of mobile modules. For all mixed pairs, only the attractor term is present in the objective function. That is, we permit overlap of a module and a pad, and interpret such overlap as an indication that the module in question should be placed very close to the pad.

Secondly, the ability to specify bounds on the desired dimensions of the rectangular facility, which is available in St-2, is also available in AR in the sense that it allows the user to specify bounds $w_T^{\min} \leq w_T^{\max}$ on the width of the facility, and $h_T^{\min} \leq h_T^{\max}$ on the height. In particular, if the user knows in advance that the facility should have width $\bar{w}$ and height $\bar{h}$, then these constraints can be enforced by setting

$$w_T^{\min} = w_T^{\max} = \bar{w}, \quad \text{and} \quad h_T^{\min} = h_T^{\max} = \bar{h}.$$

6.2.5 The AR Model

The above ideas yield the AR model:

$$
\text{(AR)}\quad
\begin{array}{ll}
\displaystyle\min_{\substack{(x_i, y_i),\, i \in M,\\ h_T, w_T}} & \displaystyle\sum_{i,j \in M \cup P} c_{ij} D_{ij} \quad + \quad \sum_{i,j \in M} f\left(\tfrac{D_{ij}}{t_{ij}}\right) \\
\text{subject to} & \\
\tfrac{1}{2} w_T \geq x_i + r_i \quad \text{and} & \tfrac{1}{2} w_T \geq r_i - x_i, \text{ for all } i \in M, \\
\tfrac{1}{2} h_T \geq y_i + r_i \quad \text{and} & \tfrac{1}{2} h_T \geq r_i - y_i, \text{ for all } i \in M, \\
& w_T^{\max} \;\geq\; w_T \;\geq\; w_T^{\min}, \\
& h_T^{\max} \;\geq\; h_T \;\geq\; h_T^{\min}.
\end{array}
$$

We point out that the AR model has only linear constraints (on the variables x_i, y_i, h_T, and w_T). From an optimization point of view, this is a significant advantage over St-2. In Section 7.1 we discuss the methodology we employed to test the efficiency of this model. However, we first derive in the next section a convex version of the AR model.

6.3 Convex Version of the AR Model

The AR model as stated above is not a convex problem because, although all the constraints are linear (and hence convex), the objective function is not convex. In this section we show that, under the mild assumption that $c_{ij} \neq 0$ for all pairs of modules $i, j \in M$, an appropriate modification of the objective function yields a convex problem. This convex problem can be thought of as a "convexification" of the AR model, and its derivation will naturally lead us to define certain parameters

T_{ij}. These parameters can be viewed as a generalization of the target distances t_{ij}, and also have an interesting interpretation from a practical point of view. This is further discussed in Section 6.3.1 below.

We begin by isolating the source of the non-convexity in the objective function of the AR model. We will use the well-known fact that

> If $f : \Re^m \mapsto \Re$ is twice continuously differentiable then f is convex over a convex set $C \subseteq \Re^m$ if its Hessian $\nabla^2 f$ is positive semidefinite for all points in C.

(This and other elementary results from convex analysis are presented in for example Bertsekas [15].)

Let us rewrite the objective function as

$$\sum_{i,j \in M} c_{ij} D_{ij} \quad + \quad \sum_{i \in M, j \in P} c_{ij} D_{ij} \quad + \quad \sum_{i,j \in P} c_{ij} D_{ij} \quad + \quad \sum_{i,j \in M} \left(\frac{t_{ij}}{D_{ij}} - 1 \right),$$

where we have applied the function f in the argument of the last summation. We observe that

- The term $\sum\limits_{i,j \in P} c_{ij} D_{ij}$ is constant, since the modules in P are fixed pads. Hence we can omit this term from the objective function without changing the optimal solution.

- The term $\sum\limits_{i \in M, j \in P} c_{ij} D_{ij}$ is convex. Indeed, since x_j and y_j are fixed, its Hessian is a diagonal matrix whose diagonal entries are either 2 (for the rows corresponding to x_i and y_i, $i \in M$) or 0 (for all the other rows). Such a matrix is clearly positive semidefinite over the whole space and hence this term is convex.

- Bringing the remaining two summations together, we can rewrite the remainder of the objective function as

$$\sum_{i,j \in M} c_{ij} D_{ij} + \frac{t_{ij}}{D_{ij}} - 1.$$

This is where convexity can fail and where we now focus our attention.

We consider each term in this last summation independently, so without loss of generality, let us consider the term

$$c_{12} D_{12} + \frac{t_{12}}{D_{12}} - 1.$$

For clarity of the presentation, denote c_{12} by c, t_{12} by t and D_{12} by z. We have the following result:

Theorem 6.3.1 *Let $g : \Re^4 \mapsto \Re$ be given by*

$$g(x_1, x_2, y_1, y_2) = c\,z + \frac{t}{z} - 1,$$

where $c > 0$, $t > 0$, and $z > 0$, $z = (x_1 - x_2)^2 + (y_1 - y_2)^2$. The following statements hold for g:

1. *If $z \geq \sqrt{\frac{t}{c}}$, then the Hessian of g is positive semidefinite.*
2. *If $z = \sqrt{\frac{t}{c}}$, then the gradient of g is zero.*

Proof: At any given point (x_1, x_2, y_1, y_2), define the vector

$$w := \begin{pmatrix} (x_1 - x_2) \\ -(x_1 - x_2) \\ (y_1 - y_2) \\ -(y_1 - y_2) \end{pmatrix}$$

and the scalars $\rho := \left(c - \frac{t}{z^2}\right)$ and $\sigma := \frac{t}{z^3}$.

It is straightforward to check that the gradient and Hessian of g are respectively

$$\nabla g = 2\,\rho\,w$$

and

$$\nabla^2 g = \rho \begin{pmatrix} 2 & -2 & 0 & 0 \\ -2 & 2 & 0 & 0 \\ 0 & 0 & 2 & -2 \\ 0 & 0 & -2 & 2 \end{pmatrix} + 4\,\sigma\,w\,w^T.$$

Now, the matrix

$$\begin{pmatrix} 2 & -2 & 0 & 0 \\ -2 & 2 & 0 & 0 \\ 0 & 0 & 2 & -2 \\ 0 & 0 & -2 & 2 \end{pmatrix}$$

has only two distinct eigenvalues, namely 0 and 4, so it is positive semidefinite. Furthermore,

$$z \geq \sqrt{\frac{t}{c}} \quad \Rightarrow \quad \rho \geq 0,$$

and so the first term of $\nabla^2 g$ is positive semidefinite. In the second term, the rank-one matrix $w\,w^T$ is positive semidefinite for any w, and the constant σ is always positive. Hence the second term is also positive semidefinite. Since the sum of positive semidefinite matrices is positive semidefinite, this proves the first claim.

The second claim is immediate:

$$z = \sqrt{\frac{t}{c}} \quad \Rightarrow \quad \rho = 0 \quad \Rightarrow \quad \nabla g = 0.$$

■

For $c_{ij} > 0$, $t_{ij} > 0$, and $z = (x_i - x_j)^2 + (y_i - y_j)^2$, define the piecewise function

$$f_{ij}(x_i, x_j, y_i, y_j) := \begin{cases} c_{ij}\, z + \frac{t_{ij}}{z} - 1, & z \geq \sqrt{\frac{t_{ij}}{c_{ij}}} \\ 2\sqrt{c_{ij}\, t_{ij}} - 1, & 0 \leq z < \sqrt{\frac{t_{ij}}{c_{ij}}}. \end{cases}$$

Figure 6.2 illustrates the function f_{ij} for various values of c_{ij} and t_{ij}. It was generated by taking random vectors $x, y, \Delta x$, and Δy and using h to parametrize the change in z: for each step h, $h = 0, 0.005, 0.010, \ldots, 15$, $z = \|x - y + h(\Delta x - \Delta y)\|_2^2$.

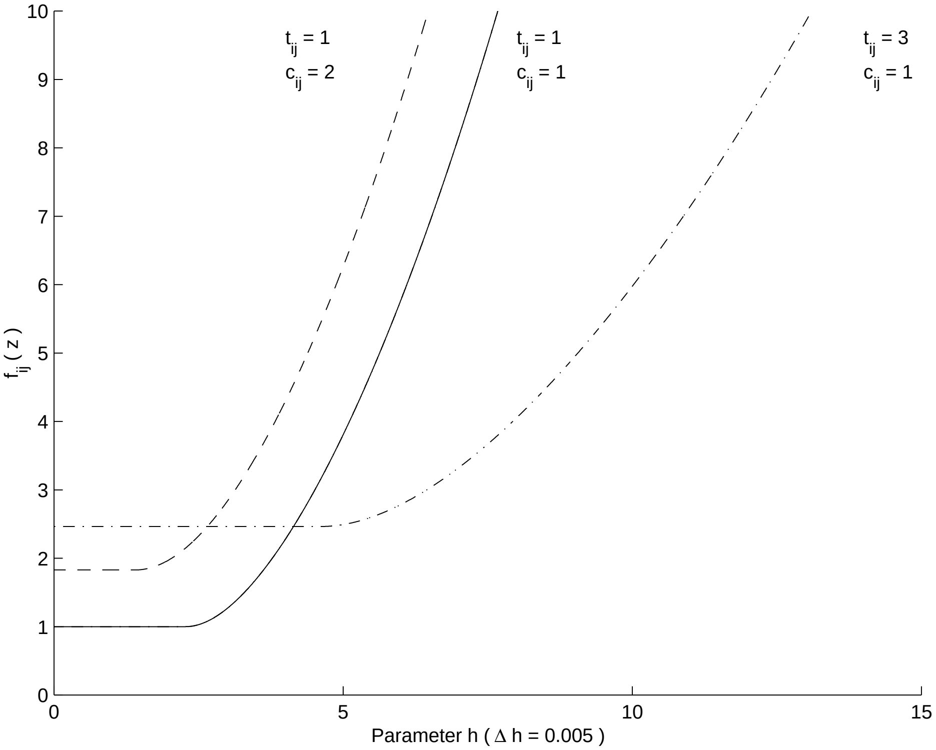


Figure 6.2: Graph of the convex function f_{ij} for several values of c_{ij} and t_{ij}

By construction, f_{ij} is continuous on $\Re^4$, and since by Theorem 6.3.1

$$\nabla f_{ij} \left(\sqrt{\frac{t_{ij}}{c_{ij}}} \right) = 0,$$

f_{ij} is continuously differentiable on $\Re^4$. Also by Theorem 6.3.1 and since

$$\nabla^2 f_{ij} = 0 \quad \text{for} \quad z < \sqrt{\frac{t_{ij}}{c_{ij}}},$$

we see that f_{ij} is convex over $\Re^4$, since its Hessian is everywhere positive semidefinite.

The "convexified" AR model, denoted CoAR, is then:

$$
\text{(CoAR)}\quad
\begin{array}{ll}
\displaystyle\min_{\substack{(x_i,y_i),\, i\in M,\\ h_T,\, w_T}} & \displaystyle\sum_{i\in M, j\in P} c_{ij}D_{ij} \quad + \sum_{i,j\in M} f_{ij}(x_i,x_j,y_i,y_j) \\
\text{subject to} & \\
& \frac{1}{2}w_T \geq x_i + r_i \quad \text{and} \quad \frac{1}{2}w_T \geq r_i - x_i, \text{ for all } i\in M, \\
& \frac{1}{2}h_T \geq y_i + r_i \quad \text{and} \quad \frac{1}{2}h_T \geq r_i - y_i, \text{ for all } i\in M, \\
& w_T^{\max} \;\geq\; w_T \;\geq\; w_T^{\min}, \\
& h_T^{\max} \;\geq\; h_T \;\geq\; h_T^{\min}.
\end{array}
$$

6.3.1 Generalized Target Distances

From the construction of f_{ij}, we observe that the minimum value of f_{ij} is attained for all the positions of modules i and j for which $D_{ij} \leq \sqrt{\frac{t_{ij}}{c_{ij}}}$. This includes the case $D_{ij} = 0$, i.e. both modules completely overlap. Since we do not want such a solution to our layout problem, the algorithm employed to solve CoAR must use a line search that knows the structure of f_{ij} and stops at a point where $D_{ij} \approx \sqrt{\frac{t_{ij}}{c_{ij}}}$, that is, close to the boundary of the flat portion of f_{ij}.

Note that if $D_{ij} \approx \sqrt{\frac{t_{ij}}{c_{ij}}}$, then the corresponding layout of the two modules has D_{ij} proportional to t_{ij}. Therefore such a solution to the model CoAR still enforces the target distances defined in Section 6.2.2.

Furthermore, if we define

$$T_{ij} := \sqrt{\frac{t_{ij}}{c_{ij}}}, \quad i,j\in M, i\neq j,$$

then we can think of the parameter T_{ij} as a generalized target distance for modules i and j which takes both t_{ij} and c_{ij} into account. Indeed, it is reasonable that the target distance be inversely proportional to c_{ij} since, from a practical point of view,

- if c_{ij} is small, then the two modules are likely to be placed far apart in the layout, and correspondingly the generalized target distance should be large;
- if c_{ij} is large, then the opposite reasoning applies and the generalized target distance should be small.

Chapter 7

Numerical Experiments with the AR model

7.1 Solution Methodology

We tested the validity and efficiency of our attractor-repeller paradigm by solving the (non-convex) AR model using the optimization package MINOS 5.3 [1] accessed via the modelling language GAMS (release 2.25) [21] on a SUNSparc. The model has $2|M| + 2$ variables and $4|M| + 4$ inequality constraints, all of which are linear. Therefore only the objective function is non-linear and MINOS specifically exploits this structure by applying a reduced-gradient approach combined with a quasi-Newton algorithm (see Murtagh and Saunders [73]). This is a significant advantage for the AR model because the quasi-Newton algorithm is generally superlinearly convergent. In practice, this means that we are able to solve the AR model quite efficiently even for large $|M|$. Some empirical evidence for the fact that the AR

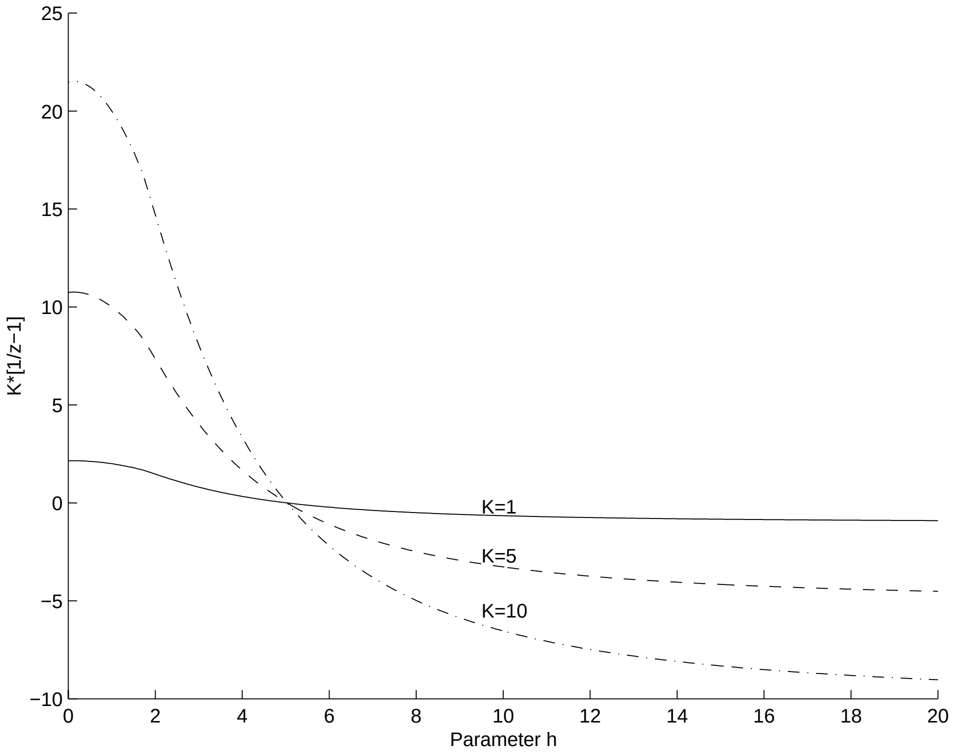


Figure 7.1: Effect of scaling on the function $f(z) = \frac{1}{z} - 1, z > 0$

model can be solved more efficiently than the Stage-1/Stage-2 sequence of the NLT method is presented in Section 7.2.

To accentuate the effect of the repeller function $f(z) = \frac{1}{z} - 1$ for $z < 1$, we scale it by a large constant K. Indeed, even mild scaling of f can significantly increase the "barrier" effect that occurs as soon as z decreases below 1, as the graphs in Figure 7.1 show. Figure 7.1 was generated by taking random vectors x, y, Δx, and Δy and using h to parametrize the change in D_{ij}. For each step h, $h = 0, 0.005, 0.010, \ldots, 20$, $z = \|x - y + h(\Delta x - \Delta y)\|_2^2$. Note that we set $t_{ij} = 1$.

Hence the problem we solve is

$$
\text{(AR)}\quad
\begin{array}{ll}
\displaystyle\min_{(x_i,y_i),h_T,w_T} & \displaystyle\sum_{i,j\in M\cup P} c_{ij}D_{ij} \quad + \quad K \cdot \sum_{i,j\in M} f\left(\tfrac{D_{ij}}{t_{ij}}\right) \\
\text{subject to} & \\
& \tfrac{1}{2}w_T \geq x_i + r_i \quad \text{and} \quad \tfrac{1}{2}w_T \geq r_i - x_i, \text{ for all } i \in M, \\
& \tfrac{1}{2}h_T \geq y_i + r_i \quad \text{and} \quad \tfrac{1}{2}h_T \geq r_i - y_i, \text{ for all } i \in M, \\
& w_T^{\max} \;\geq\; w_T \;\geq\; w_T^{\min}, \\
& h_T^{\max} \;\geq\; h_T \;\geq\; h_T^{\min}.
\end{array}
$$

For our numerical tests, we chose

$$K = 10 \;\cdot\; \left(\sum_{i,j\in M} c_{ij} \right)$$

so that K clearly dominates all the cost coefficients corresponding to pairs of mobile modules in the objective function of AR.

Since the algorithm is an iterative method, MINOS requires the user to supply an initial configuration. It is not clear a priori what the "best" starting configuration is, so we place the centres of the $|M|$ mobile modules at regular intervals around a circle of radius $r = w_T^{\max} + h_T^{\max}$. Thus, letting $\theta_i = \frac{2\pi(i-1)}{|M|}$ and $r = w_T^{\max} + h_T^{\max}$, we initialize the centre (x_i, y_i) of the i^{th} mobile module to

$$x_i = r\cos\theta_i, \quad y_i = r\sin\theta_i, \quad i = 1, \ldots, |M|.$$

We did not solve the CoAR model because this would require a carefully designed line search procedure, as discussed in Section 6.3.1. Furthermore, as the results in Section 7.2 show, the non-convex AR model with scaling by K performs remarkably well in practice, in spite of its lack of convexity.

7.2 Numerical Results

We present the results of applying the AR approach to four layout problems. These examples illustrate the characteristics of the AR model. In the first three examples some of the modules are fixed. Of course, our model applies equally to problems where the position of every module may vary. This is illustrated in the last example.

7.2.1 Macro-Cell Placement Example

We first apply our approach to a macro-cell layout example in which 8 modules out of 19 are fixed. The *optimal* layout for this problem is shown in Figure 7.2. This layout is optimal because any other layouts are obtained by pairwise cell interchanges, and these clearly increase the total “wirelength”.

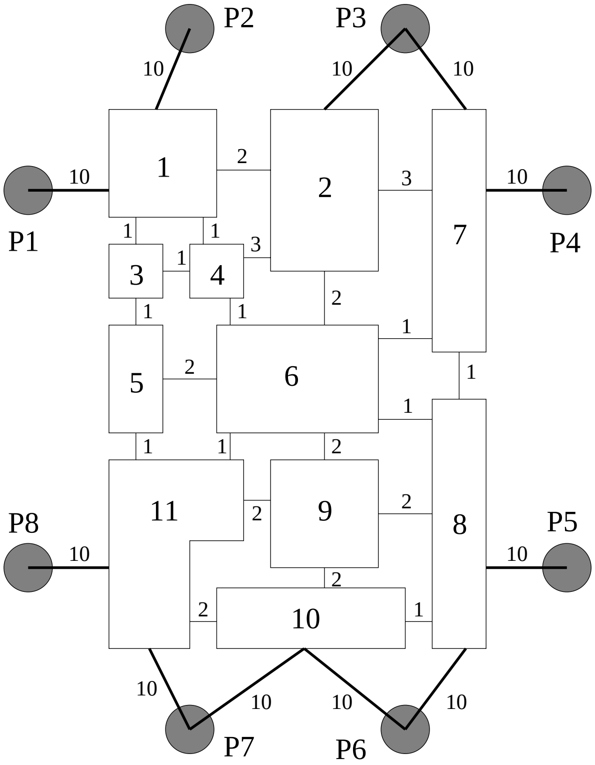


Figure 7.2: Optimal layout for the first macro-cell example

In Figure 7.2, the costs c_{ij} are indicated on the connections of the layout, and the modules P1,. . .,P8 are fixed. (They represent peripheral pads in the macro-cell circuit design.)

For our numerical tests, we used

$$w_T^{\min} = 8, \quad w_T^{\max} = 10, \quad h_T^{\min} = 10, \quad h_T^{\max} = 14$$

and the radii for the mobile modules M1, ..., M11 are (respectively)

$$1, 1.225, 0.5, 0.5, 0.707, 1.225, 1, 1, 1, 0.866, 1.225,$$

while each of the eight pads has radius equal to 0.5. The initial configuration used is shown in Figure 7.3.

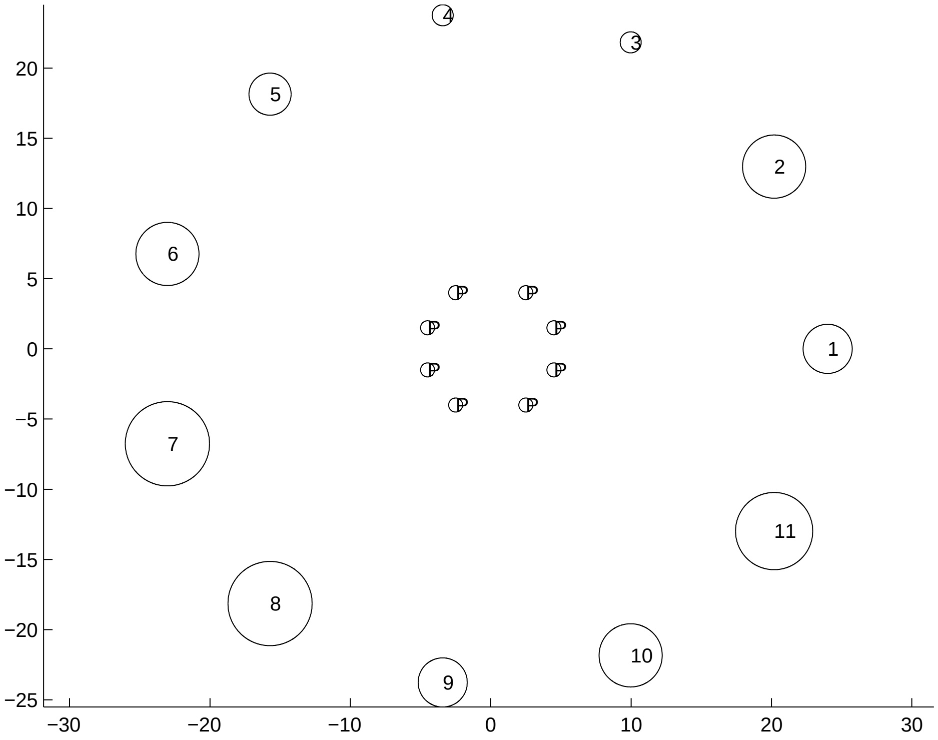
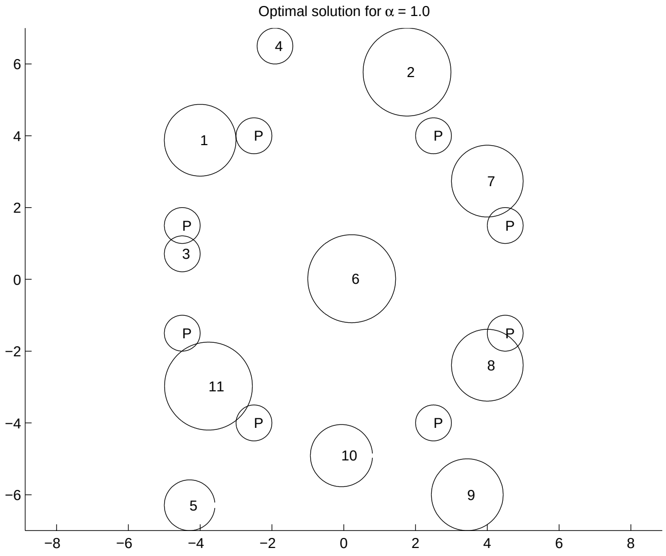
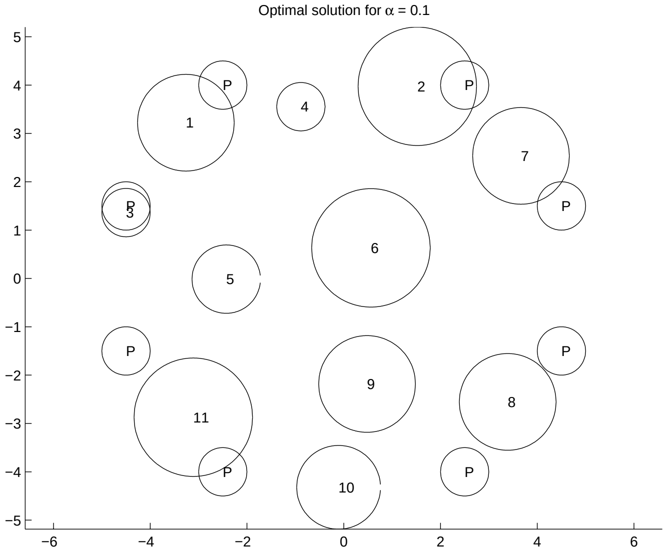
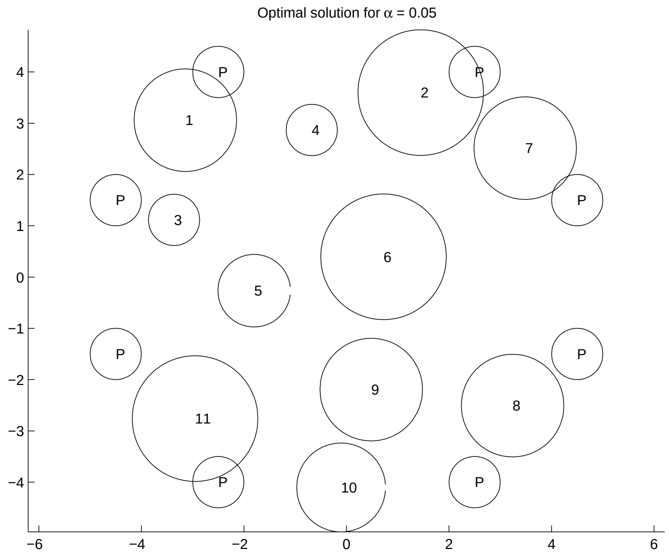


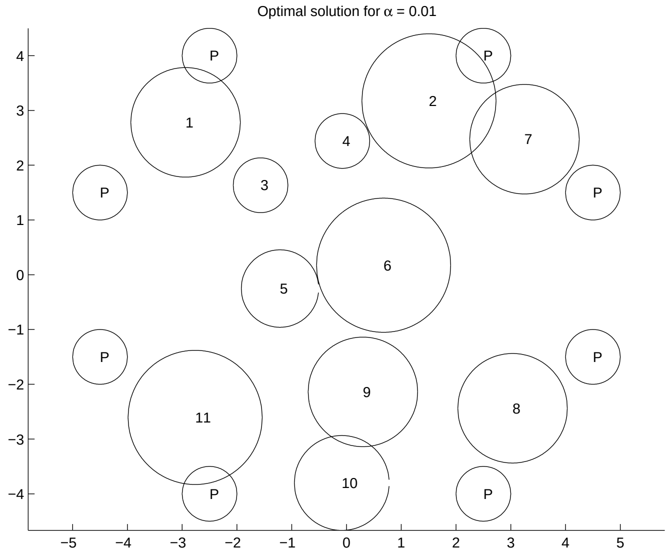
Figure 7.3: Starting configuration for the first macro-cell example

The results for varying values of α are as follows:



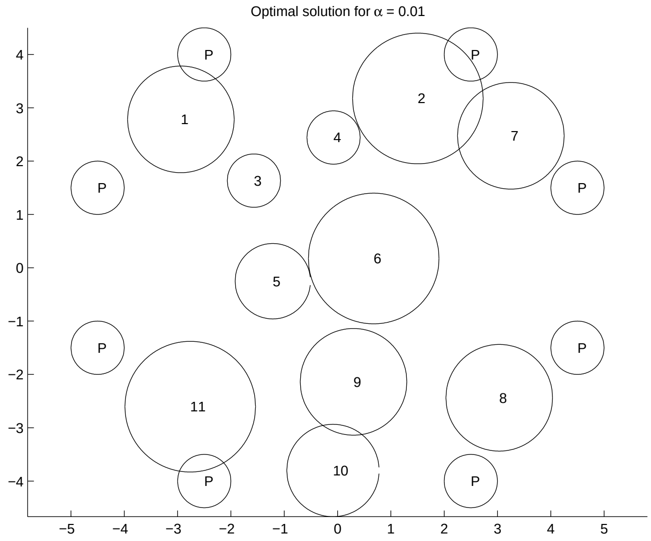


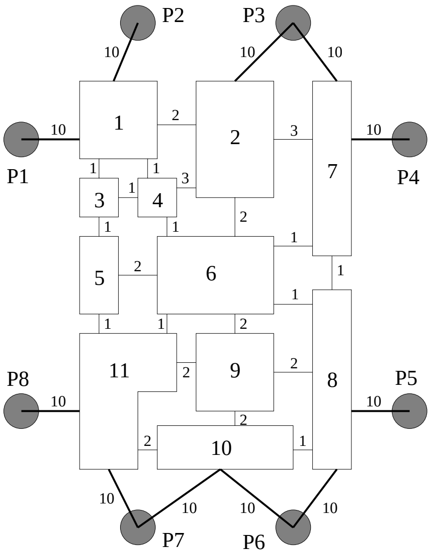




Surprisingly, we find that setting $\alpha = 1.0$ results in excessive separation of the modules, which are thrown against the boundaries of the facility. Therefore we need to reduce α in order to obtain a good placement.

Furthermore, we find that the solution for $\alpha = 0.01$ yields a layout which is very similar to the (known) global optimal solution, as can be seen if we display them side by side:



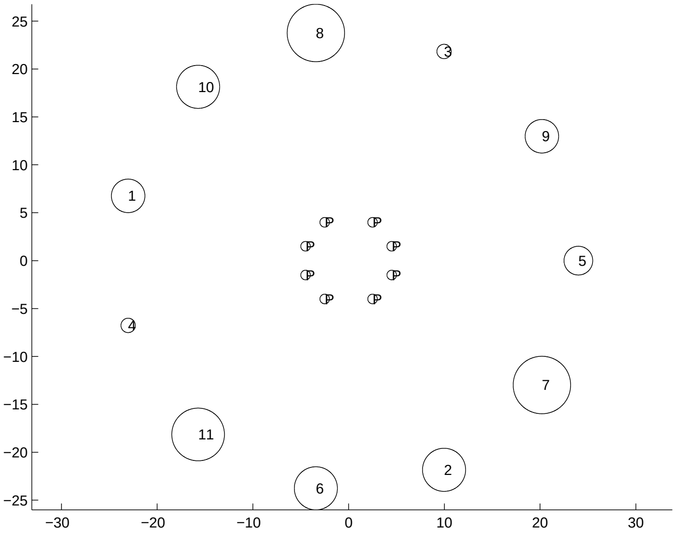


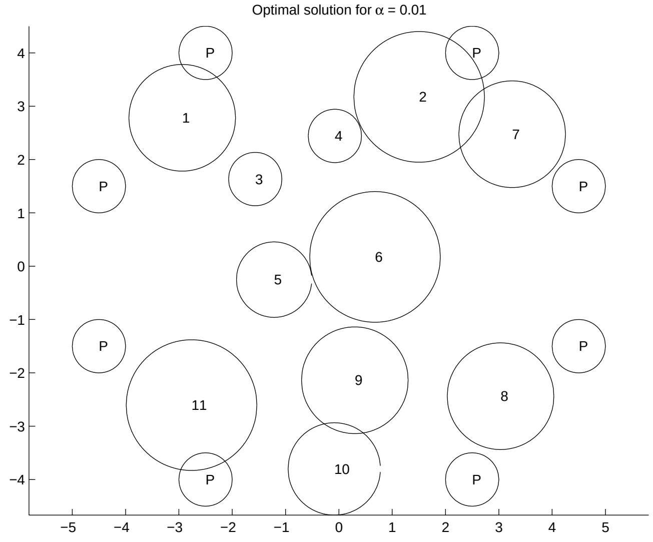
Clearly, the modules that have connections to pads are all located in the right areas, and all the modules are quite close to their (known) optimal positions.

Alternative Initial Configurations

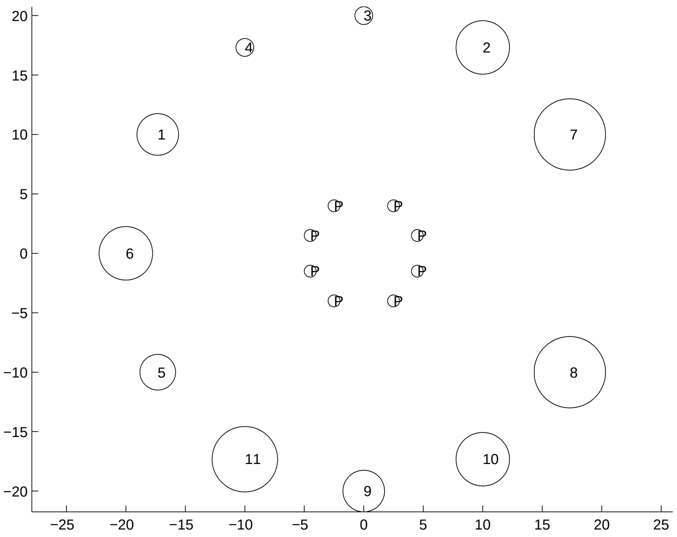
It is unclear at this point how much the success of the AR model depends upon our choice of initial configuration. Therefore we tested the robustness of the model for this example by trying two alternative initial configurations.

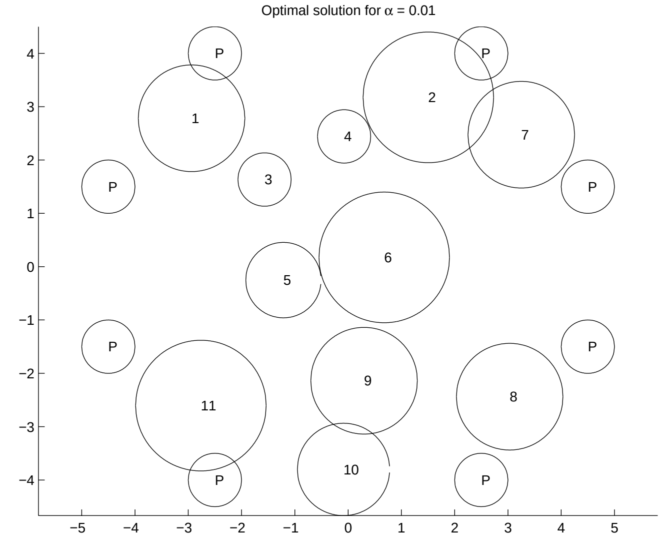
The first alternative configuration and corresponding solution are:





The second alternative configuration and corresponding solution are:

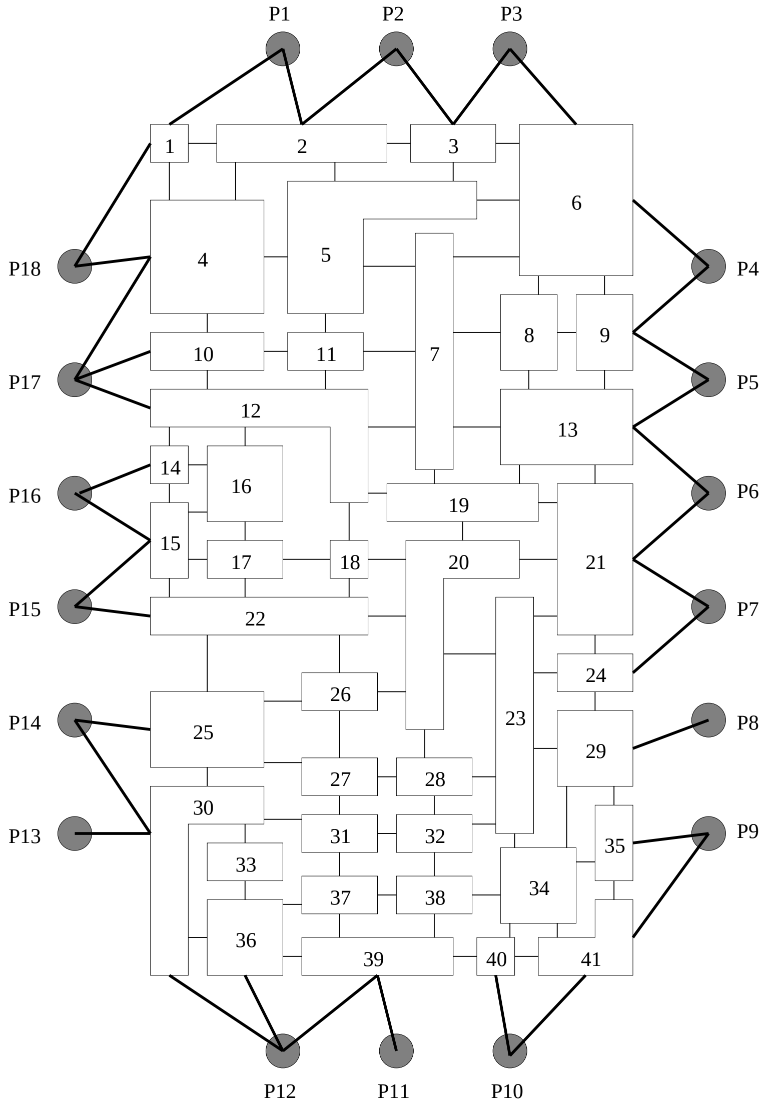




We see that with all three starting configurations, we always obtain the same optimal solution. This illustrates the robustness of the AR model when applied to this test problem.

7.2.2 Medium-Sized Macro-Cell Placement Example

Next, we experimented with a larger test problem of macro-cell layout. Again, it has the following *optimal* layout:



where $c_{ij} = 1$ for the connections with the lighter edges (these are connections between mobile modules) and $c_{ij} = 10$ for the connections with the darker edges (these are connections between fixed pads and mobile modules).

For our numerical tests, we used

$$w_T^{\min} = 25, \quad w_T^{\max} = 30, \quad h_T^{\min} = 50, \quad h_T^{\max} = 60$$

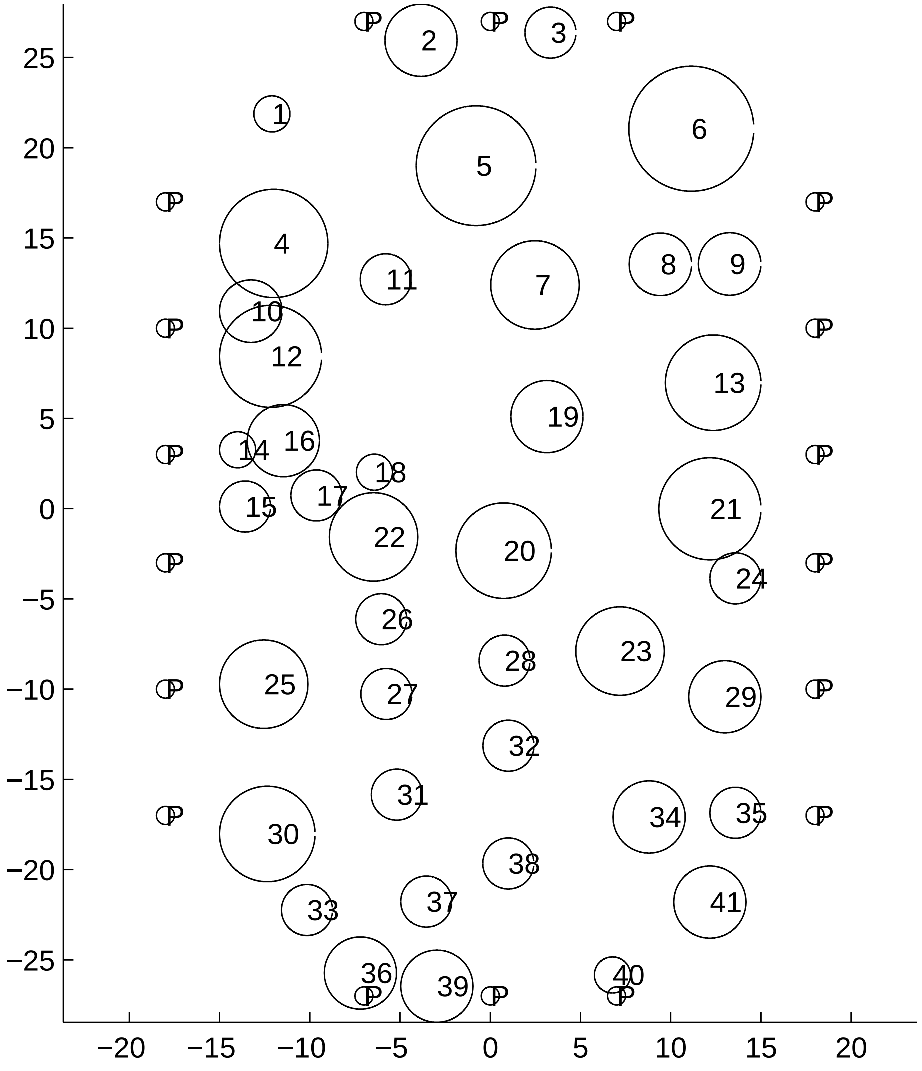
and the radii for the mobile modules M1, . . ., M41 are (respectively)

1, 2, 1.414, 3, 3.317, 3.464, 2.450, 1.732, 1.732, 1.732, 1.414, 2.828, 2.646, 1, 1.414, 2, 1.414, 1, 2, 2.646, 2.828, 2.450, 2.450, 1.414, 2.450, 1.414, 1.414, 1.414, 2, 2.646, 1.414, 1.414, 1.414, 2, 1.414, 2, 1.414, 1.414, 2, 1, 2

while each of the 18 pads has radius equal to 0.5.

The initial configuration used was as described above, that is, the 41 mobile modules were placed at regular intervals on a circle of radius 200.

The result for $\alpha = 0.01$ is:



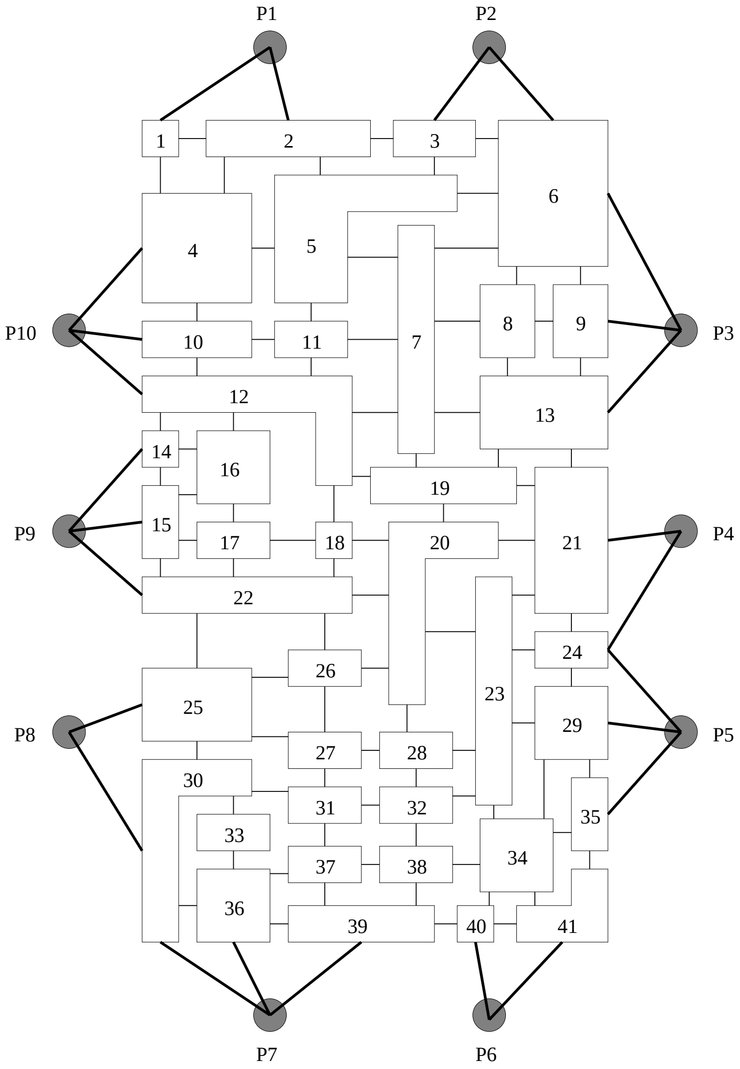
Note that, in comparison with the example in Section 7.2.1, we have significantly increased the number of modules in the problem, and furthermore, the ratio of fixed pads to mobile modules has decreased from $8/11 \approx 0.73$ to $18/41 \approx 0.44$. Nonetheless, we again obtain from the AR model (with the same value of $\alpha = 0.01$)

a solution whose layout is very similar to the (known) global optimal solution, i.e. the modules are all quite close to their optimal positions.

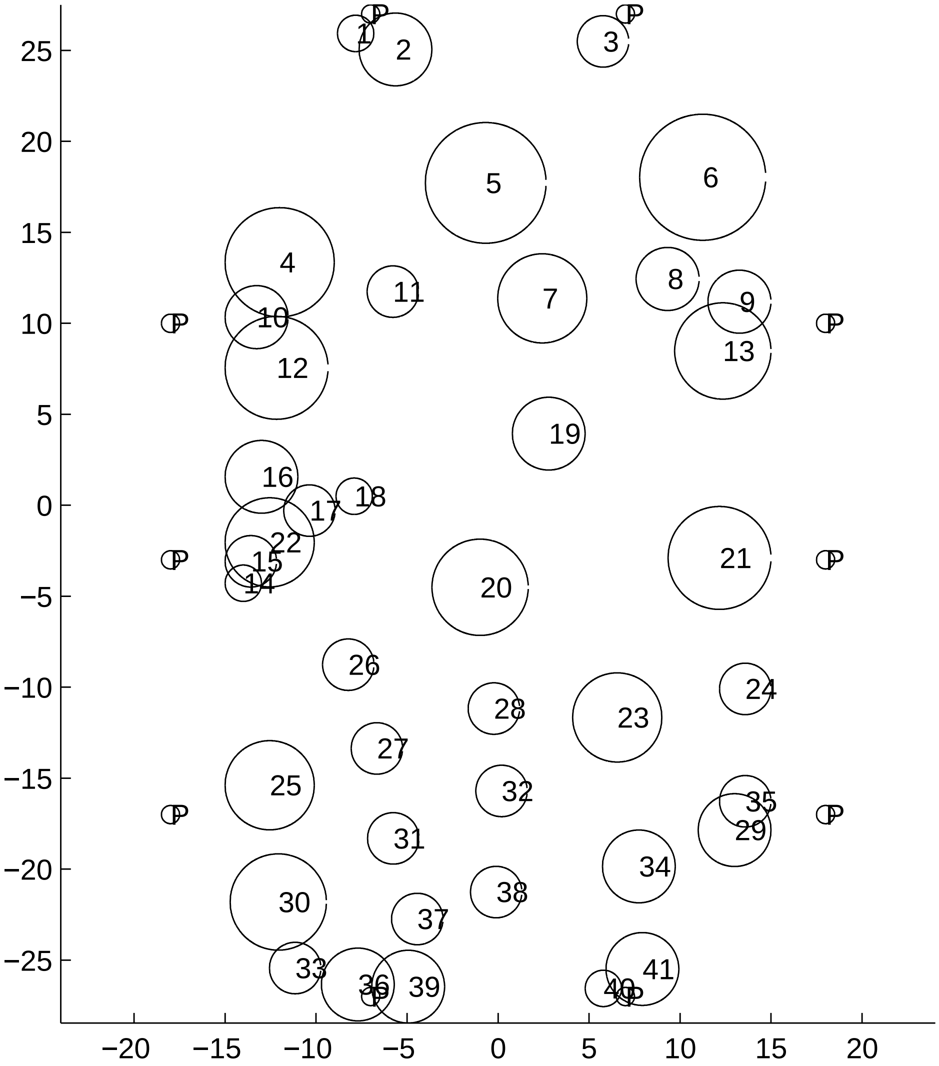
7.2.3 Alternative Medium-Sized Macro-Cell Example

To further test the robustness of the AR model for macro-cell applications, we chose our third example to be the same circuit as in the second example, but with even fewer fixed pads. This example has now only 10 pads and therefore a ratio of fixed pads to mobile modules of $10/41 \approx 0.24$. The dimensional bounds, radii, and initial configuration were left unchanged, since our purpose is to make the proportion of fixed pads *smaller* and observe the effect on the overall layout.

The optimal layout for this third example is:



And the result (again leaving $\alpha = 0.01$ unchanged) is:



We see that the obtained layout is not as clearly defined. Indeed, there are two clusters of modules, one consisting of modules M29 and M35, and the other consisting of modules M14, M15, M16, M17, and M22, whose circles overlap significantly and whose relative positions are either unclear or different from those in the optimal layout. Nonetheless, the clusters themselves are in the correct area of the layout and thus the overall result is still an excellent approximation of the optimal layout.

We tried varying α between 0.005 and 0.02 with similar results. The key observation is that a lower ratio of fixed pads to mobile modules has little effect on the overall optimal positions of the circles in the layout.

7.2.4 Facility Layout Example

Lastly, we applied the AR approach to the 10-facility example from van Camp et al. [26]. This problem comes from an existing production plant and hence the dimensions of the facility are fixed in advance. Therefore we set

$$w_T^{\min} = w_T^{\max} = 25, \quad h_T^{\min} = h_T^{\max} = 51.$$

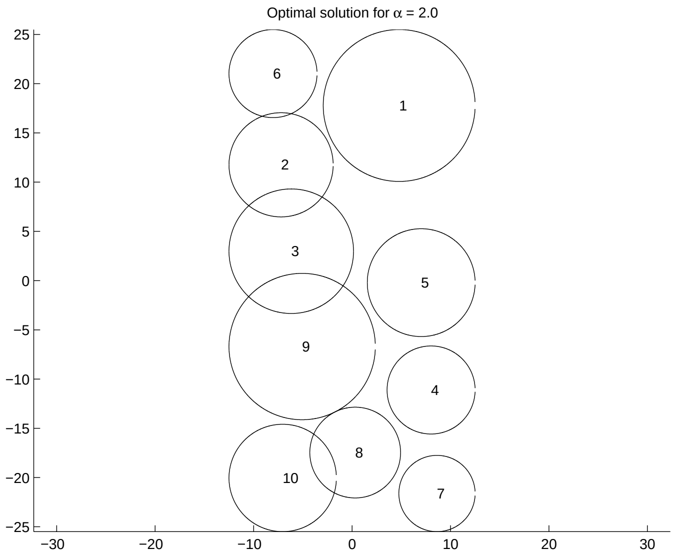
This problem has 10 mobile modules (departments) and, in contrast with our previous examples, has *no* fixed modules. We use the same approximating circles as in van Camp et al. [26], therefore the radii for M1, ..., M10 are set to

$$\{7.7136, 5.2915, 6.3246, 4.4721, 5.4772, 4.4721, 3.8730, 4.6098, 7.4330, 5.4544\}.$$

These radii were chosen so that the radius of each circle equals half the square root of the (given) area for each department. Also as in van Camp et al. [26], the costs c_{ij} are as follows:

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
M1	0	0	0	0	0	218	0	0	0	0
M2	0	0	0	0	0	148	0	0	296	0
M3	0	0	0	28	70	0	0	0	0	0
M4	0	0	28	0	0	28	70	140	0	0
M5	0	0	70	0	0	0	0	210	0	0
M6	218	148	0	28	0	0	0	0	0	0
M7	0	0	0	70	0	0	0	0	0	28
M8	0	0	0	140	210	0	0	0	0	888
M9	0	296	0	0	0	0	0	0	0	59.2
M10	0	0	0	0	0	0	28	888	59.2	0

Here is the best layout we obtained:



We see that (for $\alpha = 2.0$) the AR model yields a solution exhibiting a fairly good separation of the circles. The optimal layout for this problem is not known, and interestingly this solution is different from the Stage-2 solution reported in van Camp et al. [26].

Furthermore, we can compare the AR model and the Stage-1/Stage-2 sequence of models of the NLT method in terms of the computational effort required to find a solution for the layout of the circles. We took the cumulative summary of computational data reported in van Camp et al. [26] to solve the Stage-1 and Stage-2 problems and compared the number of iterations with the results reported by MINOS. The results are presented in Table 7.1 and illustrate the advantage of having an efficient algorithm as is the case for the AR model.

	Number of iterations
NLT method (Stage-1 & Stage-2)	6607
AR model with $\alpha = 2.0$	31

Table 7.1: Comparison of iteration counts between NLT and AR

Finally, we remark that Etawil and Vannelli studied three possible choices of "repeller" functions:

- $z - \ln(z) - 1$;
- $z + e^{1-z} - 2$;
- $\frac{1}{z} - 1$.

Any of these choices can be used in the AR model. Our experimental results showed that the choice of $\frac{1}{z} - 1$ outperformed the other two choices for the layout examples which we considered.

Chapter 8

Conclusions and Directions for Future Research

In this thesis we derived two strengthened semidefinite programming, and hence convex, relaxations for the Max-Cut problem. We provided computational evidence for their strengthening properties and proved several interesting properties of the geometry of the tighter relaxation. We also derived a convex relaxation for the VLSI layout problem where the modules are allowed to have (significantly) different sizes. Furthermore, we provided computational evidence for the potential of this relaxation to provide a reliable global placement for this hard layout problem.

Many exciting questions about both the theoretical and the computational aspects of these relaxations remain. We conclude by outlining in more detail the achievements of this thesis as well as some of these possibilities for future research.

8.1 New Semidefinite Relaxations for the Max-Cut Problem

We derived and studied two strengthened semidefinite programming relaxations for the Max-Cut problem and proved several interesting properties of these relaxations. In particular, we proved that the tighter of these two relaxations corresponds to a relaxation of the cut polytope that is strictly contained in the intersection of the elliptope and the metric polytope. We also proved that for the tighter relaxation SDP3, the rank of the matrix variable Y completely characterizes the faces of dimension 0 and 1 of the cut polytope in $\mathcal{S}^n$ (upon projecting its first column into $\mathcal{S}^n$ via the sMat operator). This is an improvement over the Goemans-Williamson relaxation which characterizes only the zero-dimensional faces (vertices) in this way.

The study of the structure of the barycenter of the feasible sets of the strengthened relaxations SDP2 and SDP3 led us to explicitly exhibit a projection of the feasible sets of these relaxations onto a face of the cone of positive semidefinite matrices where Slater's condition holds for both the primal and the dual SDPs, and hence strong duality also holds. We also presented computational results demonstrating that the relaxations SDP2 and SDP3 frequently yield a significant improvement over SDP1, and that SDP3 often finds the *optimal* value of MC.

From a theoretical point of view, it would be quite interesting to characterize all the weighted graphs, or at least some classes of weighted graphs, for which SDP3 always yields the optimal value of the maximum cut. Perhaps an even more desirable result would characterize the graphs for which SDP3 finds an optimal cut; our theoretical and computational results suggest that many weighted graphs belong in this category. More ambitiously, it seems plausible that a guaranteed approximation bound tighter than the Goemans and Williamson bound of $\alpha_{GW} \approx 0.87856$ holds for the optimal value of SDP3. It would be quite interesting to prove such a worst-case bound, although one must keep in mind Håstad's result that it is NP-hard to find an ρ-approximation algorithm for Max-Cut with ρ greater than 0.9412 [52].

Another intriguing question is how to design a more sophisticated rounding procedure than the simple one which we used. The rounding procedure should make better use of the information contained in the optimal matrix Y for SDP3. Such a procedure might be related to the well-known randomized algorithm of Goemans and Williamson [41], although their worst-case analysis does not apply to Ising spin glass problems because these problems have negative edge weights, or it might make use of the ideas in Nesterov's relative accuracy analysis [76]. This is the focus of ongoing research.

In this thesis, we did not address the design of computational algorithms tailored to solve the new SDP relaxations. In particular, we did not address the question of how to solve these relaxations efficiently by exploiting their structure and sparsity. Both relaxations have structured constraints, the sparsity of which increases rapidly with n. Current and future research efforts on sparse structured problems (see for example [14, 22, 23, 24, 25, 27, 35, 36, 74]) may lead to efficient methods for solving the relaxations. Such advances in the computational technology for solving SDPs may well make it worthwhile for enumerative techniques to invest the computational effort required to compute the SDP3 bound at the root node of the enumeration tree, and perhaps also at other branching nodes, in order to reduce the combinatorial explosion in the enumerative process.

Finally, we point out that the direct lifting approach in Section 2.4.2 shows that the strengthened relaxations SDP2 and SDP3 can also be used to compute bounds for quartic Max-Cut problems, that is, polynomial problems of the form:

$$\begin{aligned} \max \quad & \sum_{1\leq i<j<k<l\leq n} H_{T(i,j),T(k,l)}\, v_i v_j v_k v_l + \sum_{1\leq i<j\leq n} H_{0,T(i,j)}\, v_i v_j \\ \text{s.t.} \quad & \\ & v_i^2 = 1, \quad i = 1, \dots, n. \end{aligned}$$

Further research is necessary to establish the efficiency of the new relaxations for such problems as well as the theoretical properties which hold in this more general context.

8.2 New AR Model for VLSI Layout

Our computational results show that the AR (Attractor-Repeller) model was robust on several “optimal” floorplanning problems irrespective of the number of fixed pads on the boundary. This new model *globally* places modules or cells of different sizes in optimal two-dimensional locations. This global optimality is the most important feature of this new model. When compared with the Stage-1/Stage-2 sequence of the NLT method, the new AR model finds in just one stage, and furthermore much more quickly and efficiently, a good initial point for the Stage-3 model of the NLT method.

We isolated the source of the non-convexity in the AR model and derived a convex version of the AR model, CoAR. We showed that the convex CoAR model retains much of the structure of the AR model, and in particular the derivation of the CoAR model leads to the definition of “generalized” target distances. However, we further observed that the numerical solution of this model requires a carefully designed line search algorithm. Further work is required to implement such an algorithm.

Our results also show that the choice of the parameter α in the AR model has a significant influence on the optimal solution. Hence the question of how to choose α appropriately deserves further study.

Finally, we believe that the ideas presented in Sections 6.2 and 6.3 are applicable to other layout problems. For example, they may be quite effective for solving another important layout problem in circuit design, namely the standard-cell problem (see for example Etawil and Vannelli [32]). In view of our computational results here, this is yet another promising avenue for future research.

Appendix A

Matrices $\bar{B}$ and $\bar{\Lambda}$ for Example 2.5.8

The matrices $\bar{B}$ and $\bar{\Lambda}$ that follow satisfy

$$\|H \circ (A - \bar{B})\|_F^2 - \operatorname{trace} \bar{\Lambda}\bar{B} = 2.81 \times 10^{-4}.$$

The matrix $\bar{B}$ is formed with the following columns:

```
Columns 1 through 7


 1.0074    1.0000   -0.6475    1.0000   -0.6475    0.3026    1.0000
 1.0000    1.0000   -0.6427    0.9927   -0.6427    0.3004    0.9927
-0.6475   -0.6427    1.0020   -0.6427    0.3013   -0.6487   -0.6427
 1.0000    0.9927   -0.6427    1.0000   -0.6427    0.3004    0.9927
-0.6475   -0.6427    0.3013   -0.6427    1.0020   -0.6487   -0.6427
 0.3026    0.3004   -0.6487    0.3004   -0.6487    1.0026    0.3004
 1.0000    0.9927   -0.6427    0.9927   -0.6427    0.3004    1.0000
-0.6475   -0.6427    0.3013   -0.6427    0.3013    0.0257   -0.6427
 0.3026    0.3004   -0.6487    0.3004    0.0257    0.3001    0.3004
 0.3026    0.3004    0.0257    0.3004   -0.6487    0.3001    0.3004
 1.0000    0.9927   -0.6427    0.9927   -0.6427    0.3004    0.9927
 0.9300    0.9232   -0.6500    0.9232   -0.6500    0.3804    0.9232
-0.6475   -0.6427    0.9307   -0.6427    0.3369   -0.6487   -0.6427
-0.6475   -0.6427    0.3369   -0.6427    0.9307   -0.6487   -0.6427
-0.6475   -0.6427    0.3369   -0.6427    0.3369    0.0257   -0.6427
 1.0000    0.9927   -0.6427    0.9927   -0.6427    0.3004    0.9927
```

```
Columns 8 through 14


-0.6475    0.3026    0.3026    1.0000    0.9300   -0.6475   -0.6475

-0.6427    0.3004    0.3004    0.9927    0.9232   -0.6427   -0.6427

 0.3013   -0.6487    0.0257   -0.6427   -0.6500    0.9307    0.3369

-0.6427    0.3004    0.3004    0.9927    0.9232   -0.6427   -0.6427

 0.3013    0.0257   -0.6487   -0.6427   -0.6500    0.3369    0.9307

 0.0257    0.3001    0.3001    0.3004    0.3804   -0.6487   -0.6487

-0.6427    0.3004    0.3004    0.9927    0.9232   -0.6427   -0.6427

 1.0020   -0.6487   -0.6487   -0.6427   -0.6500    0.3369    0.3369

-0.6487    1.0026    0.3001    0.3004    0.3804   -0.6487    0.0257

-0.6487    0.3001    1.0026    0.3004    0.3804    0.0257   -0.6487

-0.6427    0.3004    0.3004    1.0000    0.9232   -0.6427   -0.6427

-0.6500    0.3804    0.3804    0.9232    1.0000   -0.6500   -0.6500

 0.3369   -0.6487    0.0257   -0.6427   -0.6500    1.0020    0.3013

 0.3369    0.0257   -0.6487   -0.6427   -0.6500    0.3013    1.0020

 0.9307   -0.6487   -0.6487   -0.6427   -0.6500    0.3013    0.3013

-0.6427    0.3004    0.3004    0.9927    0.9232   -0.6427   -0.6427
```

```
Columns 15 through 16


-0.6475    1.0000
-0.6427    0.9927
 0.3369   -0.6427
-0.6427    0.9927
 0.3369   -0.6427
 0.0257    0.3004
-0.6427    0.9927
 0.9307   -0.6427
-0.6487    0.3004
-0.6487    0.3004
-0.6427    0.9927
-0.6500    0.9232
 0.3013   -0.6427
 0.3013   -0.6427
 1.0020   -0.6427
-0.6427    1.0000
```

and the matrix $\bar{\Lambda}$ is formed with the following columns:

```
Columns 1 through 7


 0.0148   -0.0000    0.0050   -0.0000    0.0050    0.0052   -0.0000

-0.0000    0.0000         0         0         0         0         0

 0.0050         0    0.0039         0    0.0025    0.0027         0

-0.0000         0         0    0.0000         0         0         0

 0.0050         0    0.0025         0    0.0039    0.0027         0

 0.0052         0    0.0027         0    0.0027    0.0053         0

-0.0000         0         0         0         0         0    0.0000

 0.0050         0    0.0025         0    0.0025         0         0

 0.0052         0    0.0027         0         0    0.0001         0

 0.0052         0         0         0    0.0027    0.0001         0

-0.0000         0         0         0         0         0         0

-0.0000         0    0.0000         0    0.0000         0         0

 0.0050         0    0.0014         0         0    0.0027         0

 0.0050         0         0         0    0.0014    0.0027         0

 0.0050         0         0         0         0         0         0

-0.0000         0         0         0         0         0         0
```

```
Columns 8 through 14


 0.0050    0.0052    0.0052   -0.0000   -0.0000    0.0050    0.0050

      0         0         0         0         0         0         0

 0.0025    0.0027         0         0    0.0000    0.0014         0

      0         0         0         0         0         0         0

 0.0025         0    0.0027         0    0.0000         0    0.0014

      0    0.0001    0.0001         0         0    0.0027    0.0027

      0         0         0         0         0         0         0

 0.0039    0.0027    0.0027         0    0.0000         0         0

 0.0027    0.0053    0.0001         0         0    0.0027         0

 0.0027    0.0001    0.0053         0         0         0    0.0027

      0         0         0    0.0000         0         0         0

 0.0000         0         0         0    0.0000    0.0000    0.0000

      0    0.0027         0         0    0.0000    0.0039    0.0025

      0         0    0.0027         0    0.0000    0.0025    0.0039

 0.0014    0.0027    0.0027         0    0.0000    0.0025    0.0025

      0         0         0         0         0         0         0
```

```
Columns 15 through 16


 0.0050   -0.0000
      0         0
      0         0
      0         0
      0         0
      0         0
      0         0
 0.0014         0
 0.0027         0
 0.0027         0
      0         0
 0.0000         0
 0.0025         0
 0.0025         0
 0.0039         0
      0    0.0000
```

Appendix B

Weighted Adjacency Matrix for Problem with $n = 12$ in Table 3.1

0	2	2	0	2	4	0	2	2	2	2	2
2	0	0	2	4	2	2	0	2	2	2	2
2	0	0	0	4	4	2	4	4	0	2	2
0	2	0	0	0	2	0	0	2	4	2	2
2	4	4	0	0	2	2	2	2	4	2	4
4	2	4	2	2	0	2	0	2	2	2	2
0	2	2	0	2	2	0	0	0	4	2	2
2	0	4	0	2	0	0	0	0	4	4	2
2	2	4	2	2	2	0	0	0	2	2	2
2	2	0	4	4	2	4	4	2	0	4	2
2	2	2	2	2	2	2	4	2	4	0	4
2	2	2	2	4	2	2	2	2	2	4	0

Bibliography

[1] *GAMS – The Solver Manuals.* GAMS Development Corporation, Washington, D.C., 1996.

[2] F. ALIZADEH. *Combinatorial Optimization with Interior Point Methods and Semidefinite Matrices.* PhD thesis, University of Minnesota, 1991.

[3] F. ALIZADEH. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optim.*, 5:13–51, 1995.

[4] F. ALIZADEH, J.-P. HAEBERLY, M.V. NAYAKKANKUPPAM, M.L. OVERTON, and S. SCHMIETA. SDPpack user's guide – version 0.9 Beta. Technical Report TR1997–737, Courant Institute of Mathematical Sciences, NYU, New York, NY, June 1997.

[5] M.F. ANJOS and H. WOLKOWICZ. Strengthened semidefinite relaxations via a second lifting for the max-cut problem. To appear in *Special Issue of Discrete Applied Mathematics devoted to Foundations of Heuristics in Combinatorial Optimization*, 2001.

[6] K.M. ANSTREICHER, X. CHEN, H. WOLKOWICZ, and Y. YUAN. Strong dual-

ity for a trust-region type relaxation of the quadratic assignment problem. *Linear Algebra Appl.*, 301(1-3):121–136, 1999.

[7] K.M. ANSTREICHER and H. WOLKOWICZ. On Lagrangian relaxation of quadratic matrix constraints. *SIAM J. Matrix Anal. Appl.*, 22(1):41–55, 2000.

[8] E. BALAS. A modified lift-and-project procedure. *Math. Program.*, 79(1-3, Ser. B):19–31, 1997.

[9] E. BALAS, S. CERIA, and G. CORNUEJOLS. A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Math. Program.*, 58:295–324, 1993.

[10] E. BALAS, S. CERIA, and G. CORNUÉJOLS. Solving mixed 0-1 programs by a lift-and-project method. In *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms (Austin, TX, 1993)*, pages 232–242, New York, 1993.

[11] F. BARAHONA. On cuts and matchings in planar graphs. *Math. Program.*, 60(1, Ser. A):53–68, 1993.

[12] F. BARAHONA, M. GRÖTSCHEL, M. JÜNGER, and G. REINELT. An application of combinatorial optimization to statistical physics and circuit layout design. *Oper. Res.*, 36:493–513, 1988.

[13] F. BARAHONA and A. R. MAHJOUB. On the cut polytope. *Math. Program.*, 36(2):157–173, 1986.

[14] S.J. BENSON, Y. YE, and X. ZHANG. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM J. Optim.*, 10(2):443–461 (electronic), 2000.

[15] D.P. BERTSEKAS. *Nonlinear Programming.* Athena Scientific, Belmont, MA, 1995. 2nd edition 1999.

[16] B. BORCHERS. CSDP 2.3 user's guide. *Optim. Methods Softw.*, 11/12(1-4):597–611, 1999.

[17] B. BORCHERS. CSDP, a C library for semidefinite programming. *Optim. Methods Softw.*, 11/12(1-4):613–623, 1999.

[18] E. BOROS, Y. CRAMA, and P.L. HAMMER. Upper bounds for quadratic $0-1$ maximization. *Oper. Res. Lett.*, 9:73–79, 1990.

[19] E. BOROS and P.L. HAMMER. The max-cut problem and quadratic 0-1 optimization; polyhedral aspects, relaxations and bounds. *Ann. Oper. Res.*, 33(1-4):151–180, 1991. Topological network design (Copenhagen, 1989).

[20] E. BOROS and P.L. HAMMER. Cut-polytopes, Boolean quadric polytopes and nonnegative quadratic pseudo-Boolean functions. *Math. Oper. Res.*, 18:245–253, 1993.

[21] A. BROOKE, D. KENDRICK, and A. MEERAUS. *GAMS – A User's Guide, Release 2.25.* The Scientific Press, South San Francisco, CA, 1992.

[22] S. BURER and R.D.C. MONTEIRO. An efficient algorithm for solving the MAX-CUT SDP relaxation. Technical report, Georgia Tech, Atlanta, GA, 1999.

[23] S. BURER, R.D.C. MONTEIRO, and Y. ZHANG. Interior-point algorithms for semidefinite programming based on a nonlinear programming formulation. Technical Report TR99-27, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1999.

[24] S. BURER, R.D.C. MONTEIRO, and Y. ZHANG. Solving semidefinite programs via nonlinear programming part I: Transformations and derivatives. Technical Report TR99-17, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1999.

[25] S. BURER, R.D.C. MONTEIRO, and Y. ZHANG. Solving semidefinite programs via nonlinear programming part II: Interior point methods for a subclass of SDPs. Technical Report TR99-23, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1999.

[26] D.J. VAN CAMP, M.W. CARTER, and A. VANNELLI. A nonlinear optimization approach for solving facility layout problems. *Eur. J. Oper. Res.*, 57:174–189, 1991.

[27] C. CHOI and Y. YE. Solving sparse semidefinite programs using the dual scaling algorithm with an iterative solver. Technical report, The University of Iowa, Iowa City, IA, 2000.

[28] C. DELORME and S. POLJAK. Combinatorial properties and the complexity of a max–cut approximation. *Eur. J. Comb.*, 14:313–333, 1993.

[29] C. DELORME and S. POLJAK. Laplacian eigenvalues and the maximum cut problem. *Math. Program.*, 62(3):557–574, 1993.

[30] M.M. DEZA and M. LAURENT. *Geometry of Cuts and Metrics*. Springer-Verlag, Berlin, 1997.

[31] Z. DREZNER. DISCON: A new method for the layout problem. *Oper. Res.*, 28(6):1375–1384, 1980.

[32] H.A.Y. ETAWIL and A. VANNELLI. Target distance models for VLSI placement problem. To appear in *Journal of VLSI*.

[33] M.C. FERRIS, M.P. MESNIER, and J.J. MORÉ. NEOS and Condor: Solving optimization problems over the Internet. Technical Report ANL/MCS-P708-0398, Argonne National Laboratory, Mathematics and Computer Science Division, 1998.

[34] K.H FISCHER and J.A. HERTZ. *Spin Glasses.* Cambridge University Press, Cambridge, 1991.

[35] K. FUJISAWA, M. KOJIMA, and K. NAKATA. Exploiting sparsity in primal–dual interior-point methods for semidefinite programming. *Math. Program.*, 79(1-3, Ser. B):235–253, 1997.

[36] K. FUKUDA, M. KOJIMA, K. MUROTA, and K. NAKATA. Exploiting sparsity in semidefinite programming via matrix completion I: general framework. Technical Report B-358, Dept. of Information Sciences, Tokyo Institute of Technology, Tokyo, Japan, 1999.

[37] M.R. GAREY and D.S. JOHNSON. *Computers and Intractability: A Guide to the Theory of NP–Completeness.* Freeman, San Francisco, 1979.

[38] M.X. GOEMANS. Semidefinite programming in combinatorial optimization. *Math. Program.*, 79:143–162, 1997.

[39] M.X. GOEMANS. Semidefinite programming and combinatorial optimization. *Doc. Math.*, Extra Volume ICM 1998:657–666, 1998.

[40] M.X. GOEMANS and F. RENDL. Combinatorial optimization. In H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors, *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*, pages 343–360. Kluwer Academic Publishers, Boston, MA, 2000.

[41] M.X. GOEMANS and D.P. WILLIAMSON. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. Assoc. Comput. Mach.*, 42(6):1115–1145, 1995.

[42] B. GRONE, C.R. JOHNSON, E. MARQUES de SA, and H. WOLKOWICZ. Positive definite completions of partial Hermitian matrices. *Linear Algebra Appl.*, 58:109–124, 1984.

[43] P.L. HAMMER, P. HANSEN, and B. SIMEONE. Roof duality, complementation and persistency in quadratic 0-1 optimization. *Math. Program.*, 28:121–155, 1984.

[44] C. HELMBERG. *An Interior-Point Method for Semidefinite Programming and Max-Cut Bounds.* PhD thesis, Graz University of Technology, Austria, 1994.

[45] C. HELMBERG. SBmethod – A C++ implementation of the spectral bundle method. ZIB preprint 00-35, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Takustraße 7, 14195 Berlin, Germany, October 2000.

[46] C. HELMBERG and K.C. KIWIEL. A spectral bundle method with bounds. ZIB preprint SC-99-37, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Takustraße 7, 14195 Berlin, Germany, November 1999.

[47] C. HELMBERG and F. OUSTRY. Bundle methods and eigenvalue functions. In H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors, *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*, pages 307–337. Kluwer Academic Publishers, Boston, MA, 2000.

[48] C. HELMBERG, S. POLJAK, F. RENDL, and H. WOLKOWICZ. Combining semidefinite and polyhedral relaxations for integer programs. In *Integer Program-*

ming and Combinatorial Optimization (Copenhagen, 1995), pages 124–134. Springer, Berlin, 1995.

[49] C. HELMBERG and F. RENDL. A spectral bundle method for semidefinite programming. *SIAM J. Optim.*, 10(3):673 – 696, 2000.

[50] C. HELMBERG, F. RENDL, R. J. VANDERBEI, and H. WOLKOWICZ. An interior-point method for semidefinite programming. *SIAM J. Optim.*, 6(2):342–361, 1996.

[51] R.A. HORN and C.R. JOHNSON. *Matrix Analysis.* Cambridge University Press, Cambridge, 1990. Corrected reprint of the 1985 original.

[52] J. HÅSTAD. Some optimal inapproximability results. In *STOC '97 (El Paso, TX)*, pages 1–10 (electronic). ACM, New York, 1999.

[53] E. ISING. Beitrag zur Theorie des Ferromagnetismus. *Z. Phys.*, 31:253–258, 1925.

[54] C.R. JOHNSON. Matrix completion problems: a survey. *Proceedings of Symposium in Applied Mathematics*, 40:171–198, 1990.

[55] C.R. JOHNSON, B. KROSCHEL, and H. WOLKOWICZ. An interior-point method for approximate positive semidefinite completions. *Comput. Optim. Appl.*, 9(2):175–190, 1998.

[56] M. JÜNGER and F. LIERS. Private communication, 2000.

[57] M. JÜNGER and G. RINALDI. Relaxations of the max-cut problem and computation of spin glass ground states. In *Operations Research Proceedings 1997 (Jena)*, pages 74–83, Springer, Berlin, 1998.

[58] R.M. KARP. Reducibility among combinatorial problems. In R. E. Miller and J.W. Thatcher, editors, *Complexity of Computer Computation*, pages 85–103. Plenum Press, New York, 1972.

[59] M. KOJIMA and L. TUNÇEL. Cones of matrices and successive convex relaxations of nonconvex sets. *SIAM J. Optim.*, 10(3):750–778 (electronic), 2000.

[60] K.O. KORTANEK and J.S. ZHU. New purification algorithms for linear programming. *Nav. Res. Logist.*, 35(4):571–583, 1988.

[61] J.B. LASSERRE. Optimality conditions and LMI relaxations for 0-1 programs. Technical report, LAAS-CNRS, Toulouse, France, 2000.

[62] M. LAURENT. Tighter linear and semidefinite relaxations for max-cut based on the Lovász-Schrijver lift-and-project procedure. *To appear in SIAM J. Optim.*

[63] M. LAURENT. A tour d'horizon on positive semidefinite and Euclidean distance matrix completion problems. In *Topics in Semidefinite and Interior-Point Methods*, volume 18 of *The Fields Institute for Research in Mathematical Sciences, Communications Series*, pages 51–76, Providence, Rhode Island, 1998.

[64] M. LAURENT and S. POLJAK. On a positive semidefinite relaxation of the cut polytope. *Linear Algebra Appl.*, 223/224:439–461, 1995.

[65] M. LAURENT and S. POLJAK. On the facial structure of the correlation matrices. *SIAM J. Matrix Anal. Appl.*, 17(3):530–547, 1996.

[66] M. LAURENT, S. POLJAK, and F. RENDL. Connections between semidefinite relaxations of the max-cut and stable set problems. *Math. Program.*, 77:225–246, 1997.

[67] C. LEMARÉCHAL and F. OUSTRY. Semidefinite relaxations and Lagrangian duality with application to combinatorial optimization. Technical Report 3710, Institut National de Recherche en Informatique et en Automatique, INRIA, St Martin, France, 1999.

[68] T. LENGAUER. *Combinatorial Algorithms for Integrated Circuit Layout.* John Wiley & Sons Ltd., Chichester, 1990.

[69] A.S. LEWIS. Extreme points and purification algorithms in general linear programming. In *Infinite programming (Cambridge, 1984)*, pages 123–135. Springer, Berlin, 1985.

[70] F. LIERS. Private communication, 2001.

[71] L. LOVÁSZ and A. SCHRIJVER. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optim.*, 1(2):166–190, 1991.

[72] B. MOHAR and S. POLJAK. Eigenvalues in combinatorial optimization. In *Combinatorial and graph-theoretical problems in linear algebra (Minneapolis, MN, 1991)*, pages 107–151. Springer, New York, 1993.

[73] B.A. MURTAGH and M.A. SAUNDERS. Large-Scale linearly constrained optimization. *Math. Program.*, 14(1):41–72, 1978.

[74] K. NAKATA, K. FUJISAWA, M. FUKUDA, M. KOJIMA, and K. MUROTA. Exploiting sparsity in semidefinite programming via matrix completion II: implementation and numerical results. Technical Report B-368, Dept. of Information Sciences, Tokyo Institute of Technology, Tokyo, Japan, 2001.

[75] Y.E. NESTEROV and A.S. NEMIROVSKII. *Optimization over Positive Semidefinite Matrices: Mathematical Background and User's Manual.* USSR Acad. Sci. Centr. Econ. & Math. Inst., 32 Krasikova St., Moscow 117418 USSR, 1990.

[76] Yu. NESTEROV. Semidefinite relaxation and nonconvex quadratic optimization. *Optim. Methods Softw.*, 9(1-3):141–160, 1998.

[77] Yu. NESTEROV and A. NEMIROVSKII. *Interior-point polynomial algorithms in convex programming.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.

[78] Yu. NESTEROV, H. WOLKOWICZ, and Y. YE. Semidefinite programming relaxations of nonconvex quadratic optimization. In H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors, *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*, pages 361–419. Kluwer Academic Publishers, Boston, MA, 2000.

[79] P. PARDALOS and H. WOLKOWICZ, editors. *Quadratic Assignment and Related Problems.* American Mathematical Society, Providence, RI, 1994. Papers from the workshop held at Rutgers University, New Brunswick, New Jersey, May 20–21, 1993.

[80] P.A. PARRILO. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization.* PhD thesis, California Institute of Technology, 2000.

[81] S. POLJAK and F. RENDL. Node and edge relaxations for the max-cut problem. *Computing*, 52:123–127, 1994.

[82] S. POLJAK and F. RENDL. Nonpolyhedral relaxations of graph-bisection problems. *SIAM J. Optim.*, 5(3), 1995. 467-487.

[83] S. POLJAK, F. RENDL, and H. WOLKOWICZ. A recipe for semidefinite relaxation for $(0, 1)$-quadratic programming. *J. Global Optim.*, 7(1):51–73, 1995.

[84] A.J. QUIST, E. DE KLERK, C. ROOS, and T. TERLAKY. Copositive relaxation for general quadratic programming. *Optim. Methods Softw.*, 9:185–208, 1998.

[85] F. RENDL. Semidefinite programming and combinatorial optimization. *Appl. Numer. Math.*, 29:255–281, 1999.

[86] H.D. SHERALI and W.P. ADAMS. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discrete Math.*, 3(3):411–430, 1990.

[87] H.D. SHERALI and W.P. ADAMS. A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems. *Discrete Appl. Math.*, 52(1):83–106, 1994.

[88] N.Z. SHOR. Quadratic optimization problems. *Izv. Akad. Nauk SSSR Tekh. Kibern.*, 222(1):128–139, 222, 1987.

[89] C. DE SIMONE. The cut polytope and the Boolean quadric polytope. *Discrete Math.*, 79(1):71–75, 1989/90.

[90] C. DE SIMONE. A note on the Boolean quadric polytope. *Oper. Res. Lett.*, 19(3):115–116, 1996.

[91] R. STERN and H. WOLKOWICZ. Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations. *SIAM J. Optim.*, 5(2):286–313, 1995.

[92] L. TUNÇEL. Private communication, 1999.

[93] L. TUNÇEL. On the Slater condition for the SDP relaxations of nonconvex sets. Technical Report CORR 2000-13, University of Waterloo, Department of Combinatorics and Optimization, 2000.

[94] H. WOLKOWICZ, R. SAIGAL, and L. VANDENBERGHE, editors. *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*. Kluwer Academic Publishers, Boston, MA, 2000. xxvi+654 pages.

[95] V.A. YAKUBOVICH. The S-procedure and duality theorems for nonconvex problems of quadratic programming. *Vestnik Leningrad. Univ.*, 1973(1):81–87, 1973.

[96] V.A. YAKUBOVICH. Nonconvex optimization problem: the infinite-horizon linear-quadratic control problem with quadratic constraints. *Systems Control Lett.*, 19(1):13–22, 1992.

[97] Y. YE. Approximating quadratic programming with bound and quadratic constraints. *Math. Program.*, 84:219–226, 1999.

[98] Q. ZHAO. *Semidefinite Programming for Assignment and Partitioning Problems.* PhD thesis, University of Waterloo, 1996.

[99] Q. ZHAO, S.E. KARISCH, F. RENDL, and H. WOLKOWICZ. Semidefinite programming relaxations for the quadratic assignment problem. *J. Comb. Optim.*, 2(1):71–109, 1998.