Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

C. HELMBERG[1]

Semidefinite Programming for Combinatorial Optimization

[1]Konrad-Zuse-Zentrum für Informationstechnik Berlin, Takustraße 7, D-14195 Berlin, Germany, helmberg@zib.de, http://www.zib.de/helmberg

ZIB-Report 00-34 (October 2000)

Abstract. This book offers a self-contained introduction to the field of semidefinite programming, its applications in combinatorial optimization, and its computational methods.

We equip the reader with the basic results from linear algebra on positive semidefinite matrices and the cone spanned by them. Starting from linear programming, we introduce semidefinite programs and discuss the associated duality theory. We then turn to semidefinite relaxations of combinatorial optimization and illustrate their interrelation.

In the second half we deal with computational methods for solving semidefinite programs. First, the interior point approach, its iteration complexity, and implementational issues are discussed. Next, we explain in great detail the spectral bundle method, which is particularly suited for large scale semidefinite programming.

One of the most successful techniques in integer linear programming is the cutting plane approach which improves an initial relaxation by adding violated inequalities. We explore possibilities to combine the two solution methods with the cutting plane approach in order to strengthen semidefinite relaxations of combinatorial optimization problems.

Mathematics subject classification (MSC 2000). Primary 90C22; secondary 90C27, 90-02

Keywords. Positive semidefinite matrices, semidefinite cone, semidefinite programming, semidefinite duality, combinatorial optimization, max-cut, quadratic 0-1 programming, graph partitioning, semidefinite relaxations, interior point methods, spectral bundle method, eigenvalue optimization, cutting plane algorithms.

Software. An implementation of the spectral bundle method is now available at URL
`http://www.zib.de/helmberg/SBmethod`

Note. Except for a few minor corrections, this text is identical to my *Habilitationsschrift* of January 2000.

Christoph Helmberg

Semidefinite Programming for Combinatorial Optimization

Konrad-Zuse-Zentrum
für Informationstechnik Berlin
January 2000

To my family

Contents

Introduction

Combinatorial optimization is the task of finding, with respect to a given cost function, an optimal subset in a given family of subsets of a finite set. In general, the solution of this kind of problems requires complete enumeration. Typically, structural properties of the feasible set and the cost function allow to reduce the number of subsets that have to be enumerated. In this context, the technique of convex relaxation has proven highly effective. First, the combinatorial optimization problem is reformulated (if possible) as the task of optimizing a linear cost function over a finite set of integral points. Then the set of integral points is relaxed to some convex set that contains all feasible integral points and is accessible for standard optimization methods.

The technique of convex relaxation is strongly connected to linear programming. After all, the tightest convex relaxation is the convex hull of the integral points, a polyhedral set, which can be described by linear constraints. Unfortunately, for many problems of practical relevance it is computationally too expensive to provide the full description of the convex hull. None the less, numerous examples of linear relaxations of combinatorial optimization problems can be found in the literature, that lead to strong theoretical results and efficient practical solution methods. The success of linear programming goes hand in hand with the general familiarity of the scientific community with its theoretical properties and the availability of efficient solvers.

If no reasonable linear approximation of the convex hull is available, it is worth to consider other than linear relaxations of the feasible set. Several classical problems, *e.g.*, quadratic 0-1 programming, suggest the use of nonlinear methods. In general, nonlinear relaxations are considered much harder to analyze and implement. In the last two decades, new results and algorithms for convex optimization have led to a better understanding and better solution methods for nonlinear optimization problems. In particular, the field of semidefinite programming has experienced (and is still experiencing) considerable progress. Within the last few years it has become a standard tool in optimization that is sufficiently easy to use. Its most important applications are found within control theory, signal processing, eigenvalue optimization, and combinatorial optimization.

In comparison to linear programming, semidefinite programming offers the additional possibility to work with positive semidefinite matrix variables. A linear cost function is optimized over the cone S_n^+ of positive semidefinite matrices (instead of the cone $\mathbb{R}_+^n$ of nonnegative variables) subject to linear constraints. Since the semidefinite cone is not polyhedral, duality theory requires more care than in linear programming. In order to guarantee strong duality, one usually assumes the Slater condition to hold, *i.e.*, the existence of a strictly feasible primal or dual solution. Given this, semidefinite programs can be solved in a routine manner by interior point methods.

In the seminal paper "On the Shannon Capacity of a Graph" Lovász [1979] introduced a surprising bound on the independence number of graphs. He gave several equivalent formulations of this bound. They involved the minimization of the maximal eigenvalue over an affine set of matrices or linear programs with positive semidefiniteness constraints. Grötschel, Lovász, and Schrijver [1981] proved that, in fact, the bound can be computed in polynomial time by means of the ellipsoid method. Unfortunately, the ellipsoid method has not been successful in implementations, so the result was mainly of theoretic interest at the time. With his nonlinear polynomial time algorithm for linear programming, Karmarkar [1984] initiated the development of interior point methods with polynomial iteration complexity for more general convex problems. To the best of our knowledge the first for semidefinite programming was proposed by Nesterov and Nemirovskii [1994]. Overton and Womersly [1993] clarified the connection between certain eigenvalue opti-

mization problems and semidefinite programs, a connection that was in part anticipated already in the fifties (cf. Overton and Womersly [1992]). Because of this connection the first semidefinite programming bounds for combinatorial optimization may arguably be attributed to Donath and Hoffman [1972]; Donath and Hoffman [1973]; they employed eigenvalue optimization in order to obtain lower bounds for graph partitioning. Alizadeh [1995], who was among the first to develop semidefinite interior point methods with polynomial iteration complexity, also presented a survey on combinatorial applications of semidefinite programming. With his work semidefinite relaxations were considered to be of potential practical value. Recent progress in approximation algorithms, initiated by the paper of Goemans and Williamson [1995], confirmed the importance of semidefinite programming for combinatorial optimization.

In spite of these promising developments, only few people have embarked on actually employing semidefinite programming as a tool for attacking hard combinatorial optimization problems. The reasons for this moderate interest might be twofold. Even though semidefinite programming is currently a "buzz-word", there is a lack of introductory texts that lead into the field. In consequence, semidefinite programming remained somewhat peculiar for many people that have grown accustomed to linear programming. Second, the development of semidefinite programming algorithms and codes is still in its beginning. In particular, it seems to be significantly more difficult to exploit structural properties of the underlying data. This has led many people to believe that semidefinite programming is applicable to small academic problems only.

In this text we address both topics. First, we hope to provide an introduction to the field that is sufficiently easy to read for anybody with some basic knowledge about linear programming and convexity, yet complete enough to establish a firm ground for subsequent deeper studies. Second, we describe in detail two semidefinite programming algorithms, the primal-dual interior point method and the spectral bundle method, which we believe to be especially useful for combinatorial applications. For these two methods we discuss various possibilities, as well as advantages and disadvantages of combining them with a cutting plane approach. We hope to convince the reader, that in particular the spectral bundle method is well suited for large scale optimization and may indeed be a helpful tool for several difficult combinatorial optimization problems.

We give a short outline of the contents. Chapter 1 reviews some important results from linear algebra and basic properties of the cone of symmetric positive semidefinite matrices. Chapter 2 is devoted to the theory of semidefinite programming, including standard formulations, duality, and a short survey on complexity results. In Chapter 3 we discuss semidefinite relaxations of several combinatorial optimization problems and their interrelation. The interior point approach to semidefinite programming in general and a particular primal-dual interior point method are the topic of Chapter 4. In Chapter 5 we explain the spectral bundle method, its connection to semidefinite programming, and its extension to bounded variables. Finally, we present our experience with cutting plane approaches in semidefinite programming in Chapter 6. In this last chapter we also provide some numerical results illustrating the quality of the relaxations obtained and describe an approach for fixing variables whose bounds are implied by the semidefiniteness constraint.

For the convenience of the reader, basic terminology and facts about linear algebra and convexity are listed in the appendices.

We would like to emphasize that this text is not a comprehensive study of semidefinite programming and its algorithmic approaches, nor does it provide a complete listing of combinatorial applications in semidefinite programming. Indeed, we have confined the material to aspects that we believe are indispensable in order to understand and work with semidefinite relaxations of combinatorial optimization problems. We try to compensate for the omission of further relevant material by including, at the end of each chapter, an extra section "Remarks on the Literature" that refers to the sources of the presented material and provides pointers to further reading on related topics. Many of the articles cited may still be available on the world wide web, see `http://www.zib.de/helmberg/semidef.html`. For a comprehensive treatment of all aspects of semidefinite programming we recommend the "Handbook of Semidefinite Programming" by Wolkowicz, Saigal, and Vandenberghe [2000], which will be published soon.

Acknowledgments. I thank Andreas Eisenblätter, Alexander Martin, and Robert Weisman-

tel for reading parts of this text and for their thoughtful suggestions that helped to improve the presentation. I thank Martin Grötschel for his challenging, yet supporting scepticism on my attempts to render semidefinite programming applicable to real world combinatorial optimization problems, for his helpful criticism on an earlier version of the text, and for providing an exceptional working environment at the Konrad-Zuse-Zentrum für Informationstechnik Berlin.

Berlin, January 2000 Christoph Helmberg

0.1 Personal Contributions to the Field

This text is my *Habilitationsschrift*, but in the hope to open it to a wider audience it is organized as a self contained introduction to the field. In view of the latter goal it is unavoidable that the major part of the contents refers to results that are not part of my work and it would not be appropriate to bore my audience by constantly explaining my contributions inside the text. Therefore I will try to point out my contributions in this separate section, so that they can be located easily within the text.

Chapter 1 and Chapter 2 contain no noteworthy personal contributions. **Chapter 1** provides the basic tools from linear algebra. These are rarely used by people working in combinatorial optimization and so a detailed treatment seemed advisable. **Chapter 2** leads into semidefinite programming and its duality theory.

In **Chapter 3** the personal contributions are the representation of the isomorphism between quadratic $\{-1,1\}$- and 0-1 formulations as a scaling (Section 3.2; Helmberg [2000]) and the work on the quadratic knapsack problem (Section 3.3.2; Helmberg, Rendl, and Weismantel [2000]).

The isomorphism between the max-cut problem and quadratic 0-1 programming has been known for a long time (Hammer [1965]; De Simone [1989]), the equivalence of the semidefinite relaxation is first noted in Helmberg, Poljak, Rendl, and Wolkowicz [1995]; Laurent, Poljak, and Rendl [1997]. The advantage of the scaling transformation given in Helmberg [2000] is that this representation also allows to map the constraints so that the dual variables and structural properties of the constraints are preserved, in particular, sparsity (sparsity was also achieved by the previous transformation) and low rank structure. The preservation of the dual variables is of importance, because certain manipulations are easier in the dual of the max-cut relaxation than in the dual of the quadratic 0-1 programming relaxation. In particular the routine for fixing variables of Section 6.3.2 exploits the constraint structure of the max-cut relaxation. With the scaling transformation it is easy to employ this routine to optimal solutions computed for appropriate formulations of the quadratic 0-1 programming relaxation. In connection with the conic representation $\mathcal{T}$ (see Lemma 3.1.7) of the approximation result of Goemans and Williamson [1995] the transformation easily yields the equivalent result for the quadratic 0-1 programming setting (see Corollary 3.2.6).

In the case of the quadratic 0-1 knapsack problem the intention of Helmberg, Rendl, and Weismantel [2000] was to devise a good representation of a linear inequality by just one constraint in the semidefinite relaxation. The motivation came from the fact that the number of constraints in a semidefinite program has a significant influence on the performance of interior point methods. The general approach of Lovász and Schrijver [1991] would require n inequalities to represent one linear inequality. For a constrained quadratic 0-1 programming problem with many linear constraints, applying the Lovász-Schrijver approach may be computationally too expensive. The square representation ((SQK2) in Lemma 3.3.3) proved not only to be better than the diagonal representation (SQK1) in theory, but also to be of good quality in comparison to the Lovász-Schrijver relaxation (SQK4) in practice. Recently, Bauvin and Goemans informed me, that we had overlooked an intermediate representation that was used in our original proof of Lemma 3.3.3 (it needs a moment of thought, though, therefore we attribute it to Bauvin and Goemans), this is now (SQK3) in Lemma 3.3.3. The semidefinite relaxation for linear cost functions (Lemma 3.3.5) has certainly little practical relevance, because the semidefinite approach is not competitive, but it is instructive to see that the representation gains something even in this simple case. For quadratic cost functions, however, the difference between diagonal representation (SQK1) and (SQK2)/(SQK3) may be significant, see the numerical results of Section 6.1.2. We consider this an important first step towards a general cutting plane routine for constrained quadratic 0-1 programming.

In **Chapter 4** the main contribution has already been part of my PhD-thesis Helmberg [1994]. It is the development of the first primal-dual interior point algorithm for semidefinite programming by Helmberg, Rendl, Vanderbei, and Wolkowicz [1996]. Independently the same algorithm was developed by Kojima, Shindoh, and Hara [1997]. Whereas we presented good numerical results but could only prove convergence of the method (see also Helmberg [1994]), they provided a proof

of polynomial convergence but had not yet implemented the algorithm. One year later Monteiro [1997] rediscovered the same search direction. It is now often referred to as the HRVW/KSH/M or HKM direction. In the text, the direction is listed under (4.6). It is the basis of the algorithm described in Sections 4.2 and 4.4. The search direction still yields one of the fastest methods in terms of CPU time, in particular for combinatorial applications. Other than through the development of the search direction the theoretic results of Chapter 4 are not influenced by my work. In an introduction to the field, however, these results are indispensable and so I tried to present them in a unified framework that may differ considerably from the original works. Section 4.4 is based on personal experience with implementing the method, but several recommendations are still open to discussion.

The spectral bundle method of **Chapter 5** forms the core of the *Habilitationsschrift*. The chapter is based on Helmberg and Rendl [2000], Helmberg and Kiwiel [1999], but includes some new material as well. The motivation for this work was to devise an algorithm that can compute, within reasonable time, approximate solutions of well structured semidefinite programs with large matrix variables and many constraints.

Helmberg and Rendl [2000] employ, to the best of our knowledge, for the first time a nonpolyhedral model in bundle methods (Section 5.2). The concept of combining semidefinite interior point methods and nonsmooth optimization techniques in order to derive an efficient algorithm for solving semidefinite programs is also new. The structure of the semidefinite cutting plane model is unusual in that not the subgradients are stored themselves but rather a set for generating them. On the one hand this makes it possible to use the bundle approach also for cutting plane methods, see Chapter 6. On the other hand, this opened new possibilities for aggregation in combination with the interior point algorithms. For linear cutting planes, aggregation consisted of linear combinations of cutting planes; here, aggregation is achieved in part by identifying and maintaining an appropriate nonpolyhedral face of the semidefinite cone.

Helmberg and Kiwiel [1999] propose a new technique for including box constraints (Section 5.4) that is particularly easy to implement and very efficient in practice. This allows the use of inequality constraints in the associated primal semidefinite program at almost the same cost as equality constraints. The inexact evaluation approach of Section 5.6 reduces the time spent in the eigenvalue computation significantly. Kiwiel [1995] already suggested inexact function evaluation and subgradient computations; The inexact evaluation criterion used in Helmberg and Kiwiel [1999] differs slightly from Kiwiel [1995] and exploits the special structure of the problem. The proofs of convergence follow the framework of Kiwiel [1990], but several subtle changes had to be made in order to cover bounds and inexact evaluation.

The proof of Theorem 5.3.8, establishing convergence of the optimal solutions of the subproblems to a solution of the associated primal problem, is new. The result is not very surprising, but of fundamental importance for cutting plane methods, because it justifies the use of the solutions of the subproblems for separation. The inexact stopping criterion for the quadratic semidefinite subproblem of Lemma 5.5.3 is published here for the first time, fruitful discussions with K. C. Kiwiel are gratefully acknowledged. It ensures finite convergence of the interior point method and sufficient relative precision of the respective solution. The associated Lemma 5.5.4 helps to establish convergence of the entire algorithm (Theorem 5.5.5) for inexact model evaluation.

Chapter 6 on cutting plane algorithms in semidefinite programming is based on the implementations of Helmberg [1994]; Helmberg and Rendl [1998]; Helmberg, Rendl, and Weismantel [2000]; Helmberg [2000] and new work on the spectral bundle method. The importance of implementational contributions is difficult to judge because of their heuristic nature. I believe that the early separation and restarting technique developed for max-cut in my PhD-thesis Helmberg [1994] does make good use of the possibilities offered by primal-dual interior point methods. At first it was a surprise that the approach did not work at all for the quadratic 0-1 knapsack problem (Helmberg and Weismantel [1998]), so I try to highlight this problem dependence. Helmberg, Rendl, and Weismantel [2000] introduce a new class of valid inequalities for the quadratic knapsack polytope (see Lemma 6.1.5), but the proof does not match the topic of this text. On the implementational side, the development of separation routines for quadratic knapsack cuts seems important, but this has not yet led to direct use in real world applications.

The cutting plane approach employing the spectral bundle method (Section 6.2) is new but still very much in its beginnings. Usually, bundle methods are not well suited for solving the dual to cutting plane methods, because the constant change in dimension destroys the collected subgradients. This can be avoided in the spectral bundle method if appropriate aggregation information is stored, because the subgradients are maintained and generated via an independent semidefinite set. The ideas presented should make it possible to attack larger problems. We try to illustrate the potential of this approach by including preliminary computational results for max-cut and max k-cut. In particular, the results on max k-cut yield, to the best of our knowledge, the first significant lower bounds on co-channel interference in some real world frequency assignment models. Attempts to provide such bounds by pure linear cutting plane approaches failed so far (A. Eisenblätter [personal communication]).

Finally, Helmberg [2000] proposes a method for fixing variables in semidefinite relaxations (Section 6.3). This is a basic technique needed in branch and cut approaches. We investigate the case that the bounds on the variables are implied by the semidefiniteness constraint; thus, the corresponding duality information has to be extracted from the dual semidefinite slack matrix. In order to do this, we develop a more general approach for fixing constraints. To obtain an implementable algorithm we have to exploit the special structure of semidefinite programs that are equivalent to eigenvalue optimization problems. This is, *e.g.*, the case for max-cut and extends, via the scaling of Section 3.2, to constrained quadratic 0-1 programming.

Chapter 1

Basics from Linear Algebra

In this chapter we recapitulate some basic facts about semidefinite matrices and matrix calculus that are fundamental for duality and algorithmic approaches in semidefinite programming. In addition we give some insight into the geometry of the cone of positive semidefinite matrices. The reader somewhat acquainted with the field may safely skip this chapter and return for selected topics upon need.

Section 1.1 introduces some notation and lists well known results on symmetric and positive semidefinite matrices. In Section 1.2 we discuss properties of the cone of positive semidefinite matrices and look at its facial structure. Section 1.3 is devoted to the Kronecker product. This will turn out to be useful when working with matrix equations and derivatives of matrix functions. Finally, Section 1.4 mentions some basic facts from matrix calculus that are of importance for interior point algorithms.

1.1 Symmetric and Positive Semidefinite Matrices

The set $M_{m,n}$ of $m \times n$ real matrices can be interpreted as a vector space in $\mathbb{R}^{n \cdot m}$. In this vector space the natural inner product between two elements $A, B \in M_{m,n}$ is

$$\langle A, B \rangle = \operatorname{tr}(B^T A) = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} b_{ij}. \tag{1.1}$$

The *trace* $\operatorname{tr}(\cdot)$ is the sum of the diagonal elements of a square matrix. For brevity, the set of square matrices of dimension n is denoted by M_n. It is an important fact that the trace of $A \in M_n$ is equal to the sum of the eigenvalues of A. The trace is a linear function. If the argument of the trace is a product of matrices the matrices may be rotated without affecting the result,

$$\langle AB, C \rangle = \operatorname{tr}(C^T AB) = \operatorname{tr}(BC^T A) = \left\langle A, CB^T \right\rangle.$$

The norm associated with this inner product is the Frobenius norm,

$$\|A\|_{\mathrm{F}} = \sqrt{\langle A, A \rangle}.$$

Instead of $M_{m,n}$ we will usually work with the set of symmetric matrices S_n which is a vector space in $\mathbb{R}^{\binom{n+1}{2}}$. All results on the positive semidefinite cone will be stated with respect to this $\binom{n+1}{2}$-dimensional space. It is convenient to use the inner product of $M_{m,n}$ for S_n, as well. For $A, B \in S_n$,

$$\langle A, B \rangle = \operatorname{tr}(B^T A) = \operatorname{tr}(AB).$$

A matrix A is called *skew-symmetric* if it satisfies $A = -A^T$. Obviously, the diagonal of a skew-symmetric matrix is zero. The dimension of the vector space of skew-symmetric matrices is $\binom{n}{2}$.

The linear subspaces of symmetric and skew-symmetric matrices are orthogonal in M_n. For $A \in S_n$ and a skew-symmetric matrix $B \in M_n$ we have

$$\langle A, B \rangle = \mathrm{tr}(B^T A) = \sum_{i<j} a_{ij} b_{ij} + \sum_{i>j} a_{ji}(-b_{ji}) = 0. \tag{1.2}$$

Symmetric and skew-symmetric matrices together span M_n. Any matrix $A \in M_n$ can be decomposed into its symmetric part $(A + A^T)/2$ and skew-symmetric part $(A - A^T)/2$,

$$A = \frac{A + A^T}{2} + \frac{A - A^T}{2}.$$

All eigenvalues of a symmetric matrix $A \in S_n$ are real and there is an orthonormal matrix $P \in M_n$ which diagonalizes A, $P^T A P = \Lambda_A$ (see Theorem A.0.1). Λ_A is a diagonal matrix with the eigenvalues of A on its main diagonal. Since the eigenvalues are the solutions of the characteristic polynomial $\det(A - \lambda I)$, they depend continuously on the matrix elements. We denote the eigenvalues of A by $\lambda_i(A)$, $i = 1, \ldots, n$. For our purposes it is convenient to sort the eigenvalues non-increasingly, $\lambda_{\max}(A) = \lambda_1(A) \geq \lambda_2(A) \geq \ldots \geq \lambda_n(A) = \lambda_{\min}(A)$. We will usually drop the zero eigenvalues. For a matrix $A \in S_n$ with $\mathrm{rank}(A) = k$ the eigenvalue decomposition is given by a diagonal matrix $\Lambda_A \in S_k$ and a matrix $P \in M_{n,k}$ with $P^T P = I_k$ so that $A = P \Lambda_A P^T$.

Observation 1.1.1 *Let $A \in S_n$. Since $\lambda_{\max}(A) = \max_{v \in \mathbb{R}^n, \|v\|=1} v^T A v$ by the Rayleigh-Ritz theorem A.0.4, it follows that $\lambda_{\max}(A) \geq \max\{a_{ii} : i \in \{1, \ldots, n\}\}$.*

We now turn to positive semidefinite matrices. Although it is possible to define this term for arbitrary square matrices we will use it exclusively in connection with symmetric matrices.

Definition 1.1.2
$A \in S_n$ *is* positive semidefinite *($A \in S_n^+$, $A \succeq 0$) if $x^T A x \geq 0 \quad \forall x \in \mathbb{R}^n$.*
$A \in S_n$ *is* positive definite *($A \in S_n^{++}$, $A \succ 0$) if $x^T A x > 0 \quad \forall x \in \mathbb{R}^n \setminus \{0\}$.*

We state some immediate consequences of this definition that will be useful in the following.

Observation 1.1.3 *Any principal submatrix of a positive definite/semidefinite matrix is again positive definite/semidefinite. In particular, all diagonal elements of a positive definite matrix must be positive.*

Observation 1.1.4 *Let $A_i \in S_{n_i}$ for $i = \{1, \ldots, n\}$. The symmetric block diagonal matrix*

$$A = \begin{bmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & A_m \end{bmatrix}$$

is positive definite/semidefinite if and only if all of the A_i are so.

Observation 1.1.5 *For $A \in S_n^+$ there is always a diagonal element a_{ii} among the elements of largest absolute value,* i.e., *$\exists i \in \{1, \ldots, n\} : a_{ii} = \max\{|a_{ij}| : i, j \in \{1, \ldots, n\}\}$.*

Observation 1.1.6 *If $A \in S_n^+$ and $a_{ii} = 0$ for some $i \in \{1, \ldots, n\}$ then $a_{ij} = 0$ for all $j \in \{1, \ldots, n\}$.*

The next proposition states that positive semidefiniteness is invariant under basis transformations.

Proposition 1.1.7 *Let $B \in M_n$ be a nonsingular matrix. Then $A \in S_n^+$ if and only if $B^T A B \in S_n^+$ and $A \in S_n^{++}$ if and only if $B^T A B \in S_n^{++}$.*

Proof. For $x \in \mathbb{R}^n$ and $y = B^{-1} x$ we obtain $x^T A x = x^T B^{-T} B^T A B B^{-1} x = y^T B^T A B y$. ■

There are several equivalent characterizations for positive definite matrices.

Theorem 1.1.8 *(Characterizations of positive definite matrices)*
For $A \in S_n$ the following statements are equivalent:

1. *A is positive definite.*
2. *$\lambda_i(A) > 0$ for $i = 1, \ldots, n$.*
3. *$\exists C \in M_n$ with $\operatorname{rank}(C) = n$ such that $A = C^T C$.*
4. *For an arbitrary nested sequence[1] $A_i \in S_i$, $i = 1, \ldots, n$, of principal submatrices of A: $\det(A_i) > 0$ for $i = 1, \ldots, n$.*

Proof. We show (1) $\Longrightarrow$ (2) $\Longrightarrow$ (3) $\Longrightarrow$ (1) and (2) $\Longleftrightarrow$ (4).
(1) $\Longrightarrow$ (2). Let $v \in \mathbb{R}^n$ be an eigenvector of norm 1 to the eigenvalue λ of A. Then, by the positive definiteness of A, $0 < v^T A v = \lambda v^T v = \lambda$.
(2) $\Longrightarrow$ (3). Let $A = P \Lambda P^T$ be the eigenvalue decomposition of A and denote by $\Lambda^{\frac{1}{2}}$ the diagonal matrix whose elements are the square roots of the elements of Λ. Then $C = \Lambda^{\frac{1}{2}} P^T$ is a matrix as required in (3).
(3) $\Longrightarrow$ (1). For arbitrary $v \in \mathbb{R}^n \setminus \{0\}$ let $w = Cv$. Then $w \neq 0$ and $v^T A v = v^T C^T C v = w^T w > 0$.
(2) $\Longrightarrow$ (4). The determinant is the product of the eigenvalues and any principal submatrix is positive definite by Observation 1.1.3. Together with (2) $\Longleftrightarrow$ (1) this implies that the determinant of any principal submatrix is positive.
(4) $\Longrightarrow$ (2). We proceed inductively. Clearly, $A_1 \in \mathbb{R}$ has exactly one positive eigenvalue, because $\det(A_1) > 0$. In extending A_i to A_{i+1} we make use of the interlacing theorem for bordered matrices A.0.5. By this theorem, the eigenvalues of A_i interlace with the eigenvalues of A_{i+1}. Since all of the eigenvalues of A_i are positive, at most one eigenvalue of A_{i+1} could be nonpositive. But this is impossible, because then $det(A_{i+1})$ would be nonpositive as well. ■

A is positive definite if and only if the inverse A^{-1} is positive definite, because the eigenvalues of A^{-1} are $1/\lambda_i(A)$.

The following theorem gives a characterization of the positive semidefiniteness of a matrix via the positive semidefiniteness of the so-called Schur complement with respect to a block partitioning of the matrix.

Theorem 1.1.9 *(Schur complement) Let $A \in S_m^{++}$, $C \in S_n$, and $B \in M_{m,n}$. Then*

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \succ 0 \qquad \Longleftrightarrow \qquad C - B^T A^{-1} B \succ 0$$

and

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \succeq 0 \qquad \Longleftrightarrow \qquad C - B^T A^{-1} B \succeq 0.$$

Proof. Since A is positive definite we may define the following nonsingular matrix,

$$X = \begin{bmatrix} I_m & -A^{-1}B \\ 0 & I_n \end{bmatrix}.$$

The theorem now follows from

$$X^T \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} X = \begin{bmatrix} A & 0 \\ 0 & C - B^T A^{-1} B \end{bmatrix},$$

Proposition 1.1.7, and Observation 1.1.4. ■

[1] A nested sequence is determined by a sequence of proper subsets $J_1 \subset J_2 \subset \ldots \subset J_n = \{1, \ldots, n\}$ of indices with $|J_i| = i$ for $i = 1, \ldots, n$.

This theorem may be used to construct an algorithm for recognizing and factorizing positive definite matrices. Partition a given matrix $A_n \in S_n$ into blocks $a_n \in \mathbb{R}$, $b_n \in \mathbb{R}^{n-1}$, and $C \in S_{n-1}$. Then

$$A_n = \begin{bmatrix} a_n & b_n^T \\ b_n & C_n \end{bmatrix} \succ 0 \quad \Longleftrightarrow \quad a_n > 0 \ \text{ and } \ A_{n-1} = C_n - \frac{1}{a_n} b_n b_n^T \succ 0. \tag{1.3}$$

The positiveness of a_n is easy to check and for checking $A_{n-1} \succ 0$ we can apply the algorithm recursively until it remains to test $A_1 = a_1 \in \mathbb{R}$ for positiveness. The vectors

$$v_i = \begin{bmatrix} \sqrt{a_i} \\ \frac{1}{\sqrt{a_i}} b_i \end{bmatrix} \quad \text{for } i = n, \ldots, 1$$

give rise to a factorization via the relation

$$\begin{aligned} A_n &= v_n v_n^T + \begin{bmatrix} 0 & 0 \\ 0 & A_{n-1} \end{bmatrix} \\ &= \begin{bmatrix} v_n \begin{bmatrix} 0 \\ v_{n-1} \end{bmatrix} \end{bmatrix} \begin{bmatrix} v_n^T \\ \begin{bmatrix} 0 & v_{n-1}^T \end{bmatrix} \end{bmatrix} + \begin{bmatrix} 0_{2\times 2} & 0_{2\times(n-2)} \\ 0_{(n-2)\times 2} & A_{n-2} \end{bmatrix} = \ldots \end{aligned} \tag{1.4}$$

Continuing this process we obtain a lower triangular matrix L satisfying $A = LL^T$. This is called the *Cholesky factorization*.

Theorem 1.1.10 ***(Cholesky factorization)*** *For $A \succ 0$ there is a unique lower triangular matrix L such that $A = LL^T$. Furthermore, L can be constructed via* (1.3) *and* (1.4).

Proof. Existence is clear from (1.3) and (1.4). Since L is lower triangular, element $a_{11} = a_n$ determines l_{11} uniquely, which again determines the first column of L uniquely. By (1.3) and (1.4) this argument can now be applied recursively to prove uniqueness of L itself. ■

We continue with a sufficient condition for positive definiteness.

Definition 1.1.11
A matrix $A \in M_n$ is strictly diagonally dominant *if $|a_{ii}| > \sum_{\substack{j=1 \\ i \neq j}}^{n} |a_{ij}|$ for all $i = 1, \ldots, n$.*

Theorem 1.1.12 *If $A \in S_n$ is strictly diagonally dominant and if all diagonal elements are positive then A is positive definite.*

Proof. From the Geršgorin disc theorem A.0.6 and the definition of diagonally dominant matrices it follows that all Geršgorin discs of A lie in the positive halfspace, and so all eigenvalues are positive. By Theorem 1.1.8 the matrix A is positive definite. ■

Diagonal dominance is easy to check and it often comes in handy when constructing initial feasible solutions for semidefinite programs.

For positive semidefinite matrices almost the same characterizations as in Theorem 1.1.8 are valid.

Theorem 1.1.13 *(Characterizations of positive semidefinite matrices)*
For $A \in S_n$ the following statements are equivalent:

1. *A is positive semidefinite.*
2. *$\lambda_i(A) \geq 0$ for $i = 1, \ldots, n$.*
3. *$\exists C \in M_{m,n}$ such that $A = C^T C$. For any such C,* $\operatorname{rank}(C) = \operatorname{rank}(A)$.

Proof. Analogous to the proof of Theorem 1.1.8. ■

For a positive semidefinite matrix A the determinant of any principal submatrix of A is certainly non-negative. However, the existence of a nested sequence of non-negative principal minors is no longer sufficient for positive semidefiniteness. On the other hand, it is again possible to test algorithmically, whether a matrix $A_n \in S_n$ is positive semidefinite. Like in (1.3) we may use Observation 1.1.6 and the Schur complement theorem 1.1.9 to set up a recursive test,

$$A_n = \begin{bmatrix} a_n & b_n^T \\ b_n & C_n \end{bmatrix} \succeq 0 \iff \begin{cases} \text{either} & a_n > 0 \text{ and } A_{n-1} = C_n - \frac{1}{a_n} b_n b_n^T \succeq 0 \\ \text{or} & a_n = 0,\ b_n = 0 \text{ and } A_{n-1} = C_n \succeq 0. \end{cases} \tag{1.5}$$

The following theorem is now easy to prove.

Theorem 1.1.14 *The positive semidefiniteness of a matrix can be checked in $n^3/3 + O(n^2)$ arithmetic operations.*

Remark 1.1.15 *In fact, for a matrix with rational entries, the length of the binary encoding can be shown to remain polynomially bounded in the input length of the matrix throughout the algorithm (see Grötschel, Lovász, and Schrijver [1988], (9.3.30), Claim 4.)*

The fact that a positive semidefinite matrix A can equivalently be written in the form $C^T C$ is very important and will be exploited in many proofs. A very intriguing interpretation of a factorization is to view the columns of C as vectors v_i. Element a_{ij} is the scalar product $\langle v_i, v_j \rangle$ of the vectors v_i and v_j. The matrix A is referred to as the *Gram matrix* of the vectors $v_1, \ldots, v_n$. The factorization is not unique and there are several algorithmic possibilities to construct one. Examples are Cholesky decomposition of Theorem 1.1.10 and eigenvalue decomposition as in the proof of Theorem 1.1.8.

Eigenvalue decomposition is also used to prove the following theorem.

Theorem 1.1.16 *(root of a positive semidefinite matrix)*
Let $A \in S_n$ be positive semidefinite and $k \geq 1$ integral. Then there exists a unique positive semidefinite matrix B with $B^k = A$. Furthermore, this matrix B satisfies $AB = BA$ and $\operatorname{rank}(A) = \operatorname{rank}(B)$.

Proof. Such a matrix B exists because for an eigenvalue decomposition $P \Lambda_A P^T$ of A the matrix $B = P \sqrt[k]{\Lambda_A} P^T$ fulfills the requirements. We show uniqueness. Since B^k commutes with A, i.e., $B^k A = A B^k$, Theorem A.0.3 implies that there is an orthonormal matrix $P \in M_n$ that diagonalizes B^k and A. Denote the diagonal matrix by $\Lambda_A = P^T A P = P^T B^k P$. Any eigenvector of B to an eigenvalue λ_B of B is an eigenvector of B^k to the eigenvalue λ_B^k, so P diagonalizes B as well, $\Lambda_B = P^T B P$. Therefore $B^k = P \Lambda_B^k P^T = P \Lambda_A P^T$ and the spectral information determines the matrix entries uniquely. ■

We denote the k-th root of A by $A^{\frac{1}{k}}$ and, if A is positive definite, the k-th root of A^{-1} by $A^{-\frac{1}{k}}$.

1.2 The Cone of Semidefinite Matrices

We will now take a close look at the set of positive semidefinite matrices, interpreted as a subset of S_n. Some fundamental concepts from convexity will be needed. For the convenience of the reader we have collected basic terminology and results in Appendix B.

Definition 1.2.1 *A set $C \subseteq \mathbb{R}^n$ is a* cone *if it is closed under nonnegative multiplication and addition ($x, y \in C \Longrightarrow \lambda(x+y) \in C \ \ \forall \lambda \geq 0$). A cone C is* pointed *if $C \cap (-C) = \{0\}$.*

Note, that this definition implies that a cone is a convex set.

Proposition 1.2.2 *S_n^+ is a full dimensional, closed pointed (convex) cone in $\mathbb{R}^{\binom{n+1}{2}}$.*

Proof. For $A, B \in S_n^+$ it is not difficult to check that $\lambda(A+B) \in S_n^+$ for all $\lambda \geq 0$ by making use of the definition of positive semidefiniteness. Closedness is a consequence of the continuity of the eigenvalues and Theorem 1.1.13. The cone is full dimensional, because sufficiently small perturbations of the identity I in any direction $A \in S_n$ yield a positive definite matrix, $I + A/(2\max\{|\lambda_{\max}(A)|, |\lambda_{\min}(A)|\}) \succ 0$. ■

The set of positive definite matrices S_n^{++} is not a cone because $0 \notin S_n^{++}$. It is easy to see that S_n^{++} is the interior of the cone S_n^+ and that the boundary of S_n^+ consists of the positive semidefinite matrices having at least one zero eigenvalue.

The next lemma proves that, at its apex, the semidefinite cone opens with an angle of $\frac{\pi}{2}$.

Lemma 1.2.3 *Let $A, B \in S_n^+$. Then $\langle A, B\rangle \geq 0$ and $\langle A, B\rangle = 0$ if and only if $AB = 0$.*

Proof. Let $A, B \in S_n^+$ and $k = \operatorname{rank}(A)$. Let the eigenvalue decomposition of A be given by $A = P\Lambda_A P^T$ with $\Lambda_A \in S_k^{++}$, $P \in M_{n,k}$, $P^T P = I_k$. Then

$$\langle A, B\rangle = \operatorname{tr}(P\Lambda_A P^T B) = \operatorname{tr}(\Lambda_A P^T B P) = \sum_{i=1}^{k} \lambda_i(A) \cdot P_{\cdot,i}^T B P_{\cdot,i} \geq 0.$$

As B is positive semidefinite, $P_{\cdot,i}^T B P_{\cdot,i}$ is nonnegative for $i = 1, \ldots, k$. $\langle A, B\rangle = 0$ implies that eigenvectors corresponding to positive eigenvalues of A belong to the null space of B and thus $AB = 0$. ■

We will need the following simple bounds on the inner product of semidefinite matrices when proving the boundedness of sequences of semidefinite matrices.

Lemma 1.2.4 *Let $A, B \in S_n^+$. Then $\langle A, B\rangle$ is bounded by*

$$\lambda_{\min}(A)\lambda_{\max}(B) \leq \lambda_{\min}(A)\operatorname{tr}(B) \leq \langle A, B\rangle \leq \lambda_{\max}(A)\operatorname{tr}(B) \leq n\lambda_{\max}(A)\lambda_{\max}(B).$$

Proof. We only prove the two left hand side inequalities. The proof of the right hand side inequalities is left to the reader. Let $P \in M_n$ be an orthonormal matrix such that $A = P\Lambda_A P^T$. Then

$$\begin{aligned} \langle A, B\rangle &= \operatorname{tr}(AB) = \operatorname{tr}(P\Lambda_A P^T B) = \operatorname{tr}(\Lambda_A P^T B P) \\ &\geq \lambda_{\min}(A)\operatorname{tr}(P^T B P) = \lambda_{\min}(A)\operatorname{tr}(B) \\ &\geq \lambda_{\min}(A)\lambda_{\max}(B). \end{aligned}$$

■

We will now prove a key property of the cone of semidefinite matrices, its self-duality. We start with a definition.

Definition 1.2.5 *For a cone C the* polar cone C^* *is the set $\{y : \langle x, y\rangle \geq 0$ for all $x \in C\}$.*

For a cone C the polar cone C^* can be seen as the set of tight valid linear inequalities for C or, equivalently, as the set of tangent planes to C. It is therefore natural to speak of C^* as the *dual cone* to C. Cones that satisfy $C = C^*$ are called *self-polar* or *self-dual*.

Lemma 1.2.6 $S_n^+ = S_n^{+*}$.

Proof. $S_n^+ \subseteq S_n^{+*}$ follows from Lemma 1.2.3. To show $S_n^{+*} \subseteq S_n^+$ note that for all $x \in \mathbb{R}^n$ the matrix xx^T is positive semidefinite. For $A \in S_n^{+*}$ we have

$$0 \leq \left\langle A, xx^T \right\rangle = x^T A x \qquad \forall x \in \mathbb{R}^n$$

and therefore $A \in S_n^+$. ■

Lemma 1.2.6 is equivalent to Fejer's Trace Theorem, which we formulate as a Corollary.

Corollary 1.2.7 *(Fejer's Trace Theorem)*
A is positive semidefinite if and only if $\langle A, B\rangle \geq 0$ for all $B \in S_n^+$.

What are the faces of the cone of positive semidefinite matrices? We first repeat the definition of a face of a convex set.

Definition 1.2.8 *A convex set $F \subseteq C$ is called a* face *of a convex set C if for any two elements $x, y \in C$ with $\alpha x + (1-\alpha)y \in F$ for some $\alpha \in (0,1)$ we have $x, y \in F$.*

The faces of the positive semidefinite cone can be characterized as the cones of semidefinite matrices with the property that the eigenvectors to non-zero eigenvalues are restricted to some subspace of $\mathbb{R}^n$.

Theorem 1.2.9 *(Barker and Carlson [1975]) The faces of S_n^+ are*

1. *the trivial faces $\emptyset$ and the set containing the zero matrix $\{0\}$,*
2. *or they are generated by a rank k matrix $P \in M_{n,k}$ in the form*

$$F = \{X : \ X = PWP^T, \ W \in S_k^+\}.$$

Proof. We first show that the given sets are faces. It is easy to check that $\emptyset$ and $\{0\}$ are faces. Let $F = \{X : \ X = PWP^T, \ W \in S_k^+\}$ for some k and P as required above. F is a cone because S_k^+ is a cone and for $V, W \in S_k^+$ and $\alpha \geq 0$ we have $\alpha(PVP^T + PWP^T) = P(\alpha(V+W))P^T \in F$, so F is convex. Assume, for contradiction, that there exist $X \in F$, $A, B \in S_n^+$, not both in F, such that $X = \alpha A + (1-\alpha)B$ for some $\alpha \in (0,1)$. W.l.o.g. let $A \notin F$. Then there is a $y \in \mathbb{R}^n$ with $y^T A y > 0$ and $P^T y = 0$. Therefore $0 = y^T X y = \alpha y^T A y + (1-\alpha) y^T B y > 0$, a contradiction.

Now we show that all faces are obtained this way. Observe that $0 \in F$ for any face $F \neq \emptyset$ because $X \in F$ is a convex combination of 0 and $\alpha X \in S_n^+$ for $\alpha \geq 1$. Furthermore, $\{0\}$ is the only face containing only one element. Now let F be a face with $|F| > 1$ and choose some element from the relative interior of F, say $\hat{X} \in \text{relint}(F)$. Let $k = \text{rank}(\hat{X})$ and $\hat{P} \in M_{n,k}$ contain, as columns, the eigenvectors of $\hat{X}$ belonging to the nonzero eigenvalues of $\hat{X}$. Consider the set

$$\hat{F} = \{X : \ X = \hat{P}W\hat{P}^T, \ W \in S_k^+\}.$$

We have already proved that $\hat{F}$ is a face. Since $\hat{X} \in \text{relint}(\hat{F})$ and $\hat{X} \in \text{relint}(F)$ and both are faces, we must have $\hat{F} = F$. ■

In this theorem the columns of P span a subspace of $\mathbb{R}^n$. All eigenvectors that belong to non-zero eigenvalues of matrices in this face are restricted to this subspace. Whenever we say that a face is spanned by a subspace of $\mathbb{R}^n$ we will refer to this characterization.

Any positive semidefinite matrix A can be written as a nonnegative linear combination of n rank one positive semidefinite matrices (Theorem 1.1.13 and eigenvalue decomposition),

$$A = \sum_{i=1}^{n} \lambda_i x_i x_i^T \quad \text{with} \quad \lambda_i \geq 0.$$

By the theorem above $\{\lambda x x^T : \ \lambda \geq 0\}$ is a face for any $x \in \mathbb{R}^n$. These faces cannot be expressed as the convex combination of smaller faces. Consequently, $\{X : \ X = xx^T, \ x \in \mathbb{R}^n, \ \|x\| = 1\}$ forms a minimal generating system for S_n^+. In contrast to polyhedral cones this cone cannot be generated by a finite set. It is also interesting that the faces of S_n^+ have dimension $\binom{k+1}{2}$, so by going from a large face to a smaller one there is a considerable jump in dimension.

The cone of semidefinite matrices induces a partial order on the set of symmetric matrices.

Definition 1.2.10 *(Löwner partial order)*
For $A, B \in S_n$ $A \succeq B$ if $(A - B) \in S_n^+$.
For $A, B \in S_n$ $A \succ B$ if $(A - B) \in S_n^{++}$.

This is the origin of the notation $A \succeq 0$ ($A \succ 0$) in Definition 1.1.2 for $A \in S_n^+$ ($A \in S_n^{++}$).

The *Hadamard product* or sometimes *Schur product* refers to the componentwise multiplication of matrices; for $A, B \in M_{m,n}$

$$A \circ B = [a_{ij} \cdot b_{ij}].$$

The Hadamard product of two positive semidefinite matrices is again a positive semidefinite matrix.

Theorem 1.2.11 *(Schur Product Theorem)*
Let $A, B \in S_n^+$. Then $A \circ B \in S_n^+$. If $A \in S_n^{++}$ and $B \in S_n^{++}$ then $A \circ B \in S_n^{++}$.

Proof. For $v \in \mathbb{R}^n$, one verifies directly that $(A \circ B)v = \operatorname{diag}(A \operatorname{Diag}(v) B)$.

$$\begin{aligned} v^T (A \circ B) v &= v^T \operatorname{diag}(A \operatorname{Diag}(v) B) = \operatorname{tr}(\operatorname{Diag}(v) A \operatorname{Diag}(v) B) \\ &= \langle \operatorname{Diag}(v) A \operatorname{Diag}(v), B \rangle \geq 0. \end{aligned}$$

The last inequality follows from Lemma 1.2.3, because $\operatorname{Diag}(v) A \operatorname{Diag}(v) \in S_n^+$. If $A \in S_n^{++}$ and $v \neq 0$ then $\operatorname{Diag}(v) A \operatorname{Diag}(v)$ is nonzero. If, in addition, $B \in S_n^{++}$ then the inner product must be positive. ■

1.3 Kronecker Products

Kronecker products facilitate the handling of matrix-equations and derivatives. The Kronecker product is a map $\otimes : M_{m,n} \times M_{k,l} \to M_{mk,nl}$ which is defined via

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}.$$

In connection with Kronecker products we will often have to transform a matrix $A \in M_{m,n}$ into a vector in $\mathbb{R}^{mn}$ by stacking the columns of A on top of each other. More precisely, the vec-operator $\operatorname{vec}(\cdot)$ is defined via

$$\operatorname{vec}(A) = \begin{bmatrix} A_{\cdot,1} \\ \vdots \\ A_{\cdot,n} \end{bmatrix}.$$

We list some important properties of Kronecker Products.

Proposition 1.3.1 *Let A, B, C, and D be matrices of appropriate sizes.*

$$(A \otimes B)^T = (A^T \otimes B^T) \tag{1.6}$$
$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \tag{1.7}$$
$$\operatorname{vec}(ABC) = (C^T \otimes A) \operatorname{vec}(B) \tag{1.8}$$
$$\operatorname{vec}(AB + BC) = (I \otimes A + C^T \otimes I) \operatorname{vec}(B) \tag{1.9}$$

Proof. (1.6) to (1.8) are proved by direct computation. For (1.9) assume $A \in M_{m,n}$, $B \in M_{n,k}$, $C \in M_{k,l}$, then by (1.8)

$$\operatorname{vec}(AB + BC) = \operatorname{vec}(AB I_k) + \operatorname{vec}(I_n BC) = (I_k \otimes A + C^T \otimes I_n) \operatorname{vec}(B).$$

■

If the eigenvalues and eigenvectors of two matrices A and B are known then it is easy to construct the eigenvalues and eigenvectors of $A \otimes B$.

Proposition 1.3.2 *Let A (B) have eigenvalues λ_i (μ_i) with corresponding (orthogonal) eigenvectors x_i (y_i) $\in \mathbb{C}^n$. Then all eigenvalues of $A \otimes B$ are given by $\lambda_i \mu_j$ with eigenvector $x_i \otimes y_j$.*

Proof. By (1.7)

$$(A \otimes B)(x_i \otimes y_j) = (Ax_i) \otimes (By_j) = (\lambda_i x_i) \otimes (\mu_j y_j) = \lambda_i \mu_j (x_i \otimes y_j),$$

and because $(x_i \otimes y_j)^T (x_h \otimes y_k) = (x_i^T x_h) \otimes (y_j^T y_k)$ the eigenvectors are orthogonal. ■

The set of symmetric matrices S_n is isomorphic to $\mathbb{R}^{\binom{n+1}{2}}$ via the map $\operatorname{svec}(A)$. The svec operator is the symmetric analogue to vec() and is defined as stacking the columns of the lower triangle of A on top of each other and multiplying the offdiagonal elements with $\sqrt{2}$,

$$\operatorname{svec}(A) := \left[a_{11}, \sqrt{2}a_{21}, \ldots, \sqrt{2}a_{n1}, a_{22}, \sqrt{2}a_{32}, \ldots, a_{nn}\right]^T.$$

The factor $\sqrt{2}$ for offdiagonal elements ensures that, for $A, B \in S_n$,

$$\langle A, B\rangle = \operatorname{tr}(AB) = \operatorname{svec}(A)^T \operatorname{svec}(B).$$

The symmetric Kronecker product $\otimes_s$ is defined for arbitrary square matrices $A, B \in M_n$ by its action on a vector $\operatorname{svec}(C)$ for a symmetric matrix $C \in S_n$,

$$(A \otimes_s B)\operatorname{svec}(C) := \frac{1}{2}\operatorname{svec}(BCA^T + ACB^T).$$

The operator svec and the symmetric Kronecker product were introduced by Alizadeh, Haeberly, and Overton [1998]; here we use the notation of Todd, Toh, and Tütüncü [1998]. Both established several convenient properties of this product.

Proposition 1.3.3

1. $A \otimes_s B = B \otimes_s A$
2. $(A \otimes_s B)^T = B^T \otimes_s A^T$
3. $A \otimes_s I$ *is symmetric if and only if* A *is.*
4. $(A \otimes_s A)^{-1} = A^{-1} \otimes_s A^{-1}$
5. $(A \otimes_s B)(C \otimes_s D) = \frac{1}{2}(AC \otimes_s BD + AD \otimes_s BC)$
6. *If* $A \succ 0$ *and* $B \succ 0$ *then* $(A \otimes_s B) \succ 0$

Proof. Properties 1 and 5 can be verified by direct computation. For the other properties we will need a $\binom{n+1}{2} \times n^2$ matrix U defined by (row and column ordering corresponding to svec and vec)

$$U_{ij,kl} = \begin{cases} 1 & \text{if } i = j = k = l, \\ \frac{1}{\sqrt{2}} & \text{if } i = k \neq j = l, \text{ or } i = l \neq j = k, \\ 0 & \text{otherwise.} \end{cases}$$

For $A \in M_n$ it is the linear operator $U\operatorname{vec}(A) = \operatorname{svec}(\frac{A+A^T}{2})$. For $A \in S_n$ it allows to switch between svec and vec representations via the relations

$$U\operatorname{vec}(A) = \operatorname{svec}(A) \quad \text{and} \quad U^T\operatorname{svec}(A) = \operatorname{vec}(A). \tag{1.10}$$

These properties are straightforward to check. Furthermore, $UU^T = I_{\binom{n+1}{2}}$ and $U^TU\operatorname{vec}(A) = \operatorname{vec}(A)$ for all $A \in S_n$. In particular U^TU is the orthogonal projection matrix onto the space of symmetric matrices. Using (1.10) and (1.8) we obtain

$$A \otimes_s B = \frac{1}{2}U(A \otimes B + B \otimes A)U^T.$$

Property 2 now follows from (1.6), property 3 is due to $A \otimes_s B$ being symmetric if and only if $A \otimes B + B \otimes A$ is symmetric. Finally, properties 4 and 6 are a consequence of Proposition 1.3.2. ∎

1.4 Matrix Calculus

We will now study functions depending on matrices, matrix functions, and their derivatives. Initially, derivatives in matrix variables may cause confusion due to simple arrangement problems of the terms. In this case it helps to resort to the vec-operator. We will therefore define the gradient for functions depending on vectors only, but we will also rearrange the resulting gradient in some other matrix form if it is convenient.

Let x denote a vector in $\mathbb{R}^n$ and let $f : \mathbb{R}^n \to \mathbb{R}^m ; x \mapsto [f_1(x), \ldots, f_m(x)]^T$ be a continuously differentiable function. Then the gradient or derivative of $f(x)$ with respect to x is the $n \times m$ function

$$\nabla_x f = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \cdots & \frac{\partial f_m(x)}{\partial x_1} \\ \vdots & & \vdots \\ \frac{\partial f_1(x)}{\partial x_n} & \cdots & \frac{\partial f_m(x)}{\partial x_n} \end{bmatrix}.$$

The linearization of f in x^0 is

$$f(x^0) + [\nabla_x f(x^0)]^T \Delta x.$$

It is the first order approximation of $f(x^0 + \Delta x)$. Two basic facts about derivatives are listed below.

Proposition 1.4.1 *Let $A \in M_{m,n}$, and $y : \mathbb{R}^n \to \mathbb{R}^m$ and $z : \mathbb{R}^m \to \mathbb{R}^k$ be two continuously differentiable functions.*

$$\nabla_x (Ax) = A^T \tag{1.11}$$

$$\nabla_x z(y(x)) = \nabla_x y \cdot \nabla_y z \qquad \textit{(chain-rule)} \tag{1.12}$$

We illustrate the application of (1.11) to matrices for the products AX and XA with $A, X \in S_n$. For AX we have $\operatorname{vec}(AX) = (I \otimes A) \operatorname{vec}(X)$ by (1.8).

$$\nabla_X \operatorname{vec}(AX) = (I \otimes A) \quad \text{and analogously} \quad \nabla_X \operatorname{vec}(XA) = (A \otimes I).$$

To obtain agreeable representations of the linearizations we observe that $(I \otimes A) \operatorname{vec}(\Delta X) = \operatorname{vec}(A \Delta X)$ and therefore

$$[\nabla_X \operatorname{vec}(AX)]^T \operatorname{vec}(\Delta X) = \operatorname{vec}(A \Delta X) \text{ and } [\nabla_X \operatorname{vec}(XA)]^T \operatorname{vec}(\Delta X) = \operatorname{vec}(\Delta X A).$$

More intuitively, we will write $A\Delta X$ and $\Delta X A$ for the linearizations of AX and XA.

In Chapter 4, when discussing interior point methods, the function

$$-\log\det X = -\log \prod_{i=1}^{n} \lambda_i(X) = -\sum_{i=1}^{n} \log \lambda_i(X).$$

plays a paramount. Since $\det X$ is a continuous function which is positive for positive definite matrices and zero for singular semidefinite matrices, this function grows to infinity as $X \succ 0$ approaches the boundary of the positive semidefinite cone, it acts as a barrier for the iterates. One of its most important properties is its strict convexity.

Lemma 1.4.2 $\log\det X$ *is strictly concave on the set of positive definite matrices.*

Proof. Let $A, B \in S_n^{++}$. We have to show that for $0 \leq \alpha \leq 1$

$$\log\det(\alpha A + (1-\alpha)B) \geq \alpha \log\det A + (1-\alpha)\log\det B$$

is satisfied and that equality only holds for $A = B$ or $\alpha \in \{0,1\}$. We spilt A into $A^{\frac{1}{2}}A^{\frac{1}{2}}$ and denote the i-th eigenvalue of $A^{-\frac{1}{2}}BA^{-\frac{1}{2}}$ by λ_i.

$$\begin{aligned}
&\log\det(A^{\frac{1}{2}}(\alpha I + (1-\alpha)A^{-\frac{1}{2}}BA^{-\frac{1}{2}})A^{\frac{1}{2}}) = \\
&= \log(\det(A)\det(\alpha I + (1-\alpha)A^{-\frac{1}{2}}BA^{-\frac{1}{2}})) \\
&= \log\det(A) + \log\prod_{i=1}^{n}(\alpha + (1-\alpha)\lambda_i) \\
&= \log\det(A) + \sum_{i=1}^{n}\log(\alpha + (1-\alpha)\lambda_i) \\
&\geq \log\det(A) + (1-\alpha)\log\prod_{i=1}^{n}\lambda_i \\
&= \log\det(A) + (1-\alpha)\log\det(A^{-1}B) \\
&= \log\det(A) - (1-\alpha)\log\det A + (1-\alpha)\log\det B
\end{aligned}$$

Strict concavity follows from the strict concavity of the logarithm. ■

To compute the first order optimality conditions for the barrier problem in Section 4.1 we will need the derivative of $\log\det X$.

Theorem 1.4.3 *For a nonsingular matrix $X \in M_n$*

$$\nabla_X \det(X) = \det(X)\,\mathrm{vec}(X^{-T}).$$

Proof. Let $X \in M_n$ be nonsingular. We denote by $X_{ij} \in M_{n-1}$ the matrix obtained from X by deleting the i-th row and j-th column. Laplace expansion along row i yields

$$\det(X) = \sum_{j=1}^{n} x_{ij}(-1)^{i+j}\det(X_{ij}).$$

Therefore

$$\frac{\partial\det(X)}{\partial x_{ij}} = (-1)^{i+j}\det(X_{ij}) = \det\left[X_{\cdot,1},\ldots,X_{\cdot,j-1},e^i,X_{\cdot,j+1},\ldots,X_{\cdot,n}\right].$$

By Cramer's rule the vector $y = \left[\frac{\partial\det(X)}{\partial x_{i1}},\ldots,\frac{\partial\det(X)}{\partial x_{in}}\right]^T$ solves $Xy = \det(X)e^i$. y^T forms the "i-th row" of $\nabla_X\det(X)$. Consequently, the transpose of the solution Y of $XY = \det(X)I$ yields, after application of the vec-operator, $\nabla_X\det(X)$. ■

We obtain the derivative of $\log\det(X)$ for positive definite matrices X by the chain rule,

$$\nabla_X \log\det(X) = \frac{1}{|\det(X)|}\nabla_X\det(X) = \mathrm{vec}(X^{-1}). \tag{1.13}$$

For scalar matrix functions it is common practice to represent the gradient in the form of a matrix. We will keep to this practice and, by a slight abuse of notation, write $\nabla_X\log\det(X) = X^{-1}$.

Remark 1.4.4 *For symmetric matrices we have ignored the fact that $x_{ij} = x_{ji}$, thus the offdiagonal variables are dependent. The correct derivative with respect to the $\binom{n+1}{2}$ variables of a symmetric matrix requires another application of the chain rule. However, the linearizations using ΔX remain correct as long as ΔX is symmetric.*

1.5 Remarks on the Literature

Most results of this chapter can be found in the standard literature on linear algebra, in particular in the outstanding books Horn and Johnson [1985]; Horn and Johnson [1991]. Theorem 1.2.9, the characterization of the faces of the positive semidefinite cone, appears in Barker and Carlson [1975] where it is cited as "part of the oral tradition of the subject". The proof of the Schur product Theorem is taken from Nesterov [1998]. A short survey on some properties of the cone of semidefinite matrices is given in Hill and Waters [1987]. The operator svec and the symmetric Kronecker product were introduced by Alizadeh, Haeberly, and Overton [1998]; here we use the notation of Todd, Toh, and Tütüncü [1998]. The standard reference for convex analysis in general is Rockafellar [1970]. For further reading on Kronecker products and matrix calculus we recommend Horn and Johnson [1991]; Graham [1981].

Chapter 2

Semidefinite Programming

In this chapter we introduce our standard formulation of a primal semidefinite program (PSDP) and derive its dual (DSDP). Semidefinite programming includes several classical optimization problems. Three of them are of relevance in subsequent chapters, namely linear programming, quadratic programming, and semidefinite linear complementarity problems. We show how they can be formulated as standard semidefinite programs. Next we explain how to scale a problem in semidefinite programming. Scaling is a simple but important technique that is often used to facilitate computations or arguments in a proof. Due to the nonpolyhedral structure of semidefinite sets the duality theory of semidefinite programming is somewhat more involved than in linear programming. It may happen that primal and dual optimal values do not coincide, but the problem is well behaved if there exists a positive definite feasible point in the primal or the dual feasible set. We illustrate the basic difficulties by several examples, prove the strong duality theorem, and discuss possibilities to ensure the existence of strictly feasible solutions. We conclude the chapter with some remarks about the geometry of the feasible sets and the computational complexity of semidefinite programming in general.

2.1 Semidefinite Programs

Semidefinite programming is linear programming over the cone of semidefinite matrices. In comparison to standard linear programming the vector $x \in \mathbb{R}^n_+$ of variables is replaced by a matrix variable $X \in S^+_n$. In other words, the cone of the nonnegative orthant $x \geq 0$ is replaced by the cone of semidefinite matrices $X \succeq 0$. In order to pronounce this similarity we first formulate the problem with respect to the vector representation of X,

$$\begin{array}{ll} \min & c^T \operatorname{vec}(X) \\ \text{s.t.} & A \operatorname{vec}(X) = b \\ & X \succeq 0 \end{array}$$

for given vectors $c \in \mathbb{R}^{n^2}$, $b \in \mathbb{R}^m$, and a constraint matrix $A \in M_{m,n^2}$.

Usually semidefinite programs arise in a natural way from problems whose data is given by matrices. The use of the vec-operator tends to hide the obvious and complicates the formulation. It pays to introduce a more agreeable notation by interpreting c and the rows of A as matrices.

Let $C \in M_n$ denote the matrix corresponding to c, *i.e.*,

$$C = \begin{bmatrix} c_1 & c_{1+n} & \cdots & c_{1+(n-1)n} \\ c_2 & c_{2+n} & & c_{2+(n-1)n} \\ \vdots & \vdots & & \vdots \\ c_n & c_{2n} & & c_{nn} \end{bmatrix}$$

so that $c = \mathrm{vec}(C)$. Then the inner product $c^T \mathrm{vec}(X)$ in vector space can equivalently be written as the inner product $\langle C, X \rangle$ in matrix space (1.1),

$$c^T \mathrm{vec}(X) = \sum_{i,j=1}^{n} c_{i+(j-1)n} x_{ij} = \sum_{i,j=1}^{n} C_{ij} x_{ij} = \langle C, X \rangle .$$

Since X is a symmetric matrix, the skew-symmetric part of C is of no influence in this inner product (1.2). Without loss of generality we require C to be a symmetric matrix.

In the same vein we interpret row $A_{i,\cdot}$ as a symmetric matrix $A_i \in S_n$, rewrite the i-th constraint $A_{i,\cdot} \mathrm{vec}(X)$ as $\langle A_i, X \rangle$ and collect the constraints in a linear operator $\mathcal{A} : S_n \to \mathbb{R}^m$,

$$\mathcal{A}X = \begin{bmatrix} \langle A_1, X \rangle \\ \vdots \\ \langle A_m, X \rangle \end{bmatrix} .$$

With this notation we arrive at our standard formulation of a semidefinite program,

$$\text{(PSDP)} \qquad \begin{array}{ll} \min & \langle C, X \rangle \\ \text{s.t.} & \mathcal{A}X = b \\ & X \succeq 0. \end{array} \tag{2.1}$$

In order to derive the dual of this program we need the adjoint operator to $\mathcal{A}$. By definition, it is the operator $\mathcal{A}^T : \mathbb{R}^m \to S_n$ satisfying $\langle \mathcal{A}X, y \rangle = \langle X, \mathcal{A}^T y \rangle$ for all $X \in S_n$ and $y \in \mathbb{R}^m$. Since

$$\langle \mathcal{A}X, y \rangle = \sum_{i=1}^{m} y_i \,\mathrm{tr}(A_i X) = \mathrm{tr}(X \sum_{i=1}^{m} y_i A_i) = \langle X, \mathcal{A}^T y \rangle ,$$

we obtain

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

With respect to the initial vector formulation, $\mathcal{A}^T y$ is simply a different representation of $A^T y$ emphasizing the fact that we are working with matrices.

For constructing the dual we use a Lagrangian approach. The primal equality constraints are lifted into the objective by means of a Lagrange multiplier $y \in \mathbb{R}^m$ so that the primal problem reads $\inf_{X \succeq 0} \sup_{y \in \mathbb{R}^m} \langle C, X \rangle + \langle b - \mathcal{A}X, y \rangle$. The dual of (PSDP) is obtained by interchanging inf and sup,

$$\inf_{X \succeq 0} \sup_{y \in \mathbb{R}^m} \langle C, X \rangle + \langle b - \mathcal{A}X, y \rangle \geq \sup_{y \in \mathbb{R}^m} \inf_{X \succeq 0} \langle b, y \rangle + \langle X, C - \mathcal{A}^T y \rangle . \tag{2.2}$$

The construction implies that the right hand side value cannot exceed the value of the primal problem (see, *e.g.*, Rockafellar [1970], Lemma 36.1; an explicit justification will be given in (2.9)). For the supremum on the right hand side to be finite the inner minimization over $X \succeq 0$ must remain finite for some $\hat{y} \in \mathbb{R}^m$. This requires $C - \mathcal{A}^T \hat{y}$ to be positive semidefinite (Corollary 1.2.7). We write this condition by introducing a slack matrix Z,

$$\text{(DSDP)} \qquad \begin{array}{ll} \max & \langle b, y \rangle \\ \text{s.t.} & \mathcal{A}^T y + Z = C \\ & y \in \mathbb{R}^m ,\ Z \succeq 0. \end{array} \tag{2.3}$$

This is the standard formulation of the dual semidefinite program to (PSDP).

The use of the free variables y in (DSDP) may raise doubts whether (DSDP) is indeed a semidefinite program. To remove these doubts we give a slightly different representation of (PSDP) and (DSDP) that highlights their common structure (see Nesterov and Nemirovskii [1994]). To this end we assume that the system $\mathcal{A}X = b$ is consistent, *i.e.*, there exists an $\hat{X} \in S_n$ satisfying

$\mathcal{A}\hat{X} = b$. In this case we can eliminate all y variables in (DSDP). We first express the cost function in terms of Z,

$$\langle b, y\rangle = \left\langle \mathcal{A}\hat{X}, y\right\rangle = \left\langle \hat{X}, \mathcal{A}^T y\right\rangle = \left\langle \hat{X}, C - Z\right\rangle.$$

y still serves to span the feasible set of Z-values. Let $\mathcal{R}(\mathcal{A}^T)$ denote the range space of $\mathcal{A}^T$, let $\mathcal{N}(\mathcal{A})$ denote the null space of $\mathcal{A}$, and observe that these two subspaces are perpendicular, $\mathcal{R}(\mathcal{A}^T) = \mathcal{N}(\mathcal{A})^\perp$. The dual (primal) equality constraints require a feasible Z (X) to be contained in the affine subspace $\{C + \mathcal{N}(\mathcal{A})^T\}$ ($\{\hat{X} + \mathcal{N}(\mathcal{A})\}$). In these terms the primal-dual pair of problems takes the following form.

$$\begin{array}{lll} \min & \langle C, X\rangle & \\ \text{s.t.} & X \in (S_n^+ \cap \{\hat{X} + \mathcal{N}(\mathcal{A})\}) & \end{array} \qquad \begin{array}{lll} \max & \left\langle \hat{X}, C - Z\right\rangle & \\ \text{s.t.} & Z \in (S_n^+ \cap \{C + \mathcal{N}(\mathcal{A})^\perp\}) & \end{array} \tag{2.4}$$

Thus both, (PSDP) and (DSDP), are semidefinite programs. Any property holding for the primal formulation has its analogue in the dual formulation.

2.1.1 Optimization problems comprised in SDP

So far we have been considering just one semidefinite matrix variable. We may also formulate problems that consist of several semidefinite variables,

$$\begin{array}{lll} \min & \sum_{i=1}^k \langle C_i, X_i\rangle & \\ \text{s.t.} & \sum_{i=1}^k \mathcal{A}_i X_i = b & \\ & X_1 \in S_{n_1}^+, \ldots, X_k \in S_{n_k}^+ & \end{array} \qquad \begin{array}{lll} \max & \langle b, y\rangle & \\ \text{s.t.} & \mathcal{A}_i^T y + Z_i = C_i \quad i = 1, \ldots, k & \\ & y \in \mathbb{R}^m, Z_1 \in S_{n_1}^+, \ldots, Z_k \in S_{n_k}^+. & \end{array} \tag{2.5}$$

In many practical applications such a structure arises naturally and it is important to exploit it to gain in efficiency. However, for theoretical purposes the standard primal dual pair of problems is sufficient. In fact, any semidefinite program in several semidefinite variables of varying dimensions can be formulated equivalently within standard (PSDP), because by Observation 1.1.4

$$X_1 \succeq 0,\ X_2 \succeq 0,\ \ldots,\ X_k \succeq 0 \quad \Longleftrightarrow \quad \begin{bmatrix} X_1 & 0 & \cdots & 0 \\ 0 & X_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & X_k \end{bmatrix} \succeq 0.$$

It is now easy to see that linear programming is indeed a special case of semidefinite programming: Interpret each single component $x_i \geq 0$ as a 1×1 positive semidefinite matrix in (2.5).

Several other convex optimization problems may also be formulated as semidefinite programs. In particular, Schur complements (cf. Theorem 1.1.9) offer excellent possibilities to model nonlinear constraints in semidefinite programming (see, *e.g.*, Boyd, El Ghaoui, Feron, and Balakrishnan [1994]). We illustrate this for a convex quadratic constraint

$$x^T Q x \leq q^T x + c. \tag{2.6}$$

Here, $x \in \mathbb{R}^n$ is the vector of variables and $Q \in S_n^+$, $q \in \mathbb{R}^n$, $c \in \mathbb{R}$ are given constants. Since Q may be singular we factorize it, $Q = C^T C$, and use the identity I in $x^T C^T I C x$ as the positive definite matrix A in the Schur complement. The Schur complement theorem 1.1.9 allows to reformulate the quadratic constraint as

$$\begin{bmatrix} I & Cx \\ x^T C^T & q^T x + c \end{bmatrix} \succeq 0. \tag{2.7}$$

The nonlinear constraint on x is transformed into a linear constraint over the cone of positive semidefinite matrices. This proves that convex quadratically constrained convex quadratic programming problems can be formulated as semidefinite programs.

Another popular class of programming problems are monotone linear complementarity problems. We state them here over the cones of positive semidefinite matrices.

$$\text{(LCP)} \qquad \text{Find } (X,Z) \in S_n^+ \times S_n^+ \text{ with } (X,Z) \in F \text{ and } \langle X, Z\rangle = 0,$$

where F is a monotone affine subspace of $S_n \times S_n$. An affine subspace $F \subset S_n \times S_n$ is called *monotone* if

$$\langle X' - X, Z' - Z\rangle \geq 0 \quad \text{for all } (X,Z),(X',Z') \in F.$$

The special case of linear complementarity problems over the nonnegative orthant ($x, z \in \mathbb{R}^n_+$) are a strict generalization of linear programming, because they include convex quadratic programming. They attracted much interest when Megiddo [1989] showed that interior point methods for linear programming extend to this more general class in a natural way. The semidefinite linear complementarity problem (LCP), however, is not a generalization of semidefinite programming because (LCP) may itself be formulated as a semidefinite program (Kojima, Shida, and Shindoh [1997]). We will do this in several steps.

First observe that any k dimensional affine subspace of $S_n \times S_n$ can be represented in the form

$$\begin{bmatrix} \operatorname{svec}(X) \\ \operatorname{svec}(Z) \end{bmatrix} \in \left\{ \begin{bmatrix} A \\ B \end{bmatrix} y + \begin{bmatrix} a \\ b \end{bmatrix} : y \in \mathbb{R}^k \right\} = F, \tag{2.8}$$

with $A, B \in M_{\binom{n+1}{2},k}$ and $\begin{bmatrix} A \\ B \end{bmatrix}$ having full rank.

Lemma 2.1.1 *(Kojima, Shida, and Shindoh [1997]) The affine subspace F of (2.8) is monotone if and only if $y^T A^T B y \geq 0$ for all $y \in \mathbb{R}^k$.*

Proof. For arbitrary $y, y' \in \mathbb{R}^k$ let (X,Z) and (X',Z') denote the corresponding elements in F of (2.8). Then F is monotone if and only if for all $y, y' \in \mathbb{R}^k$

$$0 \leq \langle X - X', Z - Z'\rangle = \langle A(y-y'), B(y-y')\rangle = (y-y')^T A^T B (y-y').$$

■

A consequence of this Lemma is that the dimension of monotone affine subspaces is bounded by $\binom{n+1}{2}$.

Corollary 2.1.2 *(Kojima, Shida, and Shindoh [1997]) If $k > \binom{n+1}{2}$ then F of (2.8) is not monotone.*

Proof. Let $k > \binom{n+1}{2}$. Choose a $y \in \mathcal{N}(A)$ so that there is a $y' \in \mathbb{R}^k$ with $\langle By, Ay'\rangle > 0$. This is possible because the dimension of $\{By : y \in \mathcal{N}(A)\}$ plus the dimension of $\{Ay : y \in \mathcal{N}(A)^\perp\}$ is $k > \binom{n+1}{2}$ (the matrix $\begin{bmatrix} A \\ B \end{bmatrix}$ has full rank by assumption). Hence, the latter two subspaces of $\mathbb{R}^{\binom{n+1}{2}}$ cannot be perpendicular. For this choice

$$\lim_{\alpha\to\infty} (\alpha y - y')^T A^T B (\alpha y - y') = \lim_{\alpha\to\infty} (y')^T A^T B y' - \alpha (y')^T A^T B y \to -\infty.$$

Now the statement follows from Lemma 2.1.1. ■

For feasible (X,Z) of (LCP) arising from some $y \in \mathbb{R}^k$

$$0 \leq \langle X, Z\rangle = \langle Ay + a, By + b\rangle = y^T Q y + q^t y + d,$$

where $Q = \frac{1}{2}(A^T B + B^T A)$, $q = Ab + Ba$, and $d = a^t b$. For monotone F the matrix Q is positive semidefinite by Lemma 2.1.1, we may therefore rewrite (LCP) as the convex optimization problem

$$\begin{array}{ll} \min & y^T Q y + q^t y + d \\ \text{s.t.} & \operatorname{svec}(X) = A^T y + a, X \succeq 0 \\ & \operatorname{svec}(Z) = B^T y + b, Z \succeq 0 \\ & y \in \mathbb{R}^k. \end{array}$$

To complete the transformation into a semidefinite program it remains to replace the convex quadratic cost function by a semidefinite constraint like in (2.7). Even though this proves that (LCP) is not more general than semidefinite programming, this does not mean that the (LCP) is useless. Indeed, some applications are more naturally formulated as an (LCP) and conveniently solved by interior point methods in this formulation (*e.g.*, the quadratic semidefinite subproblem of the spectral bundle method in Section 5.5). A transformation into standard SDP form would increase computational burden unnecessarily. For theoretical purposes, however, considering standard semidefinite programs is sufficient.

2.1.2 Scaling

Sometimes it will be useful to transform one representation of a semidefinite program into another, for numerical or structural reasons, by transformations of the type $W = QXQ^T$ for some given nonsingular $Q \in M_n$. The map $Q(\cdot)Q^T$ is an automorphism on M_n (a linear and bijective transformation from M_n to M_n), because $\operatorname{vec}(QXQ^T) = (Q \otimes Q)\operatorname{vec}(X)$ and $(Q \otimes Q)$ is nonsingular (cf. (1.8) and Proposition 1.3.2). By Proposition 1.1.7 these linear transformations bijectively map positive definite/semidefinite matrices to positive definite/semidefinite matrices (they belong to the automorphism group of the semidefinite cone). We call this a *scaling* of the variable X (Tunçel [1998]).

Now suppose we want to scale the primal matrix X to $W = QXQ^T$. How do we have to change the constraints of (PSDP) in order to get the same semidefinite program in terms of W? Since $X = Q^{-1}WQ^{-T}$ and, for arbitrary $A \in S_n$,

$$\langle A, X\rangle = \left\langle A, Q^{-1}WQ^{-T}\right\rangle = \left\langle Q^{-T}AQ^{-1}, W\right\rangle,$$

the correct transformation of a coefficient matrix A is $Q^{-T}AQ^{-1}$, which is the adjoint to the inverse transformation of QXQ^T. With

$$\bar{C} = Q^{-T}CQ^{-1}, \quad \bar{A}_i = Q^{-T}A_iQ^{-1} \quad i = 1, \ldots, m,$$

and the linear operators

$$\bar{\mathcal{A}}W = \begin{bmatrix} \langle \bar{A}_1, W\rangle \\ \vdots \\ \langle \bar{A}_m, W\rangle \end{bmatrix} \quad \text{and} \quad \bar{\mathcal{A}}^T\bar{y} = \sum_{i=1}^{m} \bar{y}_i\bar{A}_i$$

we obtain the transformed primal dual pair

$$\begin{array}{llll} & \min & \langle \bar{C}, W\rangle & \\ (\mathrm{P}_Q) & \text{s.t.} & \bar{\mathcal{A}}W = b & \\ & & W \succeq 0 & \end{array} \qquad \begin{array}{llll} & \max & \langle b, \bar{y}\rangle & \\ (\mathrm{D}_Q) & \text{s.t.} & \bar{\mathcal{A}}^T\bar{y} + \bar{Z} = \bar{C} & \\ & & \bar{Z} \succeq 0. & \end{array}$$

Proposition 2.1.3 *X is a feasible solution of (PSDP) if and only if the associated $W = QXQ^T$ is a feasible solution of (P_Q). Furthermore, X and W satisfy $\langle C, X\rangle = \langle \bar{C}, W\rangle$. (y, Z) is a feasible solution of (DSDP) if and only if the associated $(\bar{y}, \bar{Z}) = (y, Q^{-T}ZQ^{-1})$ is a feasible solution of (D_Q). Trivially, $\langle b, y\rangle = \langle b, \bar{y}\rangle$.*

Proof. Clear by construction. ■

2.2 Duality Theory

We derived the dual to (PSDP) by a Lagrangian approach in (2.2). The gap between a dual feasible solution (y, Z) and a primal feasible solution X is

$$\langle C, X\rangle - \langle b, y\rangle = \left\langle \mathcal{A}^Ty + Z, X\right\rangle - \langle \mathcal{A}X, y\rangle = \langle Z, X\rangle \geq 0. \tag{2.9}$$

The last inequality follows from Lemma 1.2.3. The property that the objective value of any primal feasible solutions is greater or equal to the objective value of any dual feasible solution is called *weak duality*.

If $\langle Z, X \rangle$ turns out to be zero then this primal-dual pair is an optimal solution. In contrast to linear programming, however, it is no longer true that optimality implies $\langle Z, X \rangle = 0$. We illustrate this by extending an example of Vandenberghe and Boyd [1996].

Example 2.2.1 *Consider the following primal semidefinite program.*

$$\begin{array}{ll} \min & x_{12} \\ \text{s.t.} & \begin{bmatrix} 0 & x_{12} & 0 \\ x_{12} & x_{22} & 0 \\ 0 & 0 & 1+x_{12} \end{bmatrix} \succeq 0 \end{array}$$

In order to determine the dual program we write the cost function and constraints in matrix form and give the corresponding dual variables in brackets

$$\begin{array}{lll} \min & \left\langle \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, X \right\rangle & \\ \text{s.t.} & \left\langle \begin{bmatrix} 0 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, X \right\rangle = 1, & [y_1] \\ & \left\langle \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, X \right\rangle = 0, & [y_2] \\ & \left\langle \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, X \right\rangle = 0, & [y_3] \\ & \left\langle \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, X \right\rangle = 0, & [y_4] \\ & X \succeq 0. & \end{array}$$

Dualizing by the standard procedure yields

$$\max\ y_1 \text{ s.t. } Z = C - y_1 A_1 - y_2 A_2 - y_3 A_3 - y_4 A_4 \succeq 0.$$

The dual program can be written in the form

$$\begin{array}{ll} \max & y_1 \\ \text{s.t.} & Z = \begin{bmatrix} -y_2 & \frac{1+y_1}{2} & -y_3 \\ \frac{1+y_1}{2} & 0 & -y_4 \\ -y_3 & -y_4 & -y_1 \end{bmatrix} \succeq 0. \end{array}$$

A necessary condition for the primal matrix to be positive semidefinite is that x_{12} is zero because $x_{11} = 0$ (see Observation 1.1.6). Likewise we obtain from $z_{22} = 0$ that $z_{12} = 0$ and hence $y_1 = -1$ in the dual program. The gap between any pair of primal and dual optimal solutions is one.

This insufficiency of the primal-dual pair is due to the dualization procedure (2.2) which is purely algebraic and does not take into account the actual geometry of the feasible sets. In this particular example the primal equality constraints imply that any feasible $X \succeq 0$ has a zero eigenvalue with eigenvector $[1,0,0]^T$. By Theorem 1.2.9 the primal feasible set is contained in a face of the semidefinite cone that has the following form,

$$F = \left\{ PWP^T : W \succeq 0 \right\} \quad \textit{with} \quad P = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The particular choice of P is convenient, but any P whose columns form a basis of the space orthogonal to the nullspace of the feasible set will do as well. If we replace the condition $X \succeq 0$ by $X \in F$ in the primal problem, the primal problem remains unchanged. Constructing the dual in analogy to (2.2) yields

$$\inf_{X \in F} \sup_{y \in \mathbb{R}^m} \langle C, X \rangle + \langle b - \mathcal{A}X, y \rangle \geq \sup_{y \in \mathbb{R}^m} \inf_{X \in F} \langle b, y \rangle + \left\langle X, C - \mathcal{A}^T y \right\rangle .$$

Because F is a cone $\inf_{X \in F} \left\langle X, C - \mathcal{A}^T y \right\rangle$ is finite if and only if $\left\langle X, C - \mathcal{A}^T y \right\rangle \geq 0$ for all $X \in F$, i.e., *$C - \mathcal{A}^T y$ is in the dual cone to F. For $Z = C - \mathcal{A}^T y$ the condition reads*

$$\left\langle PWP^T, Z \right\rangle = \left\langle W, P^T Z P \right\rangle \geq 0 \quad \forall W \succeq 0.$$

In words, Z must be positive semidefinite on the subspace spanned by P. Applying this to the current example we obtain

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -y_2 & \frac{1+y_1}{2} & -y_3 \\ \frac{1+y_1}{2} & 0 & -y_4 \\ -y_3 & -y_4 & -y_1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -y_4 \\ -y_4 & -y_1 \end{bmatrix} \succeq 0.$$

For this specialized dual problem the optimal solution is attained for $y_1 = 0$ and the gap between the optimal values of the primal problem and the specialized dual problem has disappeared.

The primal cone of the original primal problem was too large, this restricted the dual cone too much. Reducing the size of the primal cone to its non-redundant part increased the dual cone. The additional freedom enabled the dual to reach the same objective value.

In the following we will prove that the gap between optimal primal and dual objective value is guaranteed to be zero (*i.e.*, strong duality holds) if at least one of (PSDP) and (DSDP) has a strictly feasible point.

Definition 2.2.2
A point X is strictly feasible *for (PSDP) if it is feasible for (PSDP) and satisfies $X \succ 0$.*
A pair (y, Z) is strictly feasible *for (DSDP) if it is feasible for (DSDP) and satisfies $Z \succ 0$.*

Geometrically, the existence of a strictly feasible primal solution ensures that the primal cone is non-redundant in the sense that we cannot restrict it to one of its faces without changing the primal feasible set.

The assumption of the existence of such a point is referred to as *Slater condition* (see Hiriart-Urruty and Lemaréchal [1993a]; Bertsekas [1995]). Regularity assumptions of this kind form sufficient conditions for strong duality in general convex programming (see Hiriart-Urruty and Lemaréchal [1993a]; Rockafellar [1970]), but for the sake of simplicity we will follow the specialized approach of Alizadeh [1995].

Here is a rough outline of the proof of strong duality. We will assume that (DSDP) has a strictly feasible point and that, in spite of this assumption, the primal optimal objective value p^* is strictly greater than the dual optimal value d^*. The existence of a dual strictly feasible point guarantees that the set $\left\{ \begin{bmatrix} \langle C, X \rangle \\ \mathcal{A}X \end{bmatrix} : X \succeq 0 \right\}$ is a closed convex set (see Lemma 2.2.3 below). By assumption, $\begin{bmatrix} d^* \\ b \end{bmatrix}$ is not contained in this set, so there is a hyperplane strictly separating this point from the feasible set. Via Lemma 2.2.4 we will use this separating hyperplane to construct a better dual solution than d^*. This will establish the desired contradiction.

We will prove the two lemmas mentioned above for a general linear operator $\mathcal{B}X$. $\mathcal{B}X$ corresponds to $\begin{bmatrix} \langle C, X \rangle \\ \mathcal{A}X \end{bmatrix}$ in the proof of the strong duality theorem. We start with the closedness of the image of $\mathcal{B}$.

Lemma 2.2.3 *Let $\mathcal{B} : S_n \to \mathbb{R}^m$ be a linear operator. Assume that there exists a $\hat{y} \in \mathbb{R}^m$ such that $\hat{Z} = \mathcal{B}^T\hat{y} \succ 0$. Then the set $\{\mathcal{B}X : X \succeq 0\}$ is closed.*

Proof. Let X_i, $i \in \mathbb{N}$, be a sequence of positive semidefinite matrices with $\lim_{i\to\infty} \mathcal{B}X_i = b$. We first prove the boundedness of this sequence. We use Lemma 1.2.4 to bound the maximal eigenvalue of the X_i.

$$\langle \hat{y}, \mathcal{B}X_i \rangle = \left\langle \mathcal{B}^T\hat{y}, X_i \right\rangle = \left\langle \hat{Z}, X_i \right\rangle \geq \lambda_{\min}(\hat{Z})\, \lambda_{\max}(X_i).$$

Since

$$\lim_{i\to\infty} \langle \hat{y}, \mathcal{B}X_i \rangle = \langle \hat{y}, b \rangle,$$

the maximal eigenvalue $\lambda_{\max}(X_i)$ is bounded for this sequence and hence, by Observations 1.1.5 and 1.1.1, the elements of X_i remain bounded as well. The sequence remains in a compact subset of S_n^+. Therefore there is a convergent subsequence converging to some $\hat{X} \succeq 0$ with $\mathcal{B}\hat{X} = b$. ■

Since $\mathcal{B}$ is a linear operator, the image of a convex set is again convex. If the image of the semidefinite cone is closed we can apply a separation theorem to obtain the following analogue of the Farkas-Lemma.

Lemma 2.2.4 *Assume that $\{\mathcal{B}X : X \succeq 0\}$ is closed and let $b \in \mathbb{R}^m$. Then either there exists an $X \succeq 0$ such that $\mathcal{B}X = b$ or there is an $y \in \mathbb{R}^m$ such that $\mathcal{B}^T y \succeq 0$ and $b^T y < 0$.*

Proof. First suppose that there exists an $\hat{X} \succeq 0$ with $\mathcal{B}\hat{X} = b$. From $b^T y < 0$ we conclude

$$\langle b, y \rangle = \left\langle \mathcal{B}\hat{X}, y \right\rangle = \left\langle \hat{X}, \mathcal{B}^T y \right\rangle < 0.$$

Therefore, by Corollary 1.2.7, $\mathcal{B}^T y$ cannot be positive semidefinite. Now suppose that there is no $X \succeq 0$ with $\mathcal{B}X = b$. Because the set $\{\mathcal{B}X : X \succeq 0\}$ is convex and closed there exists a hyperplane $\hat{y}$ separating b from this set (Rockafellar [1970], Corollary 11.4.2), *i.e.*, $b^T\hat{y} < 0$ and $\langle \hat{y}, \mathcal{B}X \rangle = \left\langle \mathcal{B}^T\hat{y}, X \right\rangle \geq 0$ for all $X \succeq 0$. By Corollary 1.2.7 this ensures $\mathcal{B}^T\hat{y} \succeq 0$. ■

We are now ready to prove the strong duality theorem.

Theorem 2.2.5 *(Strong Duality)*
Assume that there exists a strictly feasible solution $(\hat{y}, \hat{Z})$ for (DSDP) and let

$$\begin{aligned} p^* &= \inf\{\langle C, X \rangle : \mathcal{A}X = b, X \succeq 0\} \quad \textit{and} \\ d^* &= \sup\left\{\langle b, y \rangle : \mathcal{A}^T y + Z = C, Z \succeq 0\right\}. \end{aligned}$$

Then $p^* = d^*$ and if p^* is finite it is attained for some $X \in \{X \succeq 0 : \mathcal{A}X = b\}$.

Proof. We note that $p^* = -\infty$ is impossible because by weak duality $p^* \geq b^T\hat{y}$. Observe that $\left\{ \begin{bmatrix} \langle C, X \rangle \\ \mathcal{A}X \end{bmatrix} : X \succeq 0 \right\}$ is closed because $\begin{bmatrix} 1 \\ -\hat{y} \end{bmatrix}$ yields the correct vector to apply Lemma 2.2.3. Assume, for contradiction, that $p^* > d^*$ with d^* finite. Then the system

$$\begin{aligned} \langle C, X \rangle &= d^* \\ \mathcal{A}X &= b \\ X &\succeq 0 \end{aligned}$$

is infeasible, implying by Lemma 2.2.4 that there exists some vector $\begin{bmatrix} y_0 \\ \bar{y} \end{bmatrix}$ such that

$$d^* y_0 + b^T\bar{y} < 0 \qquad \text{and} \qquad y_0 C + \mathcal{A}^T\bar{y} \succeq 0.$$

Now consider the following cases.

(i) $y_0 = 0$. We get $b^T\bar{y} < 0$ and $\mathcal{A}^T\bar{y} \succeq 0$. $\hat{y} + \alpha(-\bar{y})$ with $\alpha \geq 0$ is a dual feasible ray along which the dual objective function is strictly increasing. Thus $d^* = \infty$.

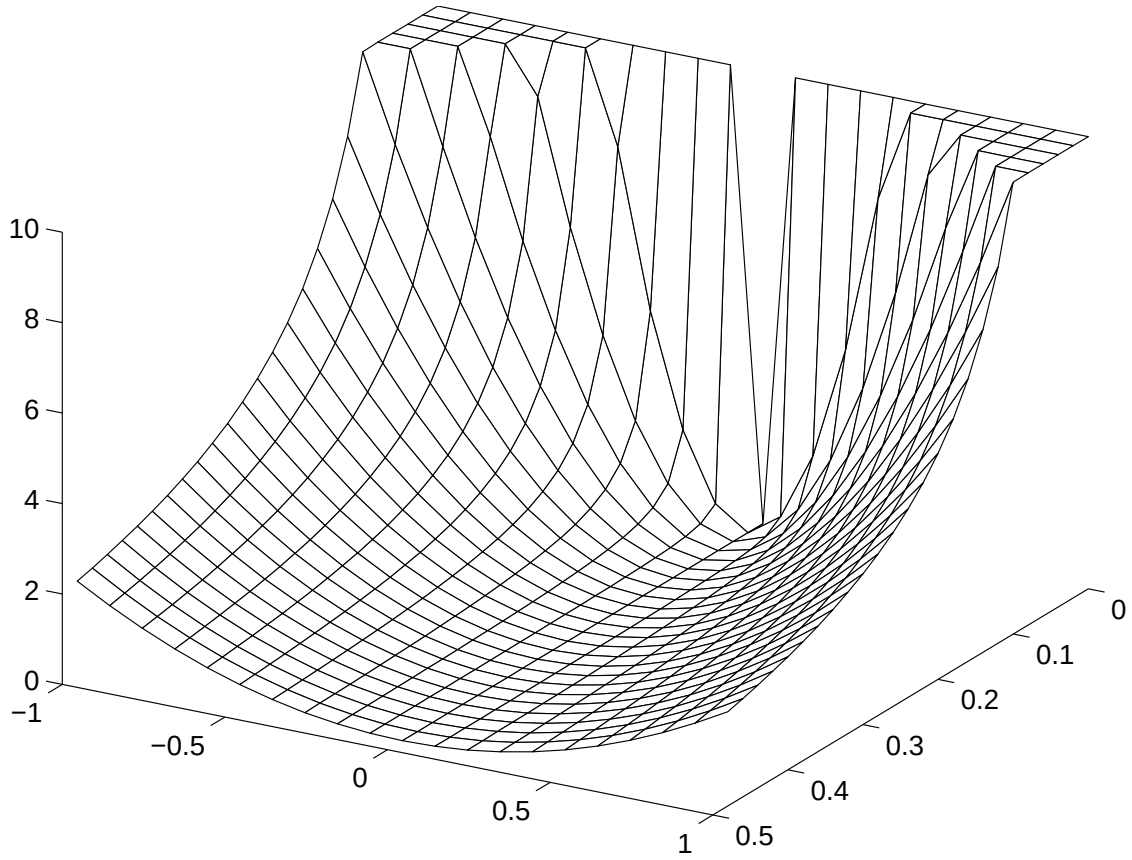

Figure 2.1 An illustration of the feasible set of Example 2.2.7.
$\varepsilon \in (0, 0.5]$, $x_{12} \in [-1, \infty)$, $x_{22}(\varepsilon, x_{12}) = x_{12}^2/\varepsilon$

(ii) $y_0 > 0$. Dividing by y_0 yields $d^* + b^T\bar{y}/y_0 < 0$ and $C + \mathcal{A}^T\bar{y}/y_0 \succeq 0$. The choice $y = -\bar{y}/y_0$ yields a dual feasible solution with better objective value than d^*.

(iii) $y_0 < 0$. Dividing by $-y_0$ yields $-d^* - b^T\bar{y}/y_0 < -\varepsilon_1$ with $\varepsilon_1 > 0$ small enough and $-C - \mathcal{A}^T\bar{y}/y_0 \succeq 0$. Select some feasible solution $\tilde{y}$ with $d^* - b^T\tilde{y} \leq \varepsilon_2$ with $0 < \varepsilon_2 < \varepsilon_1$ and $C - \mathcal{A}^T\tilde{y} \succeq 0$. We add these inequalities to obtain $b^T(-\tilde{y} - \bar{y}/y_0) < 0$ and $\mathcal{A}^T(-\tilde{y} - \bar{y}/y_0) \succeq 0$. As in case (i) this yields an improving ray $\hat{y} + \alpha(\tilde{y} + \bar{y}/y_0)$ ($\alpha \geq 0$) for (DSDP).

This proves $p^* = d^*$. Attainment of the primal optimal solution for finite p^* follows from the closedness of the set $\left\{ \begin{bmatrix} \langle C, X\rangle \\ \mathcal{A}X \end{bmatrix} : X \succeq 0 \right\}$. ■

We state the result for all possible primal-dual combinations in the following corollary.

Corollary 2.2.6 *Let p^* and d^* be defined as in Theorem 2.2.5.*
(i) If (PSDP) is strictly feasible with p^* finite, then $p^* = d^*$ and this value is attained for (DSDP).
(ii) If (DSDP) is strictly feasible with d^* finite, then $p^* = d^*$ is attained for (PSDP).
(iii) If (PSDP) and (DSDP) are both strictly feasible, then $p^* = d^*$ is attained for both problems.

In view of the proof of the Strong Duality Theorem let us turn once more to Example 2.2.1.

Example 2.2.7 *For the primal program of Example 2.2.1, the set*

$$\mathcal{B} = \left\{ \begin{bmatrix} \langle C, X\rangle \\ \mathcal{A}X \end{bmatrix} : X \succeq 0 \right\} = \left\{ [x_{12}, x_{33} - x_{12}, x_{11}, 2x_{13}, 2x_{23}]^T : X \succeq 0 \right\}$$

is not closed, because for the sequence defined by the matrices

$$X_k = \begin{bmatrix} \frac{1}{k} & -1 & 0 \\ -1 & k & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{for} \quad 1 \leq k \in \mathbb{N}$$

the images converge to $\lim_{k\to\infty} \begin{bmatrix} \langle C, X_k \rangle \\ \mathcal{A}X_k \end{bmatrix} = [-1, 1, 0, 0, 0]^T$. *The latter vector is clearly in the closure of* $\mathcal{B}$ *but cannot be contained in* $\mathcal{B}$, *because a positive semidefinite matrix* $X \succeq 0$ *cannot satisfy* $x_{11} = 0$ *and* $x_{12} = -1$.

In order to get a better geometric understanding why this causes a gap for the optimal solutions of Example 2.2.1 we examine the case where the diagonal element x_{11} *is set to some positive value* $\varepsilon > 0$. *Then the standard primal dual pair reads*

$$\begin{array}{lll} \min & x_{12} & \\ \text{s.t.} & \begin{bmatrix} \varepsilon & x_{12} & 0 \\ x_{12} & x_{22} & 0 \\ 0 & 0 & 1 + x_{12} \end{bmatrix} \succeq 0 \end{array} \qquad \begin{array}{lll} \max & y_1 + \varepsilon y_2 & \\ \text{s.t.} & \begin{bmatrix} -y_2 & \frac{1+y_1}{2} & -y_3 \\ \frac{1+y_1}{2} & 0 & -y_4 \\ -y_3 & -y_4 & -y_1 \end{bmatrix} \succeq 0. \end{array}$$

Clearly, for $\varepsilon > 0$ *the primal has a strictly feasible solution. The optimal solution of the dual problem is* -1 *as before and it is attained for* $y_1 = -1$ *and* $y_2 = 0$. X *is positive semidefinite for*

$$x_{12} \in [-1, \infty) \quad \text{and} \quad x_{22} \geq \frac{x_{12}^2}{\varepsilon}.$$

For any $\varepsilon > 0$, *the optimal solution of the primal problem is* -1 *and it is attained for* $x_{12} = -1$ *and* $x_{22} \geq \frac{1}{\varepsilon}$; *there is no gap between primal and dual optimal solutions.*

Figure 2.1 displays the lower bound on x_{22} *(truncated at value 10 for better visibility) in dependence of* $x_{12} \in [-1, 1]$ *and* $\varepsilon \in (0, 0.5]$. *The epigraph of this function can be interpreted as the primal feasible set.*

At $\varepsilon = 0$ *the feasible range of* x_{12} *contracts to a single point,* $x_{12} \in \{0\}$. *The optimal value of the primal problem suddenly rises from* -1 *to* 0. *As in the proof of Theorem 2.2.5 we would like to construct an improving direction for the dual solution with* $d^* = -1$. *However, there is no hyperplane strictly separating the point* $\begin{bmatrix} d^* \\ b \end{bmatrix} = [-1, 1, 0, 0, 0]^T$ *from* $\mathcal{B}$, *because we have proved above that the point is contained in the closure of* $\mathcal{B}$.

If there is no strictly feasible dual solution, the primal optimal solution may not be attained. This is illustrated by the following example.

Example 2.2.8

$$\begin{array}{lll} \min & x_{11} & \\ s.t. & \begin{bmatrix} x_{11} & 1 \\ 1 & x_{22} \end{bmatrix} \succeq 0 \end{array} \qquad \begin{array}{lll} \max & 2y_1 & \\ s.t. & \begin{bmatrix} 1 & -y_1 \\ -y_1 & 0 \end{bmatrix} \succeq 0 \end{array}$$

The primal problem has a strictly feasible solution ($x_{11} = 2$, $x_{22} = 2$*) and indeed the dual optimal solution* 0 *is attained for* $y_1 = 0$ *which is also the only feasible solution. By the strict feasibility of the primal problem we know that the infimum of the objective values of the primal feasible solutions must be zero. To check for primal attainment we compute the feasible set explicitly. Because of the semidefiniteness constraint the determinants of the principal submatrices must be nonnegative,*

$$x_{11} \geq 0, \qquad x_{22} \geq 0, \qquad x_{11}x_{22} - 1 \geq 0.$$

This yields the lower bound $x_{11} \geq \frac{1}{x_{22}}$ *which is zero for* $x_{22} \to \infty$. *The primal optimal value is not attained. Note, that the dual does not have a strictly feasible solution and that the primal feasible set is not polyhedral.*

What can we do if the semidefinite program at hand does not have a strictly feasible point? If we know the minimal face of the positive semidefinite cone that contains the feasible set then we can project the problem onto this face and obtain a well posed problem (Wolkowicz [1981]). In particular, let $F = \{PWP^T : W \in S_k^+\}$ with $P \in M_{n,k}$ denote this minimal face. The columns of P are a basis of the subspace that is orthogonal to the null space common to all feasible matrices X. In the projected problem W takes the place of X. The projected cost matrix is P^TCP, because

$$\langle C, X\rangle = \langle P^TCP, W\rangle \quad \text{for} \quad X = PWP^T \in F.$$

Likewise we define the projected constraints by

$$\mathcal{A}_P W = \begin{bmatrix} \langle P^TA_1P, W\rangle \\ \vdots \\ \langle P^TA_mP, W\rangle \end{bmatrix}.$$

Then the projected problem

$$\begin{array}{ll} \min & \langle P^TCP, W\rangle \\ \text{s.t.} & \mathcal{A}_P W = b \\ & W \succeq 0 \end{array}$$

is equivalent to the original problem and has strictly feasible solutions.

The minimal cone can be constructed explicitly if a point in the relative interior of the feasible set is known.

Lemma 2.2.9 *Let $X \in \mathcal{X} = \{X \succeq 0 : \mathcal{A}X = b\}$ with eigenvalue decomposition $X = P\Lambda P^T$, $P^TP = I$ and $\Lambda \succ 0$ diagonal. Denote by $S_P = \{PVP^T : V \succeq 0\}$ the face of S_n^+ spanned by P. Then X is in the relative interior of $\mathcal{X}$ if and only if S_P is the smallest face of S_n^+ containing $\mathcal{X}$.*

Proof. Let X be in the relative interior of $\mathcal{X}$. The set S_P is the smallest face of S_n^+ containing X, so it remains to prove that S_P contains $\mathcal{X}$. Suppose it does not, then there is a $\bar{X} \in \mathcal{X}$ not contained in S_P. This $\bar{X}$ must have a normalized eigenvector v with vv^T not contained in S_P. Since X is in the relative interior of $\mathcal{X}$ there is an $\varepsilon > 0$ so that $\tilde{X} = X - \varepsilon(\bar{X} - X) \in \mathcal{X}$. But for $w = v - PP^Tv$ we have $Xw = 0$ and therefore $w^T\tilde{X}w = -\varepsilon w^T\bar{X}w < 0$, yielding a contradiction.

Now let S_P be the smallest face of S_n^+ containing $\mathcal{X}$. Since for $\bar{X} \in \mathcal{X}$ it follows that $\bar{X} \in S_P$ there is a $\bar{V} \succeq 0$ with $\bar{X} = P\bar{V}P^T$. For $\bar{V}$ there is $\varepsilon > 0$ so that for all $|\delta| < \varepsilon$ the matrix $V = \Lambda + \delta(\bar{V} - \Lambda) \succeq 0$ is positive semidefinite and therefore $X + \delta(\bar{X} - X) = PVP^T \in \mathcal{X}$. So X is in the relative interior of $\mathcal{X}$. ■

If we do not know this minimal face there is no obvious way to arrive at an equivalent well posed problem. In theory it is possible to construct, via an algebraic description of the minimal face, an extended dual semidefinite program, called the extended Lagrange-Slater dual, that guarantees that the gap between primal optimal value and dual optimal value is zero (Ramana [1997]). The construction is quite involved and requires the introduction of several additional semidefinite variables. Although the extended slater dual can be constructed in polynomial time, the computational burden is too high for practical applications.

2.3 Geometry and Complexity

Next we study the facial structure of feasible sets. A feasible set of a semidefinite program is the intersection of an affine subspace with the semidefinite cone. The faces of intersections of convex sets are the intersections of the faces of the convex sets. Consequently we can expect that the facial structure of the semidefinite cone has a strong influence on the facial structure of the feasible set. One such consequence is that in general feasible sets are not polyhedral. Another one is that optimal solutions are likely to have small rank. As the low dimensional faces of the semidefinite cone correspond to matrices of small rank, we can expect that matrices contained in low dimensional faces of the feasible set will have small rank, as well. The following lemma gives a mathematically precise explanation of this phenomenon.

Lemma 2.3.1 *(Pataki [1998])*
(i) Let F be a face of dimension k of the feasible set of (PSDP). For $X \in F$ the rank $r = \operatorname{rank}(X)$ is bounded by

$$\binom{r+1}{2} \leq m + k.$$

(ii) Let F be a face of dimension k of the set $\{Z \succeq 0 : \exists y \in \mathbb{R}^m : Z + \mathcal{A}^T y = C\}$ of feasible Z-values of (DSDP). For $Z \in F$ the rank $r = \operatorname{rank}(Z)$ is bounded by

$$\binom{r+1}{2} \leq \binom{n+1}{2} - m + k.$$

Proof. Since (ii) is the dual result to (i), we only have to prove (i). Take some feasible $\hat{X}$ of rank r and let $P \in M_{n,r}$ contain, as columns, the eigenvectors corresponding to the positive eigenvalues of $\hat{X}$. For $\varepsilon > 0$ small enough $\hat{X} + \varepsilon \{PWP^T : \mathcal{A}(PWP^T) = 0, W \in S_r, \|W\| \leq 1\}$ is a convex set feasible for (PSDP) with $\hat{X}$ in its relative interior. The affine dimension of this set is at least $\binom{r+1}{2} - m$, because S_r has dimension $\binom{r+1}{2}$. Thus a face containing $\hat{X}$ must have dimension at least $\binom{r+1}{2} - m$. Reversely, $\binom{r+1}{2} > m + k$ implies that X cannot be contained in F. ■

Before discussing semidefinite relaxations of combinatorial optimization problems as well as practical methods for solving semidefinite programs it is worth to spend some thought on the complexity of the problem. It is well known that "under reasonable assumptions" convex programming is of polynomial complexity. In particular, if we have a full dimensional, compact convex set given by a weak violation oracle with its "interesting region" contained in a ball centered at the origin with radius R then there exists an oracle polynomial time algorithm that solves the weak optimization problem (Grötschel, Lovász, and Schrijver [1988], Corollary 4.2.7). In the case of semidefinite programming a polynomial weak violation routine is obtained by Gaussian elimination pivoting on diagonal elements (cf. Remark 1.1.15). The next two examples illustrate that the assumptions are indeed necessary for semidefinite programming.

Example 2.3.2 *(Ramana [1997])*

$$\begin{array}{ll} \min & x_m \\ s.t. & (x_1 - 4) \succeq 0, \begin{bmatrix} 1 & x_1 \\ x_1 & x_2 \end{bmatrix} \succeq 0, \begin{bmatrix} 1 & x_2 \\ x_2 & x_3 \end{bmatrix} \succeq 0, \ldots, \begin{bmatrix} 1 & x_{m-1} \\ x_{m-1} & x_m \end{bmatrix} \succeq 0. \end{array}$$

The encoding length of this program is $O(m)$. What can we say about its feasible set? We get $x_1 \geq 2^2$, $x_2 \geq x_1^2 \geq (2^2)^2 = 2^{(2^2)}$, $x_3 \geq x_2^2 \geq 2^{(2^3)}$, ..., $x_m \geq x_{m-1}^2 \geq 2^{(2^m)}$. A strictly feasible solution exists and the optimal solution is obtained by setting all variables to their respective lower bounds. But the optimal solution is doubly exponential in m and the feasible region is doubly exponentially far away from the origin, i.e., *R grows doubly exponentially in m. Thus, the encoding length of any feasible solution is $\Omega(2^m)$ if binary encoding is used.*

Example 2.3.3

$$\begin{array}{ll} \min & x_{12} \\ \text{s.t.} & \begin{bmatrix} 1 & x_{12} \\ x_{12} & 2 \end{bmatrix} \succeq 0. \end{array}$$

Although all coefficients are integers the optimal solution is $x_{12} = -\sqrt{2}$. In contrast to linear programming we cannot expect solutions to be rational numbers when coefficients are restricted to integers.

Strong bounds on the complexity of semidefinite programming were obtained by Porkolab and Khachiyan [1997]. They employ complexity results of the first order theory over the reals, where solutions may be described as the roots of polynomials with integral coefficients. We cite the main results in the following theorem.

Theorem 2.3.4 *(Porkolab and Khachiyan [1997]) For integral $\mathcal{A} : S_n \to \mathbb{R}^m$ and $b \in \mathbb{Z}^m$ let*

$$\mathcal{F} = \{X \succeq 0 : \mathcal{A}X \leq b\},$$

and let l denote the maximum bitlength of the coefficients in $\mathcal{A}$ and b.

1. *If $\mathcal{F} \neq \emptyset$ then it has a solution X satisfying $\|X\| \leq R$ with $\log R = l \cdot n^{O(\min\{m,n^2\})}$. If, in addition, $\mathcal{F}$ is bounded, then $\|X\| \leq R$ for all $X \in \mathcal{F}$.*

2. *It can be tested in $mn^{O(\min\{m,n^2\})}$ arithmetic operations over $l \cdot n^{O(\min\{m,n^2\})}$ bit numbers whether F is empty.*

Another important result, due to the same authors, states that integer semidefinite programming is polynomially solvable in fixed dimensions. In this case, the coefficients of $\mathcal{A}$ and b may be algebraic numbers, *i.e.*, roots of polynomials.

Theorem 2.3.5 *(Khachiyan and Porkolab [1997]) For fixed n there exists a polynomial time algorithm that finds an integral $X \in S_n^+$ satisfying $\mathcal{A}X \leq b$ or decides that no such matrix exists.*

Note that the result does not depend on the number of constraints m, but only on the dimension of X.

In terms of complexity classes, Ramana [1997] proved by means of his extended Lagrange-Slater dual that there is an exact theorem of the alternative for the semidefinite feasibility problem: Either an appropriate modification of the primal problem is feasible or its extended slater dual is feasible. Since the size of a binary encoding of both problems is polynomially bounded in the size of a binary encoding of the primal problem, this established that in the Turing model of computation, the semidefinite feasibility problem is either in *NP*∩*co-NP* or outside *NP*∪*co-NP*. In the real number model of computation it is in *NP*∩*co-NP* (this follows from Theorem 1.1.14). It is not known, however, whether in the Turing model semidefinite programming is in *NP* or not. As illustrated in Example 2.3.2, the size of a binary encoding of a solution may grow exponentially in the input size, so standard encoding schemes will not suffice. Currently there is little hope to prove either result, but Example 2.3.2 is taken as an indication that SDP might not be in *NP*.

2.4 Remarks on the Literature

Semidefinite programming is a special case of linear programming over cones. Some important references are Duffin [1956]; Ben-Israel, Charnes, and Kortanek [1969]; Ben-Israel, Charnes, and Kortanek [1971]; Wolkowicz [1981]; Luo, Sturm, and Zhang [1997]. Nesterov and Nemirovskii [1994], Boyd, El Ghaoui, Feron, and Balakrishnan [1994], Vandenberghe and Boyd [1996], and Ben-Tal and Nemirovski [1998] offer a rich source for applications of (SDP) in various fields. The transformation of (LCP) to (SDP) was developed in Kojima, Shida, and Shindoh [1997]. The strong duality theorem presented here is covered by more general results from convex analysis (Rockafellar [1970]; Hiriart-Urruty and Lemaréchal [1993a]), our approach follows Alizadeh [1995]. Example 2.2.1 is based on an example of Vandenberghe and Boyd [1996], Example 2.2.8 is folklore. Papers on problem specific dualization and the extended slater dual are Wolkowicz [1981]; Ramana [1997]; Ramana, Tunçel, and Wolkowicz [1997]. Lemma 2.2.9 is folklore, Lemma 2.3.1 is due to Pataki [1998]. For nondegeneracy in semidefinite programming we refer the reader to Alizadeh, Haeberly, and Overton [1997]. Grötschel, Lovász, and Schrijver [1988] proved that well behaved positive semidefinite programming problems can be solved (in approximate sense) in polynomial time. Example 2.3.2 is taken from Ramana [1997], who attributes the first example of this type to Khachiyan.

Chapter 3

Semidefinite Relaxations of Combinatorial Optimization Problems

In this chapter we study semidefinite relaxations of several combinatorial optimization problems and present some of their major properties. We start with the semidefinite relaxation of max-cut in Section 3.1, prove the approximation result of Goemans and Williamson, and show that this relaxation is asymptotically optimal for a large class of random graphs. In Section 3.2 we derive a semidefinite relaxation for quadratic 0-1 programming, which turns out to be a scaled version of the max-cut relaxation. We then consider two examples of constrained quadratic 0-1 programs, namely the independent set problem and the quadratic 0-1 knapsack problem. We conclude with Section 3.4 by surveying different modeling techniques used to obtain approximation algorithms for max-2sat, k-partitioning, coloring, and the betweenness problem.

3.1 Max-Cut

The max-cut problem is one of the standard *NP*-complete problems defined on graphs. Let $G = (V, E)$ denote an edge-weighted undirected graph without loops or multiple edges. We use $V = \{1, \ldots, n\}$, ij for an edge with endpoints i and j, and a_{ij} for the weight of an edge $ij \in E$. For $S \subseteq V$ the *cut* $\delta(S)$ is the set of edges $ij \in E$ that have one endpoint in S and the other in $V \setminus S$. The max-cut problem asks for the cut maximizing the sum of the weights of its edges. More formally, the max-cut problem can be written as follows,

$$\text{(MC)} \qquad \text{mc}(G) = \max_{S \subseteq V} \sum_{ij \in \delta(S)} a_{ij}. \tag{3.1}$$

Here, we only work with the complete graph K_n. In order to model an arbitrary graph in this setting, define $a_{ij} = 0$ for $ij \notin E$. $A = (a_{ij}) \in S_n$ is referred to as the weighted adjacency matrix of the graph.

An algebraic formulation can be obtained by introducing cut vectors $x \in \{-1, 1\}^n$ with $x_i = 1$ for $i \in S$ and $x_i = -1$ for $i \in V \setminus S$. Consider the following problem,

$$\max_{x \in \{-1,1\}^n} \sum_{i<j} a_{ij} \frac{1 - x_i x_j}{2}. \tag{3.2}$$

The value of the expression $(1 - x_i x_j)/2$ is 0 if $x_i = x_j$, *i.e.*, if i and j are in the same set, and 1 if $x_i = -x_j$. In other words $(1 - x_i x_j)/2$ yields the *incidence vector* of the cut associated with cut

vector x, evaluating to 1 if edge ij is in the cut. Therefore problem (3.2) is equivalent to (MC). We modify this formulation by exploiting the symmetry of A and $x_i x_i = 1$,

$$\begin{aligned}\frac{1}{2}\sum_{i<j} a_{ij}(1 - x_i x_j) &= \frac{1}{4}\sum_{i,j} a_{ij}(1 - x_i x_j) \\ &= \frac{1}{4}\sum_{i=1}^{n}\left(\sum_{j=1}^{n} a_{ij}x_i x_i - \sum_{j=1}^{n} a_{ij}x_i x_j\right) \\ &= \frac{1}{4}x^T(\mathrm{Diag}(Ae) - A)x.\end{aligned} \tag{3.3}$$

The matrix

$$L(G) = \mathrm{Diag}(Ae) - A$$

is called the *Laplace matrix* of the graph G. Usually we will drop the argument and speak of L only. For $C = \frac{1}{4}L$ the max-cut problem may be interpreted as a special case of the following more general problem,

$$\max_{x\in\{-1,1\}^n} x^T C x \tag{3.4}$$

for given $C \in S_n$. In order to derive a semidefinite relaxation of (3.4) we observe that

$$x^T C x = \langle Cx, x\rangle = \left\langle C, xx^T\right\rangle.$$

For $x \in \{-1,1\}^n$ the matrix xx^T is positive semidefinite, its diagonal entries are equal to 1, and it is a rank one matrix. Now consider the relaxation of xx^T to a matrix X satisfying these three characteristic properties,

$$\begin{array}{ll}\max & \frac{1}{4}\langle L, X\rangle \\ \text{s.t.} & \mathrm{diag}(X) = e \\ & X \succeq 0 \\ & \mathrm{rank}(X) = 1.\end{array} \tag{3.5}$$

Because of the rank one constraint this is not a semidefinite program. In fact, this is again problem (3.4).

Lemma 3.1.1 *(Laurent and Poljak [1995]) Problem (3.5) is equivalent to (3.4).*

Proof. Let X be feasible for (3.5). As $X \succeq 0$ with $\mathrm{rank}(X) = 1$, there is a factorization $x \in \mathbb{R}^n$ such that $X = xx^T$. But since $X_{ii} = 1 = x_i x_i$ for $i = 1,\ldots,n$ we have $x \in \{-1,1\}^n$. Conversely, we have observed above that for any $x \in \{-1,1\}^n$ xx^T is contained in the feasible set of (3.5). Therefore (3.5) and (3.4) optimize the same objective function over the same feasible set. ■

Dropping the rank one constraint yields a semidefinite programming relaxation of (3.4), and, for $C = \frac{1}{4}L$, of max-cut.

$$\text{(SMC)}\qquad \begin{array}{rl}\max & \langle C, X\rangle \\ \text{s.t.} & \mathrm{diag}(X) = e \\ & X \succeq 0\end{array} \qquad\qquad \begin{array}{rl}\min & e^T u \\ \text{s.t.} & C + Z - \mathrm{Diag}(u) = 0 \\ & Z \succeq 0.\end{array}$$

An illustration of the primal feasible set of (SMC) for $n = 3$ is given in Figure 3.1.

The semidefinite program is well behaved, because $X = I$ is strictly primal feasible and $u_i = 1 + \sum_{j=1}^{n} |C_{ij}|$ for $i = 1,\ldots n$, $Z = \mathrm{Diag}(u) - L$ yields a strictly dual feasible solution. (SMC) implies the box constraints $-1 \le x_{ij} \le 1$, because the principal two by two submatrices of a feasible X must be positive semidefinite. If, for a feasible X, $|x_{ij}| = 1$ for some indices $i \neq j$ then the nonnegativity of the determinant of the principal three by three submatrix with indices i, j, and $k \in \{1,\ldots,n\}\setminus\{i,j\}$ implies that $x_{ik} = \mathrm{sgn}(x_{ij})\cdot x_{jk}$. Together with Lemma 3.1.1 this proves that the matrices xx^T with $x \in \{-1,1\}^n$ are the only feasible matrices with ± 1 entries.

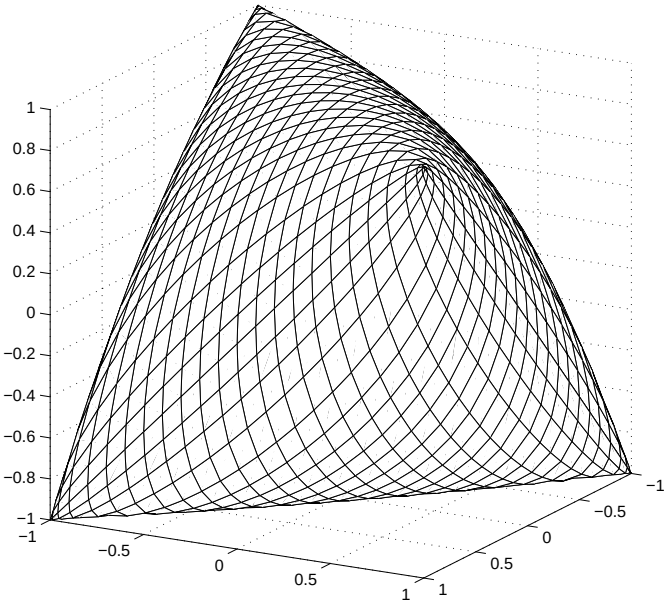

Figure 3.1 The primal feasible set of the semidefinite relaxation (SMC) of max-cut for $n = 3$. The surface is the solution of $\det \begin{bmatrix} 1 & x & y \\ x & 1 & z \\ y & z & 1 \end{bmatrix} = 0$.

Lemma 3.1.2 *(Laurent and Poljak [1995]) Let* $X \in \{X \succeq 0 : \operatorname{diag}(X) = e\}$. *Then* $-1 \leq x_{ij} \leq 1$. *If* $x_{ij} \in \{-1, 1\}$ *for all* ij *then* $X = xx^T$ *with* $x \in \{-1, 1\}^n$.

In other words, all vertices of the $[-1, 1]^{\binom{n}{2}}$ cube (associated with the upper triangle of X) that do not correspond to integral solutions of (3.4) are eliminated by this relaxation.

Goemans and Williamson [1995] provided an intriguing geometric interpretation of the feasible set of (SMC). Based on this interpretation they developed an approximation algorithm for max-cut whose performance guarantee is significantly better than that of previous algorithms. So far all other approximation algorithms employing semidefinite programming techniques build on their ideas. It is worth to study the approach in detail.

A feasible matrix X of (SMC) can be interpreted as the Gram matrix of vectors $v_i \in \mathbb{R}^n$, $i = 1, \ldots, n$. For any factorization of a feasible X into $V^T V$ with $V \in M_n$ the columns of V yield such vectors v_i. Because of $\operatorname{diag}(X) = e$ each vector is of unit length, $\|v_i\| = 1$. Associating vector v_i with node i we may interpret v_i as a relaxation of $x_i \in \{-1, 1\}$ to the n-dimensional unit sphere. The products $x_i x_j \in \{-1, 1\}$ are relaxed to $v_i^T v_j \in [-1, 1]$. Thus, formulating relaxation (SMC) in vector notation we obtain

$$\begin{array}{lll} \max & \sum_{i,j} c_{ij} v_i^T v_j & \\ \text{s.t.} & v_i^T v_i = 1 & \forall i \in V \\ & v_i \in \mathbb{R}^n & \forall i \in V. \end{array} \tag{3.6}$$

This vector formulation provides a clear intuition on how the semidefinite solution can be interpreted. Vectors v_i and v_j are unit vectors, $v_i^T v_j$ is the cosine of the angle enclosed by these vectors. If the angle between two vectors is large then we should separate the corresponding vertices, if it is small we should put them into the same set. However, we have to take care of conflicting configurations. In order to avoid conflicts consider the following alternative. Generate a random hyperplane through the origin and group all vectors on the same side of this hyperplane together. In particular the random hyperplane is constructed by a random vector h acting as the normal

vector of this hyperplane. The partition of V into $(S, V \setminus S)$ is formed by assigning all vertices i to S whose corresponding vectors i have positive inner product with h,

$$S(h) = \left\{ i \in \{1, \ldots, n\} : \ \operatorname{sgn}(v_i^T h) = 1 \right\}.$$

It is to be expected that vectors with a large angle will be separated because the random hyperplane is likely to end up between them. More formally, we have the following theorem.

Theorem 3.1.3 *(Goemans and Williamson [1995]) For n unit vectors $v_i \in \mathbb{R}^n$ and for a vector h randomly chosen from the standard normal distribution in $\mathbb{R}^n$, let*

$$H = \sum_{i,j} c_{ij} \operatorname{sgn}(v_i^T h) \operatorname{sgn}(v_j^T h).$$

Then the expected value of the integral solution is

$$E(H) = \frac{2}{\pi} \sum_{i,j} c_{ij} \arcsin(v_i^T v_j).$$

Proof. By the linearity of expectations

$$E(H) = \sum_{i,j} c_{ij} E(\operatorname{sgn}(v_i^T h) \operatorname{sgn}(v_j^T h)),$$

and $E(\operatorname{sgn}(v_i^T h) \operatorname{sgn}(v_j^T h)) = 1 - 2 \Pr(\operatorname{sgn}(v_i^T h) \neq \operatorname{sgn}(v_j^T h))$. The latter probability is equal to the probability that the projection of the random hyperplane onto the plane spanned by the two vectors v_i and v_j separates them. The projection of vector h onto this plane is again normally distributed. As the standard normal distribution is spherically symmetric the probability that the projected vector separates i and j is twice the angle between v_i and v_j divided by 2π.

$$E(\operatorname{sgn}(v_i^T h) \operatorname{sgn}(v_j^T h)) = 1 - \frac{2}{\pi} \arccos(v_i^T v_j) = \frac{2}{\pi} \arcsin(v_i^T v_j).$$

■

Remark 3.1.4 *Bertsimas and Ye [1998] provide an alternative interpretation of the rounding procedure that yields the same result: For a feasible X of (SMC) generate the vector h from a multivariate normal distribution with 0 mean and covariance matrix X and round h directly to a $\{-1, 1\}$-vector x by $x = \operatorname{sgn}(h)$ ($\operatorname{sgn}(\cdot)$ acting componentwise). The probability of edge ij to end up in the cut is again $\frac{2}{\pi} \arcsin(x_{ij})$.*

This result has a surprising consequence. Let $\arcsin(X)$ denote the matrix with entries $\arcsin(x_{ij})$.

Corollary 3.1.5 *(Goemans and Williamson [1995]) Problem (3.4) is equivalent to*

$$\begin{array}{ll} \max & \frac{2}{\pi} \langle C, \arcsin(X) \rangle \\ s.t. & \operatorname{diag}(X) = e \\ & X \succeq 0. \end{array} \tag{3.7}$$

Proof. (3.7) is a relaxation of (3.4), because for $X = xx^T$ with $x \in \{-1, 1\}^n$, X is feasible for (3.7) and the objective values of (3.7) and (3.4) coincide, $\frac{2}{\pi} \langle C, \arcsin(X) \rangle = \langle C, X \rangle = x^T C x$.

On the other hand let $X^* = V^T V$ denote an optimal solution of (3.7) and let $v_i^* = V_{\cdot,i}$. Then the expected value of an integral solution obtained via rounding these v_i^* by a random hyperplane is the optimal value of (3.7) by Theorem 3.1.3. Consequently there exists an $x \in \{-1, 1\}^n$ with at least this value. ■

For positive semidefinite cost matrices C we already obtain a reasonable bound.

Theorem 3.1.6 *(Nesterov [1998]) Let $C \succeq 0$, then*

$$\max_{x\in\{-1,1\}^n} x^T C x \geq \frac{2}{\pi} \max\left\{\langle C, X\rangle : \operatorname{diag}(X) = e,\ X \succeq 0\right\}.$$

Proof. For $2 \leq k \in \mathbb{N}$ define $X^{\circ k} = X \circ X^{\circ(k-1)}$ with $X^{\circ 1} = X$ and observe that the Taylor series expansion for $\arcsin(x_{ij})$ with $|x_{ij}| \leq 1$ implies

$$\arcsin(X) = X + \frac{1}{2}\frac{X^{\circ 3}}{3} + \frac{1\cdot 3}{2\cdot 4}\frac{X^{\circ 5}}{5} + \ldots$$

For feasible $X \succeq 0$ with $\operatorname{diag}(X) = e$ the matrix entries satisfy $-1 \leq x_{ij} \leq 1$. Therefore it follows form the Schur product Theorem 1.2.11 that $\arcsin(X) \succeq X$. Since $C \succeq 0$ Lemma 1.2.3 yields

$$\langle C, \arcsin(X)\rangle \geq \langle C, X\rangle. \tag{3.8}$$

The result now follows from Corollary 3.1.5. ■

With respect to the semidefinite relaxation of a max-cut problem with nonnegative edge weights Theorem 3.1.6 states that there is always a cut with value at least $\frac{2}{\pi}$ (> 0.6366) times the optimal value of the semidefinite relaxation, because the Laplace matrix $L(G)$ of a graph with nonnegative edge weights is positive semidefinite.

For special positive semidefinite matrices the inequality (3.8) can be sharpened. In particular, let $T_{ij}^+ \in S_n^+$ denote the matrix with $t_{ii} = t_{jj} = 1$ and $t_{ji} = t_{ij} = 1$ and which is zero otherwise, and let $T_{ij}^- \in S_n^+$ denote the matrix with $t_{ii} = t_{jj} = 1$ and $t_{ji} = t_{ij} = -1$ and which is zero otherwise. For example, for $n = 3$,

$$T_{12}^+ = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad T_{13}^- = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}.$$

Lemma 3.1.7 *Let $\mathcal{T} = \left\{T_{ij}^+, T_{ij}^- : i < j,\ i,j \in \{1,\ldots,n\}\right\}$. Then for all $T \in \mathcal{T}$ and for all $X \succeq 0$ with $\operatorname{diag}(X) = e$*

$$\langle T, \arcsin(X)\rangle \geq \beta \langle T, X\rangle$$

with $\beta = \min_{-1<\theta\leq 1} \frac{\frac{\pi}{2} + \arcsin(\theta)}{1+\theta} > 1.38$.

Proof. It suffices to consider the case $n = 2$. Let $T = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right]$ (the other case can be proved analogously) and let X be a feasible matrix. For $\langle T, X\rangle = 0$ the claim is true, therefore let $\langle T, X\rangle > 0$. Then

$$\frac{\langle T, \arcsin(X)\rangle}{\langle T, X\rangle} = \frac{\frac{\pi}{2} + \arcsin(x_{12})}{1 + x_{12}}$$

Minimizing this ratio over $-1 < x_{12} \leq 1$ yields the lemma. ■

Because of the linearity of the inner product the result extends to matrices that are nonnegative linear combinations of matrices from $\mathcal{T}$.

Observation 3.1.8 *The Laplace matrix of a nonnegatively weighted graph is in the cone $\mathcal{T}$.*

Putting Lemma 3.1.7 and Observation 3.1.8 together we obtain the result of Goemans and Williamson as a Corollary.

Corollary 3.1.9 *(Goemans and Williamson [1995]) Let $C \in \operatorname{cone}(\mathcal{T})$ and let m^* denote the optimal value of (3.4). Then for any feasible solution X of (SMC),*

$$m^* \geq E(H) \geq \alpha \langle C, X\rangle, \tag{3.9}$$

with $\alpha = \beta\frac{2}{\pi} > 0.87856$.

Observe, that (3.9) is true for any feasible solution of (SMC), in particular for an optimal solution X_*. Since (SMC) is a relaxation of (3.4), its objective value satisfies $\langle C, X_* \rangle \geq m^*$, so we obtain

$$\langle C, X_* \rangle \geq m^* \geq E(H) \geq \alpha \langle C, X_* \rangle . \tag{3.10}$$

Given the optimal solution X_*, the random hyperplane rounding scheme yields an integral vector $x \in \{-1, 1\}^n$ whose expected objective value $E(H)$ is within α of the maximum cut (ironically, the expected value of the cut gets closer to m^* as the bound $\langle C, X_* \rangle$ gets worse). Mahajan and Ramesh [1995] devised a deterministic version of the randomized rounding scheme that achieves this expected value for a given X in polynomial time.

In the Turing model of computation, we cannot hope to compute an optimal X_* exactly, but only an ε-approximation $\hat{X}$ of an optimal solution, see the discussion in Section 2.3. Since the primal feasible set of (SMC) is contained in the unit cube (Lemma 3.1.2), the ellipsoid method can be used to compute such an approximate solution in polynomial time (Grötschel, Lovász, and Schrijver [1988]). Therefore, Corollary 3.1.9 and the deterministic rounding method of Mahajan and Ramesh [1995] give rise to a polyniomal $(\alpha - \varepsilon) > 0.8785$ approximation algorithm for the max-cut problem (MC) with nonnegative edge weights. On the negative side, Håstad [1997] has shown that there is no polynomial approximation algorithm that achieves a constant of $\frac{16}{17} + \varepsilon < 0.9412$ unless $P{=}NP$, so the gap is moderate.

A natural question is, whether α also reflects the quality of the bound obtained by (SMC) or whether better constants exist for this purpose. The 5-cycle, for example, has $\frac{25+5\sqrt{5}}{32} \cdot 4$ as optimal value of (SMC) (Delorme and Poljak [1993]). Since the max-cut value of the 5-cycle is 4, the ratio max-cut to relaxation is $32/(25+5\sqrt{5}) < 0.8845$. Slightly stronger examples have been referenced in the literature (cf. Goemans [1997]). Therefore the constant cannot be improved significantly without improving on (SMC).

Remark 3.1.10 *Goemans and Williamson [1995] show that the rounding algorithm behaves even better if the the percentage of edges in the cut is relatively high, say, more than 85% of all edges are in the cut. In the opposite case, when the percentage of edges in the cut is smaller than 84.458%, then it pays off to introduce more uniform randomness into the rounding procedure (Zwick [1999]). In the interpretation of Remark 3.1.4, the covariance matrix is shifted from X towards the identity,* i.e., *pick $h \in N(0, \gamma I + (1-\gamma)X)$ for appropriate $\gamma \in [0, 1]$.*

Historically, the first bound equivalent to (SMC) was a spectral bound derived from (3.4). Indeed, (3.4) shows strong similarities to the Rayleigh-Ritz characterization of the maximal eigenvalue. As all feasible vectors x lie on a sphere of radius $\sqrt{n}$ it seems natural to relax $x \in \{-1, 1\}^n$ to $\|x\| = \sqrt{n}$, which yields an upper bound of $n\lambda_{\max}(C)$ (Mohar and Poljak [1990]),

$$\max_{x \in \{-1,1\}^n} x^T C x \leq \max_{\|x\| = \sqrt{n}} x^T C x = n \max_{\|x\|=1} x^T C x = n\lambda_{\max}(C)$$

In order to improve this bound observe that diagonal perturbations of the cost matrix in form of $C + \mathrm{Diag}(\bar{u})$ with $\bar{u} \in \mathbb{R}^n$ and $\bar{u}^T e = 0$ do not change the value of (3.4), because $x^T \mathrm{Diag}(\bar{u}) x = e^T \bar{u} = 0$ for all $x \in \{-1, 1\}^n$. These diagonal perturbations, however, have considerable influence on the maximal eigenvalue. Therefore a better bound is (Delorme and Poljak [1993])

$$\min_{\bar{u} \in \mathbb{R}^n,\ \bar{u}^T e = 0} n\lambda_{\max}(C + \mathrm{Diag}(\bar{u})). \tag{3.11}$$

It turns out that (3.11) and (SMC) are indeed the same.

Lemma 3.1.11 *(Poljak and Rendl [1995a]) (3.11) is equivalent to the dual of (SMC).*

Proof. We start by formulating (3.11) as a semidefinite program,

$$\begin{array}{ll} \min & n\lambda \\ \text{s.t.} & \lambda I - C - \mathrm{Diag}(\bar{u}) \succeq 0 \\ & \bar{u}^T e = 0. \end{array}$$

Rewrite $\lambda I - \mathrm{Diag}(\bar{u})$ as $\mathrm{Diag}(\lambda e - \bar{u})$ and use the substitution $u = \lambda e - \bar{u}$. This yields $\lambda = e^T u/n$ and $\bar{u} = e^T u \cdot e/n - u$, with $\bar{u}^T e = 0$ being satisfied generically. After this substitution the program reads

$$\begin{array}{ll} \min & e^T u \\ \text{s.t.} & -C + \mathrm{Diag}(u) \succeq 0. \end{array}$$

Setting $Z = -C + \mathrm{Diag}(u)$ completes the proof. ■

This spectral formulation of (SMC) gives rise to an extraordinary strong result for max-cut with respect to the class of random graphs with fixed edge probability. Let $G_{n,p}$ denote a random graph with n vertices where each edge is included with probability p. The adjacency matrix of this (unweighted) graph has entries $a_{ij} = 1$ if ij is an edge and $a_{ij} = 0$ if it is not. We will use the following result of Juhász [1981] which we state without proof.

Theorem 3.1.12 *(Juhász [1981]) For graphs $G_{n,p}$ with $0 < p < 1$ and for any $\varepsilon > 0$ the eigenvalues of the adjacency matrix A satisfy $\lambda_1(A) = pn + o(n^{\frac{1}{2}+\varepsilon})$ and $\max_{1<i\leq n} |\lambda_i(A)| = o(n^{\frac{1}{2}+\varepsilon})$ with probability tending to 1 as n tends to infinity.*

The theorem says, that for large n the maximal eigenvalue of the adjacency matrix is roughly of the size of the expected degree and all other eigenvalues are small in comparison. By means of this theorem we will now prove that the optimal solution of (SMC) is asymptotically optimal for $G_{n,p}$.

Theorem 3.1.13 *(Delorme and Poljak [1993]) For p fixed with $0 < p < 1$*

$$\lim_{n\to\infty} \frac{\min\limits_{\bar{u}\in\mathbb{R}^n,\ \bar{u}^T e=0} \frac{n}{4}\lambda_{\max}(L(G_{n,p}) + \mathrm{Diag}(\bar{u}))}{\mathrm{mc}(G_{n,p})} = 1.$$

Proof. Let d denote the average degree of $G_{n,p}$. Note, that a maximal cut will contain at least half of the edges, $\mathrm{mc}(G_{n,p}) \geq \frac{1}{4}nd$. To prove the result it suffices to choose $\hat{u} = de - Ae$. For this choice $L(G_{n,p}) + \mathrm{Diag}(\hat{u}) = dI - A$ and therefore, by Theorem 3.1.12,

$$\min_{\bar{u}\in\mathbb{R}^n,\ \bar{u}^T e=0} \frac{n}{4}\lambda_{\max}(L(G_{n,p}) + \mathrm{Diag}(\bar{u})) \leq \frac{n}{4}(d + o(n^{\frac{1}{2}+\varepsilon})).$$

Combining this upper bound with the lower bound for $\mathrm{mc}(G_{n,p})$ we obtain

$$\lim_{n\to\infty} \frac{\min\limits_{u\in\mathbb{R}^n,\ u^T e=0} \frac{n}{4}\lambda_{\max}(L(G_{n,p}) + \mathrm{Diag}(u))}{\mathrm{mc}(G_{n,p})} \leq \lim_{n\to\infty} \frac{\frac{1}{4}n(d + o(n^{\frac{1}{2}+\varepsilon}))}{\frac{1}{4}nd} = 1.$$

■

It is instructive to compare the semidefinite relaxation to the convex hull of the integral solutions. The feasible set of the semidefinite program (SMC) is a relaxation of the cut polytope[1]

$$P_C = \mathrm{conv}\left\{xx^T : \ x \in \{-1,1\}^n\right\}. \tag{3.12}$$

A general family of valid inequalities for P_C reads (Laurent and Poljak [1996a])

$$b^T X b \geq \min\left\{b^T xx^T b : x \in \{-1,1\}^n\right\} = \min\left\{(b^T x)^2 : x \in \{-1,1\}^n\right\}, \tag{3.13}$$

where $b \in \mathbb{Z}^n$. If the right hand side value is strictly positive, the corresponding inequality is called a *gap inequality*. Since the positive semidefiniteness of X implies $b^T X b \geq 0$, the addition of gap inequalities to (SMC) strengthens the relaxation. If $\min\left\{(b^T x)^2 : x \in \{-1,1\}^n\right\}$ is equal to one, the gap inequality is also called *hypermetric*. If $b \in \{-1,0,1\}^n$ consists of an odd number of

[1] Traditionally, the cut polytope is defined as the convex hull of the incidence vectors of all cuts in the graph. In our context, however, the definition given here is more convenient.

nonzero entries, hypermetric inequalities are called *clique inequalities*. Clique inequalities always define facets of P_C (Barahona, Grötschel, and Mahjoub [1985]). Clique inequalities with three nonzero elements specialize to the well known *triangle inequalities*,

$$\begin{array}{rcl} x_{ij} + x_{ik} + x_{jk} & \geq & -1 \\ x_{ij} - x_{ik} - x_{jk} & \geq & -1 \\ -x_{ij} + x_{ik} - x_{jk} & \geq & -1 \\ -x_{ij} - x_{ik} + x_{jk} & \geq & -1. \end{array} \tag{3.14}$$

The set of all points satisfying all triangle inequalities defines the so called *metric polytope*. It is conjectured by Deza, Laurent, and Poljak [1992] that among all valid inequalities for P_C, the triangle inequalities are those with the least distance to the center I of P_C; the conjecture is proved for all valid inequalities having coefficients in $\{-1, 0, 1\}$. Since there are $4\binom{n}{3}$ triangle inequalities one can optimize over the metric polytope in polynomial time. The metric polytope is exact for graphs not contractible to K_5 (Seymour [1981]; Barahona [1983]), this includes planar graphs. If we add the triangle inequalities to (SMC) then the bound is tight for the 5-cycle. With respect to nonnegatively weighted graphs, the worst example known for (SMC) with triangle inequalities is the antiweb AW_9^2 (a graph on 9 nodes with edges $\{i, i+1\}$ and $\{i, i+2\}$ modulo 9 for $i = 1, \ldots, 9$). The value of the relaxation is > 50.445 as opposed to the value of the maximal cut 48 (F. Rendl [personal communication]). This would yield a max-cut to relaxation ratio of approximately 0.9515.

Although the bound, determined by the optimal value of (SMC) combined with the triangle inequalities, seems to improve, the performance of the randomized algorithm does not. Karloff [1996] proved that for a class of highly symmetric graphs the optimal solution of (SMC) satisfies all triangle inequalities, and that its objective value coincides with the value of the optimal cut. Since the optimal solution of (SMC) is not influenced by the triangle inequalities, the expected value of the cuts generated by random hyperplanes is α times the value of the optimal cut.

3.2 Quadratic 0-1 Programming

Quadratic 0-1 programming refers to the problem

$$\text{(QP)} \qquad \max_{y \in \{0,1\}^n} y^T B y.$$

Since $y_i^2 = y_i$ for $y_i \in \{0, 1\}$, a linear term $b^T y$ can be modeled on the diagonal of B. In analogy to the max-cut case, we can construct a semidefinite relaxation by replacing yy^T with a semidefinite matrix Y whose diagonal elements are bounded by one. However, this relaxation turns out to be of poor quality. In order to arrive at a better relaxation, observed that the matrix $(y+v)(y+v)^T$ must be positive semidefinite for any vector $v \in \mathbb{R}^n$,

$$yy^T + vy^T + yv^T + vv^T \succeq 0 \qquad \forall v \in \mathbb{R}^n.$$

In this formulation we relax yy^T to $Y \in S_n^+$ and exploit that $\operatorname{diag}(yy^T) = y$ for all $y \in \{0, 1\}^n$,

$$\begin{array}{rcl} & & Y + v \operatorname{diag}(Y)^T + \operatorname{diag}(Y) v^T + vv^T \succeq 0 \qquad \forall v \in \mathbb{R}^n \\ \iff & & Y + (v + \operatorname{diag}(Y))(v + \operatorname{diag}(Y))^T - \operatorname{diag}(Y) \operatorname{diag}(Y)^T \succeq 0 \qquad \forall v \in \mathbb{R}^n. \end{array}$$

Since $(v + \operatorname{diag}(Y))(v + \operatorname{diag}(Y))^T$ is positive semidefinite, the condition is most restrictive if we choose $v = -\operatorname{diag}(Y)$, whereupon this matrix disappears. We end up with the following semidefinite relaxation for quadratic 0-1 programming,

$$\begin{array}{lll} (SQP) & \max & \langle B, Y \rangle \\ & \text{s.t.} & Y - \operatorname{diag}(Y) \operatorname{diag}(Y)^T \succeq 0. \end{array}$$

This relaxation is a semidefinite program, because by the Schur Complement Theorem 1.1.9

$$Y - \operatorname{diag}(Y)\operatorname{diag}(Y)^T \succeq 0 \qquad \Longleftrightarrow \qquad \bar{Y} = \begin{bmatrix} 1 & \operatorname{diag}(Y)^T \\ \operatorname{diag}(Y) & Y \end{bmatrix} \succeq 0. \tag{3.15}$$

$\bar{Y}$ may also be interpreted as the relaxation of the dyadic product of an extended 0-1 vector $\bar{y} = \left[1, y^T\right]^T$ (Lovász and Schrijver [1991]). The first component of $\bar{y}$ is referred to by index 0. Matrices of the form $\bar{y}\bar{y}^T$ are positive semidefinite, their first row and column is equal to their diagonal, and their rank is one.

Lemma 3.2.1 *The optimization problem*

$$\begin{array}{ll} \max & \langle B, Y \rangle \\ \text{s.t.} & \bar{Y} = \begin{bmatrix} 1 & \operatorname{diag}(Y)^T \\ \operatorname{diag}(Y) & Y \end{bmatrix} \succeq 0 \\ & \operatorname{rank}(\bar{Y}) = 1 \end{array} \tag{3.16}$$

is equivalent to (QP).

Proof. Similar to the proof of Lemma 3.1.1. ■

Like in max-cut, (SQP) can be interpreted as a relaxation obtained by dropping a rank one constraint. In analogy to the cut polytope P_C we define the boolean quadric polytope

$$P_B = \operatorname{conv}\left\{ \bar{y}\bar{y}^T : \bar{y} = \begin{bmatrix} 1 \\ y \end{bmatrix}, y \in \{0,1\}^n \right\}.$$

In contrast to max-cut, only part of the box constraints $0 \leq y_{ij} \leq 1$ are implied by (SQP). In particular the positive semidefiniteness of the two by two principal submatrices of $\bar{Y}$ with indices 0 and $1 \leq i \leq n$ implies $0 \leq y_{ii} \leq 1$. Since offdiagonal elements of positive semidefinite matrices may never be larger than the maximum of the corresponding two diagonal elements, the offdiagonal elements are bounded by 1 from above. Better bounds are determined via the determinant of the three by three principal submatrices of $\bar{Y}$ for indices 0 and $1 \leq i < j \leq n$,

$$\begin{array}{rcl} y_{ij} & \geq & y_{ii}y_{jj} - \sqrt{y_{ii}y_{jj}(1 + y_{ii}y_{jj} - y_{ii} - y_{jj})} \geq -\frac{1}{8} \\ y_{ij} & \leq & y_{ii}y_{jj} + \sqrt{y_{ii}y_{jj}(1 + y_{ii}y_{jj} - y_{ii} - y_{jj})} \leq 1. \end{array} \tag{3.17}$$

The bounds are sharp for $n = 3$. Observe, that the integrality of y_{ii} or y_{jj} implies $y_{ij} = y_{ii}y_{jj}$.

It turns out that the differences between the semidefinite relaxation of quadratic 0-1 programming and max-cut are superficial. In fact, (QP) in n variables and (3.4) in $n+1$ variables as well as the corresponding relaxations (SQP) and (SMC) are equivalent as we will see in the following.

To prove the equivalence of (3.4) and (QP) consider the affine transformation $y = \frac{1}{2}e + \frac{1}{2}x$ of a vector $x \in \{-1,1\}^n$ to a 0-1 vector $y \in \{0,1\}^n$. We homogenize this transformation by appending an additional component (with index 0) of value 1 to both vectors, $\bar{x} = \left[1, x^T\right]^T$ and $\bar{y} = \left[1, y^T\right]^T$, and obtain

$$\bar{y} = Q\bar{x} \quad \text{with} \quad Q = \begin{bmatrix} 1 & 0 \\ \frac{1}{2}e & \frac{1}{2}I_n \end{bmatrix}.$$

$\bar{x}$ corresponds to a cut vector with first component fixed to 1. We may interpret this as a normalized representation of cut vectors. This normalization does not affect the optimal value of (3.4), because $\bar{x}$ and $-\bar{x}$ induce the same cut (both yield the same matrix $\bar{x}\bar{x}^T$).

Since Q is invertible,

$$Q^{-1} = \begin{bmatrix} 1 & 0 \\ -e & 2I_n \end{bmatrix},$$

the transformation is bijective. In words, the linear transformation yields y_i equal to one if and only if x_i and x_0 belong to the same set of the partition, and y_i equal to zero if and only if x_i and x_0 belong to opposite sets.

For $\bar{y} = Q\bar{x}$ there is a one to one correspondence between the vertices $\bar{y}\bar{y}^T$ of the boolean quadric polytope P_B and the vertices $\bar{x}\bar{x}^T$ of the cut polytope P_C via the linear map $\bar{y}\bar{y}^T = Q\bar{x}\bar{x}^T Q^T$. Therefore these polytopes are isomorphic.

Lemma 3.2.2 *(De Simone [1989]) P_C for $n+1$ variables is isomorphic to $P_B = QP_CQ^T$ for n variables.*

Proof. By Section 2.1.2, the scaling is an automorphism on M_{n+1} and, as observed above, the vertices of P_C are mapped to vertices of P_B. ■

Likewise, this transformation establishes the equivalence of (SQP) and (SMC) via scaling by Q as descried in Section 2.1.2. In Proposition 2.1.3 the matrix X is the relaxation of $\bar{x}\bar{x}^T$ and W has to be replaced by $\bar{Y}$, the relaxation of $\bar{y}\bar{y}^T$.

Lemma 3.2.3 *Let $Q \in M_{n+1}$ be the matrix*

$$Q = \begin{bmatrix} 1 & 0 \\ \frac{1}{2}e & \frac{1}{2}I_n \end{bmatrix}.$$

Then $\bar{Y} = QXQ^T$ bijectively maps feasible solutions of (SMC) (for $n+1$ variables) to feasible solutions of (SQP).

Proof. Q is nonsingular, therefore X is positive definite if and only if QXQ^T is. The properties concerning the diagonals are verified by direct computation. ■

Moreover, the adjoint to the inverse of this transformation transforms constraints from the $\{-1,1\}$ representation into the $\{0,1\}$ representation without affecting the dual costs (see Proportion 2.1.3). From a computational point of view it is important to note that the transformation preserves the structure of the constraints (sparsity and low rank representations), as can be seen in the following example.

Example 3.2.4 *Consider the family of hypermetric inequalities $\langle \bar{b}\bar{b}^T, X\rangle = \bar{b}^T X \bar{b} \geq 1$, for $\bar{b} = \left[b_0, b^T\right]^T \in \mathbb{Z}^{n+1}$ with $\min\left\{(\bar{b}^T x)^2 : x \in \{-1,1\}^{n+1}\right\} = 1$; these are valid for P_C. To obtain the corresponding valid inequality for P_B we transform the coefficient matrix via*

$$Q^{-T}\bar{b}\bar{b}^T Q^{-1} = \begin{bmatrix} b_0 - e^T b \\ 2b \end{bmatrix} \begin{bmatrix} b_0 - e^T b & 2b^T \end{bmatrix}.$$

The constraint

$$\begin{bmatrix} b_0 - e^T b & 2b^T \end{bmatrix} \bar{Y} \begin{bmatrix} b_0 - e^T b \\ 2b \end{bmatrix} \geq 1$$

is valid for P_Q. In the special case of a triangle inequality with $b_0 = 1$, $b_i = 1$, $b_j = 1$, and all other $b_k = 0$ the inequality reads

$$1 - 4y_{0i} - 4y_{0j} + 4y_{ii} + 4y_{jj} + 4y_{ij} \geq 1.$$

Exploiting that $y_{0i} = y_{ii}$ and $y_{0j} = y_{jj}$ for feasible $\bar{Y}$ this inequality simplifies to $y_{ij} \geq 0$. Likewise we can transform and simplify all other triangle inequalities 3.14 and obtain

$$\begin{aligned} y_{ij} &\geq 0 && (3.18)\\ y_{ij} &\leq y_{ii} && (3.19)\\ y_{ii} + y_{jj} &\leq 1 + y_{ij} && (3.20)\\ y_{ik} + y_{jk} &\leq y_{kk} + y_{ij} && (3.21)\\ y_{ij} + y_{ik} + y_{jk} + 1 &\geq y_{ii} + y_{jj} + y_{kk} && (3.22) \end{aligned}$$

for $1 \leq i,j,k \leq n$, $i \neq j$, $i \neq k$, $j \neq k$. As the triangle inequalities define facets of P_C, inequalities (3.18) to (3.22) define facets of P_B. The inequalities (3.18), (3.19), and (3.20) form a well known linear relaxation of (QP) which is commonly referred to as the roof dual.

Table 3.1: Transformations between the $\{-1,1\}$ and the $\{0,1\}$ model.

MC $\to$ QP	QP $\to$ MC
$Q\bar{X}Q^T=\begin{bmatrix} y_0 & y^T \\ y & Y \end{bmatrix}$	$Q^{-1}\bar{Y}Q^{-T}=\begin{bmatrix} x_0 & x^T \\ x & X \end{bmatrix}$
$y_0=x_0$	$x_0=y_0$
$y=\frac{1}{2}(x+x_0e)$	$x=2y-y_0e$
$Y=\frac{1}{4}(X+xe^T+ex^T+x_0ee^T)$	$X=4Y-2y_0e^T-2ey_0^T+y_0ee^T$
$Q^{-T}\bar{A}Q^{-1}=\begin{bmatrix} b_0 & b^T \\ b & B \end{bmatrix}$	$Q^T\bar{B}Q=\begin{bmatrix} a_0 & a^T \\ a & A \end{bmatrix}$
$b_0=a_0-2e^Ta+e^TAe$	$a_0=b_0+e^Tb+\frac{1}{4}e^TBe$
$b=2(a-Ae)$	$a=\frac{1}{2}b+\frac{1}{4}Be$
$B=4A$	$A=\frac{1}{4}B$
$Q^{-T}\bar{v}=\begin{bmatrix} w_0 \\ w \end{bmatrix}$	$Q^T\bar{w}=\begin{bmatrix} v_0 \\ v \end{bmatrix}$
$w_0=v_0-e^Tv$	$v_0=w_0+\frac{1}{2}e^Tw$
$w=2v$	$v=\frac{1}{2}w$
$Q=\begin{bmatrix} 1 & 0 \\ \frac{1}{2}e & \frac{1}{2}I_n \end{bmatrix}$	$Q^{-1}=\begin{bmatrix} 1 & 0 \\ -e & 2I_n \end{bmatrix}$

For the convenience of the reader the transformations are provided in Table 3.1.

The results concerning the approximation algorithms for (3.4) are translated in the same way. In particular, Theorem 3.1.6 implies that there is a $(\frac{2}{\pi}-\varepsilon)$ approximation algorithm for (QP) with $B \succeq 0$. In the next example we construct the analogon of Corollary 3.1.9, which we will need for max-2sat.

Example 3.2.5 *To derive the analogon of Corollary 3.1.9 for quadratic 0-1 cost functions we transform the coefficient matrices of $\mathcal{T}$ into 0-1 representation. Let $\bar{v} \in \mathbb{R}^{n+1}$ with $\bar{v}_i = 1$, $\bar{v}_j = 1$ and zero otherwise, then $T_{ij}^+ = \bar{v}\bar{v}^T$. Likewise let $\bar{w} \in \mathbb{R}^{n+1}$ with $\bar{w}_i = 1$, $\bar{w}_j = -1$ and zero otherwise, then $T_{ij}^- = \bar{w}\bar{w}^T$. As in Example 3.2.4 these vectors transform to $Q^{-T}\bar{v}$ and $Q^{-T}\bar{w}$, and after simplification the corresponding terms read*

$$\begin{aligned}
\frac{1}{4}\langle T_{0i}^+, X\rangle &= \frac{1+x_{0i}}{2} \quad &\longleftrightarrow \quad & y_{ii} \\
\frac{1}{4}\langle T_{ij}^+, X\rangle &= \frac{1+x_{ij}}{2} \quad &\longleftrightarrow \quad & 1-y_{ii}-y_{jj}+2y_{ij} \\
\frac{1}{4}\langle T_{0i}^-, X\rangle &= \frac{1-x_{0i}}{2} \quad &\longleftrightarrow \quad & 1-y_{ii} \\
\frac{1}{4}\langle T_{ij}^-, X\rangle &= \frac{1-x_{ij}}{2} \quad &\longleftrightarrow \quad & y_{ii}+y_{jj}-2y_{ij}
\end{aligned} \tag{3.23}$$

for $i \neq j$ and $i, j \in \{1, \ldots, n\}$.

In consequence of this example we may translate Corollary 3.1.9 into the 0-1 setting as follows.

Corollary 3.2.6 *Let c^* be the optimal value of*

$$\begin{array}{ll} max & \langle \bar{C}, \bar{Y} \rangle \\ s.t. & \bar{Y} = \begin{bmatrix} 1 & \mathrm{diag}(Y)^T \\ \mathrm{diag}(Y) & Y \end{bmatrix} \succeq 0, \end{array}$$

with $\bar{C} \in S_{n+1}$ and $Y \in S_n$. If there exist $a_{ij}, b_{ij} \in \mathbb{R}^n_+$ for $i < j \in \{0, \ldots, n\}$ such that, for all feasible $\bar{Y}$,

$$\begin{aligned} \langle \bar{C}, \bar{Y} \rangle &= \sum_{j=1}^{n} (a_{0j} y_{ii} + b_{0j}(1 - y_{ii})) + \\ &\quad \sum_{0<i<j} (a_{ij}(1 - y_{ii} - y_{jj} + 2y_{ij}) + b_{ij}(y_{ii} + y_{jj} - 2y_{ij})) \end{aligned}$$

then there is an integral solution of value at least αc^* and it can be computed (in approximate sense) in polynomial time.

In Section 3.4.1 this corollary will yield a performance guarantee for a semidefinite relaxation of max-2sat.

Remark 3.2.7 *In the rounding procedure for max-cut (Theorem 3.1.3) no direction is favored in choosing the random hyperplane. This is reasonable, because any orthogonal transformation of the vectors will keep the angles between the vectors the same and there is no property that justifies directing more attention to a particular vertex. The situation is quite different in the case of transformed 0-1 programming problems. Here, the artificial vertex 0 is the decisive element. If a vector of vertex j points into roughly the same direction as the vector corresponding to vertex 0 then this is a strong indication that y_j should be set to 0. Feige and Goemans [1995] exploit this by rotating all vectors towards the straight line generated by the vector associated with vertex 0 and by rounding thereafter.*

3.3 Constrained Quadratic 0-1 Programming

We now turn to constrained quadratic 0-1 programming where constraints may be linear and/or quadratic,

$$\begin{array}{lll} \text{(CQP)} & \max & y^T B y \\ & \text{s.t.} & Ay \leq b \\ & & y^T A_i y \leq a_i \qquad i = 1, \ldots, k \\ & & y \in \{0,1\}^n, \end{array}$$

where $A \in M_{m,n}$, $b \in \mathbb{R}^m$, and $a_i \in \mathbb{R}$, $A_i \in S_n$ for $i = 1, \ldots, k$. This setting is very general, indeed, and many combinatorial problems can be expressed in this framework conveniently. In the following we study semidefinite relaxations of two particular problems of this class, the independent set and the quadratic 0-1 knapsack problem. This will illustrate how semidefinite relaxations can be constructed in a routine manner.

3.3.1 The Independent Set Problem

The independent set or stable set problem is defined in terms of an undirected graph $G = (V, E)$, $V = \{1, \ldots, n\}$, and asks for a maximal weighted subset $S \subseteq V$ of nodes with weights $w \in \mathbb{R}^n_+$ such that no two are connected by an edge from E,

$$\text{(IS)} \qquad \max \{|S| : \ S \subseteq V, \ ij \notin E \ \ \forall i, j \in S\}.$$

We derive a semidefinite relaxation by formulating (IS) as a constrained quadratic 0-1 programming problem. To this end we interpret $y \in \{0,1\}^n$ as a incidence vector of a subset of V. For any

incidence vector y of an independent set the product $y_i y_j$ must be zero for all edges $ij \in E$. This leads to the following reformulation of (IS),

$$\begin{array}{ll} \max & y^T \operatorname{Diag}(w) y \\ \text{s.t.} & y_i y_j = 0 \qquad \forall ij \in E \\ & y \in \{0,1\}^n . \end{array}$$

Note, that only quadratic constraints are present in this formulation. In line with the last section an obvious semidefinite relaxation of this problem is (Lovász and Schrijver [1991])

$$\begin{array}{lll} \text{(SIS)} & \max & \langle \operatorname{Diag}(w), Y \rangle \\ & \text{s.t.} & y_{ij} = 0 \qquad \forall ij \in E \\ & & Y - \operatorname{diag}(Y) \operatorname{diag}(Y)^T \succeq 0. \end{array}$$

It is not difficult to show that (SIS) has a strictly feasible point, so (SIS) can be solved (in an approximate sense) in polynomial time. It is, in fact, equivalent to the well known Lovász ϑ-function (Lovász [1979]). We cite it in its eigenvalue formulation,

$$\vartheta(G,w) = \min_{u \in \mathbb{R}^{|E|}} \lambda_{\max} \left(W + \sum_{ij \in E} u_{ij} E_{ij} \right), \tag{3.24}$$

where $W = \sqrt{w}\sqrt{w}^T$ (the square root being applied componentwise), and $E_{ij} \in S_n$ has a 1 in entries ij and ji and is zero otherwise.

Theorem 3.3.1 *(Lovász and Schrijver [1991]) The semidefinite programs (SIS) and* (3.24) *have the same optimal values.*

Proof. Observe, that (3.24) is equivalent to the dual of the primal-dual pair of semidefinite programs

$$\text{(SIP)} \quad \begin{array}{ll} \max & \langle W, X \rangle \\ \text{s.t.} & \operatorname{tr} X = 1 \\ & x_{ij} = 0 \qquad \forall ij \in E \\ & X \succeq 0 \end{array} \qquad\qquad \text{(SID)} \quad \begin{array}{ll} \min & \lambda \\ \text{s.t.} & \lambda I \succeq W + \sum_{ij \in E} u_{ij} E_{ij} \\ & \lambda \in \mathbb{R}, u \in \mathbb{R}^{|E|}. \end{array}$$

For (SIP), $X = I/n$ is a strictly feasible solution, and for (SID) choosing λ large enough yields a strictly feasible dual solution. By the strong duality Theorem 2.2.5, the optimal objective values of both problems coincide and are attained. We can therefore prove the theorem by constructing for any feasible solution Y of (SIS) a solution of (SIP) with greater or equal objective value, and vice versa for feasible solutions of (SIP).

Let $\bar{Y}$ (see (3.15)) be a feasible solution of (SIS). Since $\bar{Y} \succeq 0$, we may factorize it into $\bar{Y} = \bar{V}^T \bar{V}$ with $\bar{V} = [d_0 v_0, \ldots, d_n v_n]$, where the $v_i \in \mathbb{R}^{n+1}$ are unit vectors and $d_i \in \mathbb{R}$ for $i = 0$ to n. Then $\bar{Y}_{ij} = d_i d_j v_i^T v_j$. Since $\bar{Y}_{00} = 1$ and $\bar{Y}_{0i} = \bar{Y}_{ii}$ for $i = 1 \ldots n$, we obtain $d_0 = 1$ and $d_i = v_0^T v_i$ for $i = 1 \ldots n$ (exploit $v_i^T v_i = 1$). The objective value of this solution is $\theta = \sum_{i=1}^n w_i (v_0^T v_i)^2$. We claim that scaling the principal submatrix Y of $\bar{Y}$ by $S = \operatorname{Diag}(\sqrt{w/\theta})$ yields a feasible solution of (SIP) with greater or equal objective value. Indeed, the matrix SYS has the same zero pattern as Y and is positive semidefinite, since Y is. Furthermore,

$$\operatorname{tr}(SYS) = \sum_{i=1}^n \frac{w_i}{\theta} (v_0^T v_i)^2 v_i^T v_i = \frac{\sum_{i=1}^n w_i (v_0^T v_i)^2}{\theta} = 1.$$

The objective value of (SIP) reads $\sqrt{w}^T SYS \sqrt{w} = w^T Y w/\theta = w^T V^T V w/\theta$. Let $\hat{w} = Vw = \sum_{i=1}^n w_i (v_0^T v_i) v_i$. Then $v_0 v_0^T \preceq I_n$ and $\hat{w}^T v_0 = \theta$ imply $\hat{w}^T I \hat{w} \geq \hat{w}^T v_0 v_0^T w = \theta^2$. Thus, $\sqrt{w}^T SYS \sqrt{w} \geq \theta = \langle \operatorname{Diag}(w), Y \rangle$.

Now, let X be a feasible solution of (SIP) with $\theta = \sqrt{w}^T X \sqrt{w}$. Factorize $X = V^T V$ with $V = [d_1 v_1, \ldots, d_n v_n]$ where $v_i \in \mathbb{R}^n$ are unit vectors and $d_i \in \mathbb{R}_+$ for $i = 1, \ldots, n$. Define $v_0 = V\sqrt{w}/\sqrt{\theta}$, which is a unit vector, and $\bar{V} = [v_0, V]$. For the scaling matrix $S = \mathrm{Diag}(\bar{V}^T v_0)$ it is not difficult to check that $\bar{Y} = S\bar{V}^T\bar{V}S \succeq 0$ is feasible for (SIS) (use $x_{ij} = v_i^T v_j = 0$ for $ij \in E$ and $v_i^T v_i = 1$ for $i = 0, \ldots, n$ in verifying the constraints $\bar{Y}_{0i} = \bar{Y}_{ii}$ for $i = 1, \ldots, n$). In order to prove that the objective value of this solution is greater or equal θ, observe that

$$\sum_{i=1}^n \sqrt{w_i} d_i v_i^T v_0 = \sum_{i=1}^n \sqrt{w_i} e_i^T V^T V \sqrt{w} \frac{1}{\sqrt{\theta}} = \frac{1}{\sqrt{\theta}} \sqrt{w}^T X \sqrt{w} = \sqrt{\theta}.$$

Using the Cauchy-Schwarz inequality and $\sum_{i=1}^n d_i^2 = \mathrm{tr}\, X = 1$, we obtain

$$\begin{aligned} \langle W, X \rangle = \theta &= \left(\sum_{i=1}^n d_i \sqrt{w_i} v_i^T v_0 \right)^2 \\ &\leq \left(\sum_{i=1}^n d_i^2 \right) \left(\sum_{i=1}^n w_i (v_0^T v_i)^2 \right) \\ &= \sum_{i=1}^n w_i \bar{Y}_{ii} = \langle \mathrm{Diag}(w), Y \rangle . \end{aligned}$$

This completes the proof. ■

The constraints $y_{ij} \geq 0$ define facets of the boolean quadric polytope P_Q. The underlying combinatorial polyhedron of (SIS) is the convex hull of all vertices of P_Q that are contained in the intersection of the facets $y_{ij} = 0$ for all $ij \in E$. Thus this polyhedron is a face of P_Q. The polyhedron usually associated with the stable set problem is the projection of this face on the diagonal variables,

$$P_{S(G)} = \mathrm{conv}\left\{ y \in \{0,1\}^n : \; y_i y_j = 0 \;\; \forall ij \in E \right\}.$$

An important class of valid inequalities of $P_{S(G)}$ are the so called *clique inequalities* (they should not be mixed up with the clique inequalities of max-cut). Clearly, an independent set of G may contain at most one node from each clique in G. Let $(C, E(C))$ denote a clique in G and let c denote the incidence vector of its node set, then the clique inequality

$$c^T y \leq 1$$

is satisfied for all $y \in P_{S(G)}$. The polyhedron of nonnegative points satisfying all clique inequalities is known as

$$P_{SC(G)} = \left\{ y \in \mathbb{R}^n_+ : c^T y \leq 1, \forall\, (C, E(C)) \text{ clique in } G \right\}.$$

Optimizing over $P_{SC(G)}$ is *NP*-hard (Theorem 9.2.9 in Grötschel, Lovász, and Schrijver [1988]). The more it is amazing that all clique inequalities are satisfied on the diagonal of the feasible set of (SIS).

Lemma 3.3.2 *(Grötschel, Lovász, and Schrijver [1988])*
$P_{S(G)} \subseteq \left\{ \mathrm{diag}(Y) : \; y_{ij} = 0 \;\; \forall ij \in E, Y - \mathrm{diag}(Y)\,\mathrm{diag}(Y)^T \succeq 0 \right\} \subseteq P_{SC(G)}$

Proof. The left hand side inclusion is clear by construction. In order to prove the right hand side inclusion let C be a clique in G and let c denote its incidence vector. Define a vector $\bar{c} = \left[-1, c^T\right]^T$ and let Y be a feasible matrix of (SIS). Exploiting (3.15) we obtain

$$0 \leq \bar{c}^T \bar{Y} \bar{c} = 1 - 2c^T \,\mathrm{diag}(Y) + c^T Y c = 1 - c^T \,\mathrm{diag}(Y),$$

because $c^T Y c = c^T \,\mathrm{diag}(Y)$ due to the feasibility of Y. ■

Graphs satisfying $P_{S(G)} = P_{SC(G)}$ are called *perfect*. This definition is equivalent to the original definition given by Berge [1961] which calls a graph G perfect if the chromatic number is equal

to the size of the largest clique for all induced subgraphs of G (for a proof and further properties of perfect graphs, the stable set problem, and the ϑ-function see Grötschel, Lovász, and Schrijver [1988], Section 9). Since by Lemma 3.3.2 the feasible set of the semidefinite relaxation is a subset of $P_{SC(G)}$ it follows that the independent set problem can be solved (in approximate sense) for perfect graphs in polynomial time.

3.3.2 The Quadratic 0-1 Knapsack Problem

The quadratic 0-1 knapsack problem is quadratic 0-1 programming with one linear constraint,

$$\begin{array}{lll} & \max & y^T B y \\ \text{(QK)} & \text{s.t.} & a^T y \leq b \\ & & y \in \{0,1\}^n . \end{array}$$

Without loss of generality we may assume that $0 \leq a_i \leq b$ for $i = 1, \ldots, n$. For $a_i < 0$ we can flip y_i to $1 - y_i$, for $a_i \in (b, \infty)$ we can set the corresponding variable to zero. In addition, we assume that $a_i < b$, since $a_i = b$ allows to decompose the problem.

The crucial step in the design of a semidefinite relaxation for (QK) is the representation of the linear constraint within quadratic space. We will present four different approaches and compare their quality.

A natural first approach is to model the constraint on the diagonal, which is possible because $y_i = y_i^2$,

$$\begin{array}{lll} \text{(SQK1)} & \max & \langle B, Y \rangle \\ & \text{s.t.} & \langle \text{Diag}(a), Y \rangle \leq b \\ & & Y - \text{diag}(Y)\,\text{diag}(Y)^T \succeq 0. \end{array}$$

We call this the *diagonal representation*.

A second representation exploits the following observation. If $|a^T y| \leq b$ for all integral solutions then $a^T y y^T a \leq b^2$ is a valid inequality, as well. Replacing yy^T by Y we obtain the *square representation* (Helmberg, Rendl, and Weismantel [2000]),

$$\begin{array}{lll} \text{(SQK2)} & \max & \langle B, Y \rangle \\ & \text{s.t.} & \langle aa^T, Y \rangle \leq b^2 \\ & & Y - \text{diag}(Y)\,\text{diag}(Y)^T \succeq 0. \end{array}$$

In the third representation the inequality $a^T y \leq b$ is multiplied on both sides by $a^T y$. This is feasible because $a^T y \geq 0$. Now,

$$0 \leq a^T y (b - a^T y) = \begin{bmatrix} 0 \\ a \end{bmatrix}^T \begin{bmatrix} 1 \\ y \end{bmatrix} \begin{bmatrix} 1 \\ y \end{bmatrix}^T \begin{bmatrix} b \\ -a \end{bmatrix}$$

Relaxing the middle dyadic product to $\bar{Y}$, the corresponding relaxation reads (Bauvin and Goemans [personal communication 1999])

$$\begin{array}{lll} \text{(SQK3)} & \max & \langle B, Y \rangle \\ & \text{s.t.} & \left\langle \begin{bmatrix} b \\ -a \end{bmatrix} \begin{bmatrix} 0 \\ a \end{bmatrix}^T, \bar{Y} \right\rangle \geq 0 \\ & & Y - \text{diag}(Y)\,\text{diag}(Y)^T \succeq 0. \end{array}$$

Against our standard assumption that coefficient matrices should be symmetric we prefer not to symmetrize this coefficient matrix in order to keep notation simple.

The last approach may be interpreted as a particular case of a more general technique. The multiplication of any two valid nonnegative inequalities yields a valid quadratic inequality (Balas [1975]; Sherali and Adams [1990]; Lovász and Schrijver [1991]). In 0-1 programming the standard

candidates for multiplication with an arbitrary inequality $b - a^T y \geq 0$ (no restrictions on the coefficients are required in this case) are the box constraints $y_i \geq 0$ and $1 - y_i \geq 0$,

$$y_i a^T y \leq b y_i \qquad \text{and} \qquad (1 - y_i) a^T y \leq b(1 - y_i). \tag{3.25}$$

Replacing the mixed terms $y_i y_j$ by y_{ij} and the linear terms y_j by y_{jj} yields inequalities that we refer to as *y_i-representation* or *$(1 - y_i)$-representation* of $a^T x \leq b$. Using these representations we can form a fourth relaxation. To improve on (SQK3) it suffices to include the y_i-representations,

$$\begin{array}{lll} \text{(SQK4)} & \max & \langle B, Y \rangle \\ & \text{s.t.} & \sum_{j=1}^n a_j y_{ij} \leq b y_{ii} \qquad i = 1 \ldots n \\ & & Y - \operatorname{diag}(Y) \operatorname{diag}(Y)^T \succeq 0. \end{array}$$

There is a clear hierarchy between relaxations (SQK1) to (SQK4).

Lemma 3.3.3 *(Helmberg, Rendl, and Weismantel [2000], Bauvin and Goemans [personal communication 1999]) Denote by $\mathcal{Y}_1$, $\mathcal{Y}_2$, $\mathcal{Y}_3$, and $\mathcal{Y}_4$ the feasible sets of (SQK1), (SQK2), (SQK3), and (SQK4), respectively. Then $\mathcal{Y}_1 \supseteq \mathcal{Y}_2 \supseteq \mathcal{Y}_3 \supseteq \mathcal{Y}_4$.*

Proof. We first prove that a feasible point $Y \in \mathcal{Y}_2$ of (SQK2) is feasible for (SQK1). To this end we introduce the positive semidefinite matrix $Z = Y - \operatorname{diag}(Y) \operatorname{diag}(Y)^T$. Since

$$b^2 \geq a^T Y a = a^T Z a + (a^T \operatorname{diag}(Y))^2,$$

and Z is positive semidefinite by the feasibility of Y it follows that $(a^T \operatorname{diag}(Y))^2 \leq b^2$. This proves $\mathcal{Y}_1 \supseteq \mathcal{Y}_2$.

Next let $Y \in \mathcal{Y}_3$. Using $Z = Y - \operatorname{diag}(Y) \operatorname{diag}(Y)^T \succeq 0$ once more, we see that

$$\begin{aligned} 0 \leq \left\langle \begin{bmatrix} b \\ -a \end{bmatrix} \begin{bmatrix} 0 \\ a \end{bmatrix}^T, \bar{Y} \right\rangle &= b a^T \operatorname{diag}(Y) - a^T Y a \\ &= b a^T \operatorname{diag}(Y) - a^T Z a - a^T \operatorname{diag}(Y) \operatorname{diag}(Y)^T a \\ &= [b - a^T \operatorname{diag}(Y)] a^T \operatorname{diag}(Y) - a^T Z a. \end{aligned}$$

Because of $-a^T Z a \leq 0$ we obtain $b \geq a^T \operatorname{diag}(Y)$ and therefore $b^2 \geq a^T Y a$ in the first row of the equation above. Hence, $\mathcal{Y}_2 \supseteq \mathcal{Y}_3$.

Finally, multiplying each y_i-representation of (SQK4) with a_i ($a_i \geq 0$ by assumption) and summing over all i yields

$$\sum_{i=1}^n \sum_{j=1}^n a_i a_j y_{ij} = \langle a a^T, Y \rangle \leq \sum_{i=1}^n b a_i y_{ii} = b \cdot a^T \operatorname{diag}(Y).$$

This is exactly the inequality of (SQK3), and so $\mathcal{Y}_3 \supseteq \mathcal{Y}_4$. ∎

If there exist items i and j with $a_i + a_j > b$ then it is advisable to include the constraint $y_{ij} = 0$. These constraints improve the relaxation and guarantee that the dimension of the relaxed set is equal to the dimension of the underlying combinatorial polyhedron. Again, there is no difficulty in constructing a strictly feasible solution in the relative interior of the underlying combinatorial polyhedron.

Example 3.3.4 *To illustrate that the gap between the single inequality relaxations (SQK1) to (SQK3) may indeed be quite large consider the following problem for $k \leq n$,*

$$\begin{array}{ll} \max & y^T (e e^T - I) y \\ \text{s.t.} & e^T y \leq k \\ & y \in \{0,1\}^n. \end{array} \tag{3.26}$$

The optimal value of (3.26) is $k(k-1)$. Since the problem is invariant under permutations of the indices, in any quadratic relaxation there exists an optimal solution such that every diagonal element has the same value d and every off-diagonal element has the same value f, i.e., *there is an optimal solution of the form $Y = d \cdot I + f \cdot (ee^T - I)$. Thus, it suffices to compute the maximal possible f for relaxations (SQK1) to (SQK3).*

First consider (SQK1). The diagonal representation of $e^T y \leq k$ is $e^T \operatorname{diag}(Y) \leq k$ which implies $d \leq \frac{k}{n}$. Because of the positive semidefiniteness of Y we certainly cannot choose f larger than d. For $f = d = \frac{k}{n}$ we have $Y - \operatorname{diag}(Y)\operatorname{diag}(Y)^T = (f - f^2)ee^T \succeq 0$ and so this choice of f is optimal. The optimal value of (SQK1) for (3.26) is $(n-1)k$. Asymptotically, for $n \to \infty$, this bound is arbitrarily bad.

For (SQK2) the quadratic representation reads $e^T Y e \leq k^2$. Again the largest possible f is obtained by choosing $f = d$ and $e^T Y e = n^2 f = k^2$. This leads to an optimal value of $\frac{n-1}{n}k^2$ and, for $k = 2$, to an asymptotic error factor of 2.

In (SQK3) the inequality requires that $n(n-1)f + nd \leq knd$, or equivalently $f \leq \frac{(k-1)}{(n-1)}d$. The maximal value of d is bounded by $nd \leq k$, see the proof of Lemma 3.3.3. The values $d = k/n$ and $f = \frac{(k-1)k}{(n-1)n}$ can be attained and yield the optimal value $k(k-1)$ which is exact.

It is also instructive to compare the semidefinite relaxations to the standard linear relaxation in the special case of a linear cost function $C = \operatorname{Diag}(c)$, *i.e.*, $C_{ij} = 0$ for $i \neq j$. The natural linear relaxation reads (here, we may assume that, in addition, $a_i > 0$ for $i = 1, \ldots, n$)

$$\begin{array}{lll} \text{(LK)} \qquad \max & c^T y & \\ \text{s.t.} & a^T y \leq b & \\ & 0 \leq y_i \leq 1 & i = 1, \ldots, n. \end{array}$$

The optimal solution to (LK) corresponds to the solution of a greedy algorithm that includes (even fractional) items with best profit to weight ratio first. (SQK1) is equivalent to (LK), because for any feasible y vector there is a feasible matrix Y having y as its diagonal. However, this is not true for (SQK2).

Lemma 3.3.5 *(Helmberg, Rendl, and Weismantel [2000])*
Let Y^* be an optimal solution of (SQK2) for $C = \operatorname{Diag}(c)$. If (LK) has a unique optimal solution y^* which is not integral then $\operatorname{tr}(Y^*C) < c^T y^*$.

Proof. Assume that y^* is a unique, non integral, optimal solution of (LK). Since y^* is not integral, it must satisfy $a^T y^* = b$, and, by uniqueness, it has exactly one element y_i with $0 < y_i < 1$. Consider a matrix Y satisfying $\operatorname{diag}(Y) = y^*$ and $Y - \operatorname{diag}(Y)\operatorname{diag}(Y)^T \succeq 0$. By (3.17) $y_{ij} = y_{ii}y_{jj}$ for $i \neq j$, because at least one of y_{ii} and y_{jj} is either one or zero. Therefore the only non zero element of $Z = Y - \operatorname{diag}(Y)\operatorname{diag}(Y)^T$ is $z_{ii} = y_i - y_i^2$. Obviously, $a^T Z a = a_i^2 z_{ii}$ is strictly greater than zero. Thus, $a^T Y a > (a^T y^*)^2 = b^2$ and Y is not feasible for (SQK2). Finally, the fact that $y^* = \operatorname{diag}(Y^*)$ is feasible for (LK) completes the proof. ■

If the optimal solution of the linear program is not unique then it can be worked out that there exists an optimal y that is the diagonal of a feasible Y of (SQK2) and (SQK3). In theory, Lemma 3.3.5 indicates that for many linear 0/1 programming problems the canonical semidefinite relaxation yields a slightly tighter relaxation then its linear counterpart. In practice, we expect improvements to be far too lenient in comparison to the additional cost involved in solving a semidefinite relaxation.

3.4 Modeling Techniques

In the following we discuss several other modeling approaches that appear in the literature. We will start with extensions of the approximation algorithm of Goemans and Williamson to max-2sat and continue with modeling approaches for max k-cut and coloring. The last application is an approximation algorithm for the betweenness problem that uses a geometric interpretation of semidefinite programming.

3.4.1 Max-2Sat

An instance of max-2sat consists of a number of boolean variables $\{z_1, \ldots, z_n\}$ and a collection of clauses. Each clause is the disjunction of at most two literals, where a literal is either a variable z_i or its negation $\bar{z}_i$. The task is to find a truth assignment for the variables that maximizes the number of satisfied clauses, *i.e.*, the corresponding boolean expressions evaluate to true. The following extension of the approximation algorithm for max-cut to max-2sat was described in Goemans and Williamson [1995].

We first formulate the clauses as cost coefficients for a quadratic 0-1 programming problem. To this end we introduce a vector $y \in \{0,1\}^n$. Each component y_i corresponds to a boolean variable z_i, $y_i = 1$ corresponds to z_i being true, and $y_i = 0$ to z_i being false. The clauses transform to

$$\begin{array}{rclcrcl} z_i \vee z_j & = & \overline{\bar{z}_i \wedge \bar{z}_j} & \rightarrow & 1-(1-y_i)(1-y_j) & = & y_i + y_j - y_i y_j \\ \bar{z}_i \vee z_j & = & \overline{z_i \wedge \bar{z}_j} & \rightarrow & 1-y_i(1-y_j) & = & 1 - y_i + y_i y_j \\ \bar{z}_i \vee \bar{z}_j & = & \overline{z_i \wedge z_j} & \rightarrow & 1-y_i y_j, & & \end{array} \tag{3.27}$$

having value one if the corresponding clause is true and zero otherwise. The cost function of the (extended) quadratic 0-1 problem is formed by the sum of all the terms corresponding to clauses. In order to obtain the performance guarantee provided by Corollary 3.2.6 we have to make sure that the cost function belongs to the class required by Corollary 3.2.6, *i.e.*, we have to express the clauses as nonnegative linear combination of the terms in (3.23). Indeed, all clauses turn out to be of the desired form,

$$\begin{aligned} y_{ii} + y_{jj} - y_{ij} &= \frac{1}{2}((y_{ii}) + (y_{jj}) + (y_{ii} + y_{jj} - 2y_{ij})) \\ 1 - y_{ii} + y_{ij} &= \frac{1}{2}((y_{jj}) + (1 - y_{ii} - y_{jj} + 2y_{ij}) + (1 - y_{ii})) \\ 1 - y_{ij} &= \frac{1}{2}((1 - y_{ii}) + (1 - y_{jj}) + (y_{ii} + y_{jj} - 2_{y_{ij}})). \end{aligned}$$

Therefore we may apply Corollary 3.2.6 and obtain an $(\alpha - \varepsilon)$-approximation algorithm for max-2sat.

3.4.2 k-Partitioning

The approach to label each vertex of the graph with a vector and to express relations between the vertices in terms of the inner product of their associated vector labelings is a powerful tool for modeling semidefinite relaxations. The concept first appears in Lovász [1979], where orthonormal labelings are used to introduce the Lovász ϑ-function. To illustrate the importance of this technique we will sketch a natural generalization of the Goemans-Williamson approximation algorithm for partitioning a graph into several components which was introduced independently by Frieze and Jerrum [1995] and Karger, Motwani, and Sudan [1994].

In the max-cut case, we would like to match all vectors v_i either with the vector e^1 or with its opposite, $-e^1$ (instead of e^1 we could choose any other unit vector, since rotations of the whole system do not change the angle $v_i^T v_j$ between the vectors). Vertices whose associated vectors are close to e^1 form one set of the partition, vertices whose vectors are close to $-e^1$ form the other.

We can take a similar approach for partitioning the vertices into three sets by trying to match the vertices to three distinct unit vectors. In order to discern the three positions easily we would like them to point as far apart as possible, *i.e.*, the angles between the three vectors should be as large as possible. Vectors satisfying this requirement are coplanar and their pairwise scalar product is $-\frac{1}{2}$.

For partitioning into k sets we would like to find k unit vectors pointing as far apart as possible. In the next lemma we prove that, for k unit vectors with all pairs of vectors enclosing an angle of the same size, the value $-1/(k-1)$ is the smallest possible inner product (the largest angle) between all pairs of vectors.

Lemma 3.4.1 *(Frieze and Jerrum [1995]) Let $w_1, \ldots, w_k$ be unit vectors from $\mathbb{R}^n$ with $1 \leq k-1 \leq n$ and let $w_i^T w_j = \delta$ for $i \neq j$, $i, j \in \{1, \ldots, k\}$. Then $\delta \geq -\frac{1}{k-1}$. Furthermore, if $\delta = -\frac{1}{k-1}$ the vectors $w_1, \ldots, w_k$ are linearly dependent.*

Proof. $0 \leq (w_1 + \ldots + w_k)^T (w_1 + \ldots + w_k) = k + k(k-1)\delta$, therefore $\delta \geq -\frac{1}{k-1}$. For $\delta = -\frac{1}{k-1}$ the norm of the vector $\sum_{i=1}^k w_i$ is zero, thus the vectors are linearly dependent. ■

This value can be attained.

Lemma 3.4.2 *(Karger, Motwani, and Sudan [1994]; Frieze and Jerrum [1995]) For $k \geq 2$, there exist k unit vectors $w_1, \ldots w_k \in \mathbb{R}^{k-1}$ with $w_i^T w_j = -\frac{1}{k-1}$ for $i \neq j$, $i, j \in \{1, \ldots, k\}$.*

Proof. The matrix $A = \frac{k}{k-1} I_k - \frac{1}{k-1} ee^T$ is positive semidefinite with $\operatorname{diag}(A) = e$ and e is the only eigenvector to the eigenvalue 0. Therefore a factorization of A into WW^T with $W \in M_{k,k-1}$ is possible. The rows of W yield the desired vectors. ■

Geometrically, the endpoints of the vectors form an equilateral $k-1$ dimensional simplex with barycenter coinciding with the origin.

For a given undirected graph $G = (V, E)$ max k-cut asks for a partition of the vertex set $V = \{1, \ldots, n\}$ into at most k subsets such that as many edges as possible run between vertices in distinct sets. In order to design a semidefinite relaxation for this problem we label each vertex i with a unit vector $v_i \in \mathbb{R}^n$. In the "discrete" setting the v_i are restricted to the set $\{w_1, \ldots, w_k\}$ described above. The product $v_i^T v_j$ may attain only two values, either 1, if vertices i and j belong to the same set, or $-\frac{1}{k-1}$ if they belong to distinct sets. With edge weights a_{ij} the "integral" formulation of max k-cut reads

$$\text{(MkC)} \qquad \begin{array}{ll} \max & \sum_{i<j} a_{ij} \frac{(k-1)(1-v_i^T v_j)}{k} \\ \text{s.t.} & v_i \in \{w_1, \ldots, w_k\} \qquad i = 1, \ldots, n. \end{array}$$

The natural semidefinite relaxation yielding an upper bound on max k-cut is

$$\text{(SMkC)} \qquad \begin{array}{ll} \max & \sum_{i<j} a_{ij} \frac{(k-1)(1-v_i^T v_j)}{k} \\ \text{s.t.} & \|v_i\| = 1 \qquad i = 1, \ldots, n \\ & v_i^T v_j \geq -\frac{1}{k-1} \quad i < j. \end{array}$$

Alternatively, one can use the Laplace matrix of G to formulate the same problem in matrix notation,

$$\begin{array}{ll} \max & \frac{k-1}{2k} \langle L, X \rangle \\ \text{s.t.} & \operatorname{diag}(X) = e \\ & x_{ij} \geq -\frac{1}{k-1} \\ & X \succeq 0. \end{array} \tag{3.28}$$

Note that there is no corresponding statement to Lemma 3.1.1. Restricting the rank of X to at most $k-1$ does not guarantee that the vectors v_i form a $k-1$ dimensional simplex in general.

Before describing the approximation algorithm based on the semidefinite relaxation we consider a simple randomized algorithm. If the vertices V are randomly partitioned into k sets, the probability that both vertices of an edge belong to the same set is $\frac{1}{k}$. Therefore the expected weight of this simple random k-cut is

$$\sum_{i<j} a_{ij} (1 - \tfrac{1}{k}).$$

For large k there is not much room for improvement.

Turning to the semidefinite relaxation, the decisive step in the approximation algorithm is to replace the random hyperplane used in the rounding algorithm for max-cut in Section 3.1 by k random vectors u_h independently chosen from the standard normal distribution. Each vector u_h represents one set of the partition. A vector v_i is assigned to $u_{\bar{h}}$ if the inner product $v_i^T u_h$ is

maximal for $\bar{h}$. The probability that two vertices, i and j, are assigned to the same set depends on the value of $v_i^T v_j$ only. Let us denote this probability by $P(v_i^T v_j)$. Then for any feasible configuration of the v_i the expected value of a random k-cut is

$$\sum_{i<j} a_{ij}(1 - P(v_i^T v_j)).$$

The derivation of a bound for $1 - P(v_i^T v_j)$ is rather technical and quite involved. Therefore we state the results only and refer the interested reader to Frieze and Jerrum [1995] for the proof.

Theorem 3.4.3 *(Frieze and Jerrum [1995]) For $k \geq 2$ let*

$$\alpha_k = \min_{\frac{-1}{k-1} \leq v_i^T v_j < 1} \frac{1 - P(v_i^T v_j)}{\frac{(k-1)(1-v_i^T v_j)}{k}}.$$

Then α_k satisfies

1. $\alpha_k > 1 - \frac{1}{k}$;
2. $\alpha_k - (1 - \frac{1}{k}) \approx 2k^{-2} \log k$;
3. $\alpha_2 > 0.87856$, $\alpha_3 > 0.8002$, $\alpha_4 > 0.8503$, $\alpha_5 > 0.8742$, $\alpha_{10} > 0.9266$, *and* $\alpha_{100} > 0.9906$.

As one would expect, for large k the improvement turns out to be marginal in comparison to the trivial algorithm. Observe, that α_2 is exactly the value of the Goemans-Williamson algorithm. The theorem says, that for a feasible set of vectors v_i (in particular for the optimal solution of (SMkC)) and for nonnegative edge weights a_{ij} the expected size of a random k-cut is at least

$$\sum_{i<j} a_{ij}(1 - P(v_i^T v_j)) \;\geq\; \alpha_k \sum_{i<j} a_{ij} \frac{(k-1)(1 - v_i^T v_j)}{k},$$

so there must exist a k-cut of this size or larger. Again, a deterministic version can be constructed by the method of Mahajan and Ramesh [1995]; this yields an $(\alpha_k - \varepsilon)$ approximation algorithm for max k-cut.

The same approach can be used to develop relaxations for the k-equipartition problem. Given k, m, and $G(V, E)$, equipartition asks for a partition of V, $|V| = k \cdot m$, into k subsets of equal size m such that the weight of the edges having their respective endpoints in distinct sets is maximized. Stated with respect to an "integral" solution X of (3.28) each row of X should contain m elements of value 1 and $(n - m)$ elements of value $-\frac{1}{k-1}$. The row sum is zero for each row. This suggests the following relaxation for k-equipartition,

$$\begin{array}{ll} \max & \frac{k-1}{2k} \langle L, X \rangle \\ \text{s.t.} & \operatorname{diag}(X) = e \\ & x_{ij} \geq -\frac{1}{k-1} \\ & e^T X e = 0 \\ & X \succeq 0. \end{array} \tag{3.29}$$

This semidefinite program clearly has no strictly feasible solution since e is always an eigenvector of X with eigenvalue zero. In order to compute a solution of this relaxation by an interior point method, it is necessary to eliminate the eigenvalue zero by projecting the problem as described in Chapter 2. In the case of $k = 2$ it is again possible to derive an approximation algorithm that has a better performance guarantee than the trivial randomized algorithm, we refer to Frieze and Jerrum [1995]; Ye [1999a] for details. Karisch and Rendl [1998]; Karisch [1995] derive an equivalent bound to (3.29) by another approach and relate it to a spectral bound of Donath and Hoffman [1973], see also Wolkowicz and Zhao [1996].

3.4.3 Coloring

The approximate graph coloring algorithm of Karger, Motwani, and Sudan [1994] is one of the finest examples of this vector labeling technique. A detailed description of the algorithm is beyond the scope of this text, we will only hint at the underlying idea. For the purpose of explanation let us assume that we can color the graph with k colors. We represent the k colors by the vectors $\{w_1, \ldots, w_k\}$. Vertices that may be colored by the same color may be assigned the same vector. Vertices i and j that are connected by an edge $ij \in E$ must be assigned distinct labels w_i and w_j with $w_i^T w_j \leq -1/(k-1)$.

For coloring problems k is not known in advance. Therefore we introduce a variable κ and require that the relaxed vectors v_i and v_j corresponding to endpoints of an edge $ij \in E$ satisfy $v_i^T v_j \leq -1/(\kappa - 1)$. Optimizing over κ yields a lower bound on the number of colors needed,

$$\begin{array}{lll} \min & \kappa & \\ \text{s.t.} & \operatorname{diag}(X) = e & \\ & x_{ij} \leq -\frac{1}{\kappa - 1} & \forall ij \in E \\ & X \succeq 0. & \end{array} \tag{3.30}$$

A graph is said to be *vector k-colorable* if k is not smaller than the optimal solution value of κ in (3.30). By the construction above every graph that can be colored by k colors is also vector k-colorable (but not necessarily vice versa). k cannot become smaller than the largest clique in G, because for a clique of size h the matrix X will have a principal submatrix of the form $\frac{\kappa}{\kappa-1} I_h - \frac{1}{\kappa-1} ee^T$ (at best) which is no longer positive semidefinite for $\kappa < h$.

Relaxation (3.30) is the starting point for the approximate graph coloring algorithm of Karger, Motwani, and Sudan [1994]. The central idea is based on using $h > k$ random vectors representing the colors. The vectors of the optimal solution of (3.30) are assigned to these vectors as in max k-cut. h is chosen such that the probability that both endpoints of an edge are assigned the same color is small enough such that at most $|V|/4$, say, vertices will have conflicting assignments. For these vertices the process is repeated using new colors. With careful parameter tuning and some additional considerations one arrives at the following theorems.

Theorem 3.4.4 *(Karger, Motwani, and Sudan [1994]) A vector 3-colorable graph G with n vertices and maximum degree Δ can be colored with $O(\min\{\Delta^{\frac{1}{3}} \log^{\frac{4}{3}} \Delta, n^{\frac{1}{4}}\} \cdot \log n)$ colors by a polynomial time randomized algorithm (with high probability).*

Theorem 3.4.5 *(Karger, Motwani, and Sudan [1994]) A vector k-colorable graph can be colored using $O(\Delta^{1-\frac{2}{k}})$ or $O(n^{1-\frac{3}{k+1}})$ colors by a polynomial time randomized algorithm (with high probability).*

3.4.4 Betweenness

Chor and Sudan [1995] propose a semidefinite relaxation of the betweenness problem that is based on a geometric interpretation of semidefinite programming. The betweenness problem consists of a set S of symbols $s_1, \ldots, s_n$ and a set B of m betweenness constraints (s_i, s_j, s_k). A betweenness constraint is satisfied, if for some specific total order $<$ of the symbols $s \in S$ the middle element s_j is between s_i and s_k, *i.e.*, either $s_i < s_j < s_k$ or $s_i > s_j > s_k$. The task is to construct a total ordering of the s_i satisfying as many betweenness constraints as possible. It is *NP*-complete to decide whether there is an ordering satisfying all constraints. The maximization problem is MAX-SNP-complete, so there exists an $\varepsilon > 0$ such that it is *NP*-complete to decide whether there is an ordering satisfying $(1 - \varepsilon)m$ constraints.

A simple random algorithm picks an arbitrary order. The expected number of satisfied constraints for this algorithm is $m/3$ since there are six possible orderings of three elements and two of them are successful. On the other hand for the betweenness problem with three symbols a, b, c and the constraints (a, b, c), (a, c, b), and (b, a, c) at most one third of the constraints can be satisfied. So there can be no algorithm satisfying more than $m/3$ constraints for all problem instances.

The algorithm to be described will either find an ordering such that $m/2$ of the constraints are satisfied, or assert that it is impossible to satisfy all constraints.

Again we associate with each symbol s_i a vector (or point) $v_i \in \mathbb{R}^n$. A feasible ordering of the symbols can be thought of as an arrangement of the associated points on a straight line with equidistant spacing between the points such that for any betweenness constraint (s_i, s_j, s_k) v_j is strictly contained in the straight line segment with endpoints v_i and v_k. Several properties of such feasible arrangements can be formulated as linear inequalities over the squares of distances between the points. The square of the Euclidean distance between two points v_i and v_j is

$$d_{ij}^2 := \|v_i - v_j\|^2 = (v_i - v_j)^T(v_i - v_j) = v_i^T v_i + v_j^T v_j - 2v_i^T v_j.$$

Let X denote the Gram matrix of the vectors $v_1, \ldots, v_n$. Then the square of the distance between the points v_i and v_j can be expressed in terms of the matrix elements of X,

$$d_{ij}^2 = x_{ii} + x_{jj} - 2x_{ij}. \tag{3.31}$$

A linear program formulated over the squares of distances of points can thus be cast as a semidefinite program with the points coded in a semidefinite matrix variable X.

A canonical way to specify a total order on n symbols is to assign a real number $p_i \in \mathbb{R}$ to each s_i ($\mathbb{R}$ is the straight line alluded to above, the p_i are one dimensional realizations of the v_i). In particular it suffices to select the numbers from $\left\{0, \frac{1}{n-1}, \ldots, \frac{n-2}{n-1}, 1\right\}$. For these values the distances between any two points are at least $\frac{1}{n-1}$ and at most 1. For a betweenness constraint (s_i, s_j, s_k) that is satisfied by an assignment p_i, p_j, p_k the distances satisfy

$$(p_i - p_j)^2 + (p_j - p_k)^2 < (p_i - p_k)^2.$$

To strengthen this constraint we observe that the ratio of the left hand side over the right hand side is largest for $|p_i - p_j| = \frac{1}{n-1}$ and $|p_j - p_k| = 1 - \frac{1}{n-1}$ or vice versa. We denote this ratio by

$$(\tfrac{1}{n-1})^2 + (1 - \tfrac{1}{n-1})^2 \leq 1 - \tfrac{1}{n} =: \delta_n.$$

Summing up, the squares of the distances of a feasible arrangement of the points must satisfy the following system of inequalities,

$$\begin{aligned} d_{ij}^2 + d_{jk}^2 &\leq \delta_n d_{ik}^2 \qquad && \forall (s_i, s_j, s_k) \in B \\ (\tfrac{1}{k-1})^2 &\leq d_{ij} \leq 1 \qquad && 1 \leq i < j \leq n. \end{aligned}$$

Using (3.31) the problem of finding points in $\mathbb{R}^n$ satisfying these constraints can be formulated as the problem of finding any feasible solution of a semidefinite program. It is easy to convince oneself that the corresponding semidefinite program has a strictly feasible solution if there exists an ordering satisfying all constraints. Thus, if the algorithm is not able to produce a feasible solution then there is no ordering satisfying all constraints. In the following we assume that $v_1, \ldots, v_n$ is a feasible solution of the system above. From this feasible solution we construct an ordering satisfying at least half of the betweenness constraints.

A total order is constructed by projecting the v_i onto a random vector $h \in \mathbb{R}^n$ chosen by the standard normal distribution (which is spherically symmetric),

$$v_i^T h < v_j^T h \qquad \rightarrow \qquad s_i < s_j.$$

The probability that a constraint (s_i, s_j, s_k) is satisfied by this ordering depends on the vectors v_i, v_j, and v_k only. In particular it depends on the angle θ between the vectors $v_i - v_j$ and $v_k - v_j$, as we know from the proof of Theorem 3.1.3,

$$\Pr(\operatorname{sgn}((v_i - v_j)^T h) \neq \operatorname{sgn}((v_k - v_j)^T h)) = \frac{\arccos\left((\frac{v_i - v_j}{\|v_i - v_j\|})^T \frac{v_k - v_j}{\|v_k - v_j\|}\right)}{\pi} = \frac{\theta}{\pi}.$$

It remains to bound θ. Because v_i, v_j and v_k satisfy $\|v_i - v_j\|^2 + \|v_k - v_j\|^2 < \|v_i - v_k\|^2$, we obtain

$$\cos(\theta) = \frac{\|v_i - v_j\|^2 + \|v_k - v_j\|^2 - \|v_i - v_k\|^2}{2\|v_i - v_j\| \, \|v_k - v_j\|} < 0,$$

and therefore $\theta > \frac{\pi}{2}$. Note, that this bound is independent of the specific constraint. In consequence each constraint is satisfied with probability at least one half proving the following Theorem.

Theorem 3.4.6 *(Chor and Sudan [1995]) If the randomized algorithm is applied to a feasible set of vectors $v_1, \ldots, v_n$ the expected number of all constraints that are satisfied is at least $\frac{m}{2}$.*

Again it is possible to construct a deterministic procedure that yields a solution with at least this expected value. For details we refer to Chor and Sudan [1995].

3.5 Remarks on the Literature

A comprehensive book about the max-cut problem is Deza and Laurent [1997], a concise survey was written by Poljak and Tuza [1995]. Delorme and Poljak [1993] introduced the eigenvalue bound for max-cut and established Theorem 3.1.13. They studied further properties of this relaxation in Delorme and Poljak [1991]. Implementational results with the eigenvalue bound are described in Poljak and Rendl [1995b]. The equivalence of the eigenvalue bound to the semidefinite formulation, that is due to Schrijver, is proved in Poljak and Rendl [1995a]. Poljak, Rendl, and Wolkowicz [1995] observed that the semidefinite relaxation turns up generically for various relaxation approaches to quadratic $\{-1, 1\}$ programming. Laurent and Poljak [1995]; Laurent and Poljak [1996b] studied the geometrical properties of the primal feasible set of this semidefinite relaxation. The randomized approximation algorithm is due to Goemans and Williamson [1995], it was derandomized by Mahajan and Ramesh [1995]. Karloff [1996] proved that the quality bound equals the performance ratio. Nesterov [1998]; Ye [1997b]; Ye [1999b]; Ye [1997a] extended the approach (with different approximation measure) to more general cost matrices and continuous quadratic optimization problems. Zwick [1999] improved the approximation ratio for max-cut instances where the percentage of edges in the cut is "small." A detailed computational study of the semidefinite relaxation within a branch and cut framework is given in Helmberg and Rendl [1998]. A promising interior point approach that is able to exploit some structure was developed by Benson, Ye, and Zhang [1998]. Homer and Peinado [1995]; Burer and Monteiro [1998] propose special purpose projected gradient methods for (SMC). An alternative method based on eigenvalue computations for solving the semidefinite relaxation without further side constraints is proposed in Klein and Lu [1996]. In Chapter 5 a subgradient method will be discussed that combines, to some extent, interior point methods with the structural advantages of eigenvalue computations.

The reference for polyhedral aspects of quadratic 0-1 programming is Padberg [1989]. De Simone [1989] proved that the max-cut polytope and the boolean quadric polytope are isomorphic. The primal semidefinite relaxation as presented seems to appear first in Lovász and Schrijver [1991], but Shor [1987] already used its dual form. The derivation of the primal as the intersection of dyadic products of shifted vectors was proposed by Balas, Ceria, Cornuejols, and Pataki [1994]. The equivalence of the semidefinite relaxations of max-cut and quadratic 0-1 programming was established in Helmberg, Poljak, Rendl, and Wolkowicz [1995]; Laurent, Poljak, and Rendl [1997]. The representation as a scaling transformation is due to Helmberg [2000].

The first semidefinite relaxation of a combinatorial optimization problem (in the form of a semidefinite program) was introduced by Lovász [1979]. In fact, he introduced several semidefinite relaxations for the independent set problem and proved that they are all equivalent. They are referred to as the Lovász ϑ-function. Grötschel, Lovász, and Schrijver [1981]; Grötschel, Lovász, and Schrijver [1984] proved that the ϑ-function can be used to solve the independent set problem on perfect graphs in polynomial time. The book Grötschel, Lovász, and Schrijver [1988] includes a comprehensive treatment of the ϑ-function. A more recent survey is Knuth [1994]. The semidefinite relaxation presented here appears in Lovász and Schrijver [1991]. The paper also gives a proof of the equivalence to previous formulations of the ϑ-function. Our proof of Theorem 3.3.1

is inspired by Lovász and Schrijver [1991] as well as Grötschel, Lovász, and Schrijver [1988]. The fact that the stable set polytope is a face of the boolean quadric polytope was observed in Padberg [1989]. Further investigations with respect to the connection between the metric polytope and the semidefinite relaxation can be found in Laurent, Poljak, and Rendl [1997].

The presentation of the quadratic 0-1 knapsack relaxations follows Helmberg, Rendl, and Weismantel [2000]. The y_i- and $(1-y_i)$-representations appear implicitly in Balas [1975], the technique of obtaining quadratic inequalities from pairwise multiplication of valid linear inequalities is fully developed in Sherali and Adams [1990]; Lovász and Schrijver [1991]. Relaxation (SQK3) and its relation to the other relaxations was developed by Bauvin and Goemans (personal communication).

The approximation algorithm for max-2sat appears in Goemans and Williamson [1995], a more involved algorithm including the triangle constraints with better performance guarantee is sketched in Feige and Goemans [1995]. Using a similar technique Karloff and Zwick [1997] develop an 7/8 approximation algorithm for max-3sat, it is optimal with respect to the approximation guarantee. The work on approximation algorithms for max-sat variations has been continued in Zwick [1998a]; Zwick [1998b]; Halperin and Zwick [1999]. de Klerk, van Maaren, and Warners [1999] investigate possibilities to detect unsatisfiable instances by semidefinite programming.

The approach for max k-cut is due to Frieze and Jerrum [1995], the semidefinite relaxation of k-equipartition is based on the same paper. For $k = 2$ the authors provide a randomized rounding scheme that produces a feasible solution within 0.65 of the optimal semidefinite solution. Ye [1999a] achieves a slightly better bound (0.699) by balancing the randomness of the rounding scheme. Karisch and Rendl [1998]; Karisch [1995] arrive at an equivalent semidefinite relaxation for k-equipartitioning by a different approach. They show a tight relation of their bound to spectral bounds of Donath and Hoffman [1973]; Rendl and Wolkowicz [1995] and a bound of Alizadeh [1995]. Computational experience with semidefinite relaxations of the equipartition problem can also be found in Wolkowicz and Zhao [1996].

The approximation algorithm for coloring k-colorable graphs is due to Karger, Motwani, and Sudan [1994]. For a result on coloring bipartite hypergraphs see Chen and Frieze [1996]. Chor and Sudan [1995] developed the semidefinite approach to the betweenness problem.

Another topic that has received increasing interest in the semidefinite community is the quadratic assignment problem, Zhao, Karisch, Rendl, and Wolkowicz [1998]; Lin and Saigal [1997]; Anstreicher, Chen, Wolkowicz, and Yuan [1998]; Anstreicher [2000]; Anstreicher and Brixius [1999]. In the field of scheduling the only reference (to the best of our knowledge) is Skutella [1998] who constructed an approximation algorithm for parallel machine scheduling.

For further references on semidefinite programming and combinatorial optimization we recommend the surveys Goemans [1997]; Goemans [1998].

Chapter 4

Interior Point Methods

In nonlinear programming many techniques have been developed to transform constrained optimization problems into unconstrained problems. One approach to handle inequality constraints consists in adding a barrier term to the cost function. In the case of minimization problems the value of the barrier term is small in the interior of the feasible region but grows to infinity when the boundary is approached. In a line search starting from the interior this prevents leaving the interior. Furthermore, a descent direction in a point close to the boundary will automatically point away from the boundary. Unfortunately, optima are usually located on the boundary. In order to produce a sequence of iterates that converge to the optimum, a mechanism has to be provided that reduces the influence of the barrier term as the optimization process continues. This is achieved by weighting the barrier term and diminishing the weight successively. Under certain conditions the minima of the sequence of barrier problems can be shown to converge to an optimal solution of the original problem. This is known as the sequential unconstrained minimization technique (Fiacco and McCormick [1968]) and forms the backbone of interior point methods. Typically, the minima of the subproblems are not computed exactly but approximated by a few Newton steps. Since Newton's method exploits second order information and works particularly well on the class of barrier problems associated with semidefinite programs, the algorithms converge very fast. An approximately optimal solution is obtained within a polynomial number of iterations (in the real number model of computation). On the other hand the computation of a single step is computationally rather expensive and structural properties of constraints are difficult to exploit. Within current technology this limits these methods to problems of moderate size, say, to about 7000 constraints.

In Section 4.1 we discuss the general approach and properties of the sequence of minimizers of the subproblems. Then we explain the typical structure of so-called primal-dual interior point algorithms. In the case of semidefinite programming the linear system for determining the step direction differs slightly from the standard Newton system; this has led to the development of several different search directions. We describe some of the more popular choices.

Section 4.2 is devoted to the semidefinite analogue of the primal-dual interior point algorithm of Monteiro and Adler [1989a] for linear programming for the search direction of Helmberg, Rendl, Vanderbei, and Wolkowicz [1996]; Kojima, Shindoh, and Hara [1997], that was later rediscovered by Monteiro [1997]. The analysis is an adapted version of the analysis of Zhang [1998] for his infeasible primal-dual algorithm.

In Section 4.3 we investigate a skew-symmetric embedding for semidefinite programs. This embedding guarantees a feasible starting point that is also the minimizer of the first barrier problem. The optimal solution to the embedded problem yields optimal solutions of the original problems if primal and dual problems are feasible and if there is no duality gap.

Implementational issues are considered in Section 4.4. We will investigate some efficiency aspects, explain the so-called predictor-corrector approach, and give typical parameter settings.

4.1 Barrier Problem, Central Path, and Search Directions

In Section 2.2 we investigated duality aspects and the existence of optimal solutions for the standard primal semidefinite program (2.1) and its standard dual (2.3). We recall their definition,

$$\begin{array}{lll} & \min & \langle C, X\rangle \\ \text{(PSDP)} & \text{s.t.} & \mathcal{A}X = b \\ & & X \succeq 0 \end{array} \qquad \begin{array}{lll} & \max & \langle b, y\rangle \\ \text{(DSDP)} & \text{s.t.} & \mathcal{A}^T y + Z = C \\ & & y \in \mathbb{R}^m,\ Z \succeq 0. \end{array}$$

By the strong duality theorem 2.2.5, a sufficient condition for the attainment of optimal primal and dual solutions is the existence of strictly feasible (cf. Definition 2.2.2) primal and dual solutions. We will see in Section 4.3 that a primal-dual pair of semidefinite programs can always be embedded in a slightly larger self-dual semidefinite program that possesses strictly feasible solutions. Therefore there is no significant loss in generality by working under the following assumption.

Assumption 4.1.1 *There exists a strictly feasible X^0 for (PSDP) and a strictly feasible pair (y^0, Z^0) for (DSDP).*

Interior point algorithms start within the cone of positive semidefinite matrices. In order to avoid leaving this cone during the optimization process the task of minimizing the original semidefinite program is replaced by approximately minimizing a sequence of auxiliary barrier problems. The auxiliary problem contains an additional barrier term $-\mu \log\det(X)$ in the cost function, where $\mu > 0$ is the so-called *barrier parameter* and $-\log\det(X)$ is the *barrier function*.

$$\begin{array}{ll} \min & \langle C, X\rangle - \mu \log\det(X) \\ \text{s.t.} & \mathcal{A}X = b \\ & X \succ 0. \end{array}$$

By Assumption 4.1.1 and the proof of Lemma 2.2.3 the level sets $\{X \succeq 0 :\ \mathcal{A}X = b, \langle C, X\rangle = d\}$ are bounded and closed for every $d \in \mathbb{R}$. The cost function is strictly convex by Lemma 1.4.2. Therefore, the optimal solution exists and is unique.

Remark 4.1.2 *Besides its strict convexity $-\log\det(X)$ has a second important property. It belongs to the class of so-called strongly self concordant functions (see Nesterov and Nemirovskii [1994]) which harmonize with Newton's method. Intuitively, Newton's method approximates a function by a quadratic model and solves the minimization problem exactly within this model. The resulting descent direction is the better the less the function deviates from this quadratic model. For strongly self concordant functions the change of the second derivative is locally bounded by a Lipschitz condition, therefore the quadratic model is of good quality for comparatively large regions.*

Since $\det(X) = \prod_{i=1}^n \lambda_i(X)$, we have $-\log\det(X) = -\sum_{i=1}^n \log\lambda_i(X)$ (in linear programming, the matrix is diagonal, $X = \mathrm{Diag}(x)$ with $x \in \mathbb{R}^n_+$, and the barrier function reads $-\sum_{i=1}^n \log x_i$). The barrier function grows to infinity if an eigenvalue of X tends to zero, *i.e.*, if X approaches the boundary of the semidefinite cone. In a line search along some search direction from inside the semidefinite cone, the barrier term guarantees that the result is again a positive definite matrix. Likewise, the optimal solution of the barrier problem is in the interior of the semidefinite cone. The influence of the barrier function on the cost function is controlled by the barrier parameter μ. For a sequence of barrier problems with $\mu \to 0$, the original cost function will eventually prevail on the interior of the feasible set except within an ε-distance of the boundary. Thus, we expect the sequence of optimizers of the barrierproblems problems to converge to an optimizer of the original problem.

We transform the barrier problem into an unconstrained problem for $X \succ 0$ by introducing a Lagrange multiplier y for the the equality constraints,

$$\mathcal{L}_\mu(X, y) = \langle C, X\rangle - \mu \log\det(X) + \langle y, b - \mathcal{A}X\rangle\,.$$

For given $y \in \mathbb{R}^m$, the function $\mathcal{L}_\mu(\cdot, y)$ is a smooth, strictly convex function on S^+_n, because it is the sum of a linear and a smooth strictly convex function. For given $X \succeq 0$ the function $\mathcal{L}_\mu(X, \cdot)$

is linear. Functions that are convex in one coordinate and concave in the other are called *saddle functions* (Rockafellar [1970]). A point $(X_\mu, y_\mu) \in S_n^+ \times \mathbb{R}^m$ is called a *saddle point* if

$$\inf_{X \succeq 0} \mathcal{L}_\mu(X, y_\mu) \geq \mathcal{L}_\mu(X_\mu, y_\mu) \geq \sup_{y \in \mathbb{R}^m} \mathcal{L}_\mu(X_\mu, y).$$

A saddle point yields correct Lagrange multipliers y_μ and the optimal solution X_μ for the barrier problem. Since for a saddle point (X_μ, y_μ) the point X_μ is a minimizer of $\mathcal{L}_\mu(\cdot, y_\mu)$ and y_μ is a maximizer of $\mathcal{L}_\mu(X_\mu, \cdot)$, the partial derivatives of $\mathcal{L}_\mu$ with respect to X and with respect to y must be zero in any saddle point. These first order necessary conditions (in our convex setting they are also sufficient, see Rockafellar [1970]) are called the Karush-Kuhn-Tucker conditions (KKT-conditions). Using (1.13) for the derivative of the barrier function we arrive at the system

$$\begin{aligned} \nabla_X \mathcal{L}_\mu &= C - \mu X^{-1} - \mathcal{A}^T y = 0 \\ \nabla_y \mathcal{L}_\mu &= b - \mathcal{A}X = 0. \end{aligned}$$

In a primal-dual formulation we set $Z = \mu X^{-1}$ and rewrite the KKT-conditions in the following form (starting from the dual barrier problem one would arrive at the same system).

$$\begin{aligned} \mathcal{A}X &= b, \quad & X \succ 0 \\ \mathcal{A}^T y + Z &= C, \quad & Z \succ 0 \\ XZ &= \mu I. & \end{aligned} \tag{4.1}$$

The first line requires primal feasibility and the second dual feasibility. For $\mu = 0$ the third line would correspond to the complementarity condition $XZ = 0$; by (2.9), any solution to system (4.1) would yield optimal primal and dual solutions. In this light we interpret the third line as a perturbed complementarity condition. The solution of this system is unique with respect to X and Z, but not necessarily for y. By Assumption 4.1.1, however, the system $\mathcal{A}X = b$ is consistent and it is possible to eliminate all y-variables as explained in (2.4). The sole purpose of y is to span the feasible set of Z. Therefore we will concentrate on the variables X and Z, and use y only if it is convenient.

We denote the solution of system (4.1) for some fixed μ by (X_μ, Z_μ). X_μ is the unique optimal solution to the primal barrier problem and Z_μ is the unique optimal solution to the analog dual barrier problem. X_μ and Z_μ are feasible primal and dual points of the original problem with a gap of $\langle Z, X \rangle = n\mu$ between the objective values, cf. (2.9). The set of solutions (X_μ, Z_μ) for $\mu > 0$ forms the so-called *central path* which is a smooth curve (see, *e.g.*, Kojima, Shindoh, and Hara [1997]).

In the following we will show that for $\mu \to 0$ the central path converges to a point (X^*, Z^*) with X^* an optimal solution of the original primal and Z^* an optimal solution of the original dual problem. The proof exploits the orthogonality of the affine subspaces spanned by the primal and dual feasible sets.

Lemma 4.1.3 *Let* $X', X'' \in \{X \in S_n : \mathcal{A}X = b\}$ *and* $Z', Z'' \in \{C - \mathcal{A}^T y : y \in \mathbb{R}^m\}$. *Then* $\langle X' - X'', Z' - Z'' \rangle = 0$.

Proof. We have $\mathcal{A}(X' - X'') = 0$ and $\mathcal{A}^T(y' - y'') = -(Z' - Z'')$. Therefore,
$0 = \langle X' - X'', \mathcal{A}^T(y' - y'') \rangle = - \langle X' - X'', Z' - Z'' \rangle$. ■

Lemma 4.1.4 *For a sequence* $\mu_k > 0$, $k \in \mathbb{N}$, *with* $\mu_k \to 0$, *the corresponding solutions* (X_{μ_k}, Z_{μ_k}) *of (4.1) converge to a pair of optimal solutions* (X^*, Z^*) *of (PSDP) and (DSDP).*

Proof. In order to prove that the sequence (X_{μ_k}, Z_{μ_k}) is contained in a compact set, we exploit Assumption 4.1.1 together with Lemma 4.1.3,

$$0 = \langle X_{\mu_k} - X^0, Z_{\mu_k} - Z^0 \rangle = \langle X_{\mu_k}, Z_{\mu_k} \rangle - \langle X_{\mu_k}, Z^0 \rangle - \langle X^0, Z_{\mu_k} \rangle + \langle X^0, Z^0 \rangle .$$

By (4.1) $\langle X_{\mu_k}, Z_{\mu_k}\rangle = n\mu_k$, therefore

$$\left\langle X_{\mu_k}, Z^0\right\rangle + \left\langle X^0, Z_{\mu_k}\right\rangle = n\mu_k + \left\langle X^0, Z^0\right\rangle. \tag{4.2}$$

Lemma 1.2.4 implies

$$\lambda_{\max}(X_{\mu_k})\lambda_{\min}(Z^0) + \lambda_{\min}(X^0)\lambda_{\max}(Z_{\mu_k}) \leq n\mu_k + \left\langle X^0, Z^0\right\rangle.$$

Since $\lambda_{\min}(Z^0) > 0$ and $\lambda_{\min}(X^0) > 0$ the iterates remain bounded. Consequently there is a convergent subsequence converging to some point (X^*, Z^*) which must be feasible with $\langle X^*, Z^*\rangle = 0$. This proves optimality of the pair (X^*, Z^*). ■

In linear programming the central path converges to a strictly complementary solution, *i.e.*, a primal dual pair of optimal vectors $x, z \in \mathbb{R}^n$ with either $x_i = 0$ or $z_i = 0$ (but not both) for $i = 1, \ldots, n$. Does an analogous property hold for semidefinite programming? To answer this we have to clarify what complementarity means in semidefinite programming. For any optimal solution $\hat{X}$ of the primal and any optimal solution $\hat{Z}$ of the dual the inner product $\langle \hat{X}, \hat{Z}\rangle$ is zero. By Lemma 1.2.3, $\langle \hat{X}, \hat{Z}\rangle = 0$ implies that $\hat{X}\hat{Z} = 0$ and so $\hat{X}$ and $\hat{Z}$ are simultaneously diagonalizable (Theorem A.0.3). The non-zero eigenvectors of any optimal primal solution $\hat{X}$ are in the null space of any optimal dual solution $\hat{Z}$ and vice versa. In other words, the two minimal faces of the semidefinite cone containing the respective convex sets of primal and dual optimal solutions are spanned by orthogonal subspaces of $\mathbb{R}^n$. This suggests the following definition (de Klerk, Roos, and Terlaky [1997]).

Definition 4.1.5 *A pair of optimal solutions* (X^*, Z^*) *is* maximally complementary *if* X^* *and* Z^* *have maximal rank among all optimal solutions.*
An optimal pair (X^*, Z^*) *is* strictly complementary *if* $\operatorname{rank}(X^*) + \operatorname{rank}(Z^*) = n$.

In contrast to linear programming the existence of strictly complementary solutions is not guaranteed. However, the central path gets as close to strict complementarity as possible. The point to which the central path converges for $\mu \to 0$ is maximally complementary. To prove this we need a technical lemma.

Lemma 4.1.6 *(de Klerk, Roos, and Terlaky [1997]) Let* X_{μ_k}, $k \in \mathbb{N}$, *denote a sequence of positive definite matrices* $X_{\mu_k} \succ 0$ *converging to a positive semidefinite matrix* $X^* \succeq 0$ *and let* $\hat{X} \succeq 0$ *be a positive semidefinite matrix. If for some* $K \in \mathbb{R}$

$$\left\langle X_{\mu_k}^{-1}, \hat{X}\right\rangle \leq K \qquad \forall k \in \mathbb{N}$$

then $\operatorname{rank}(X^*) \geq \operatorname{rank}(\hat{X})$.

Proof. For $\operatorname{rank}(\hat{X}) = 0$ the result is trivial. Let therefore $r = \operatorname{rank}(\hat{X}) > 0$. Using eigenvalue decomposition we write $\hat{X}$ in the form $\hat{X} = \hat{P}\hat{\Lambda}\hat{P}^T$ with $\hat{P}^T\hat{P} = I_r$ and diagonal matrix $\hat{\Lambda} \in S_r^{++}$. For the positive definite X_{μ_k} let $X_{\mu_k} = P_k\Lambda_k P_k^T$ with $P_k^T P_k = I$ and diagonal matrices $\Lambda_k \in S_n^{++}$. To show that X^* has at least rank r we will prove that there is a subsequence of the X_{μ_k} such that at least r eigenvalues of Λ_k are bounded away from zero.

$$\operatorname{tr}(\hat{P}^T X_{\mu_k}^{-1}\hat{P}) \leq \operatorname{tr}(\hat{\Lambda}^{-1}\hat{\Lambda}\hat{P}^T X_{\mu_k}^{-1}\hat{P}) \leq \lambda_{\max}(\hat{\Lambda}^{-1})\left\langle \hat{X}, X_{\mu_k}^{-1}\right\rangle \leq \frac{1}{\lambda_{\min}(\hat{\Lambda})}K =: \hat{K}$$

In the following let $Q_k = P_k^T\hat{P}$ which again satisfies $Q_k^T Q_k = I_r$ and let the row vectors q_{ki} denote the i-th row of Q_k for $i = 1, \ldots, n$.

$$\operatorname{tr}(\hat{P}^T X_{\mu_k}^{-1}\hat{P}) = \operatorname{tr}(Q_k^T\Lambda_k^{-1}Q_k) = \sum_{i=1}^{n}\frac{q_{ki}q_{ki}^T}{\lambda_i(\Lambda_k)} \leq \hat{K}.$$

The numerator of each summand satisfies $0 \leq q_{ki} q_{ki}^T \leq 1$ and these terms sum up to r,

$$\sum_{i=1}^{n} q_{ki} q_{ki}^T = \operatorname{tr}(Q_k Q_k^T) = \operatorname{tr}(Q_k^T Q_k) = r.$$

So at least r of the $q_{ki} q_{ki}^T$ will be larger than some constant $\alpha > 0$ independent of k (in fact, $\alpha = \frac{1}{n-(r-1)}$ is the best choice). The r corresponding $\lambda_i(\Lambda_k)$ cannot get smaller than $\frac{\alpha}{K}$. Eigenvalues depend continuously on the elements of the matrix. Therefore at least r eigenvalues of X^* are positive, because the X_{μ_k} converge to X^*. This proves $\operatorname{rank}(X^*) \geq \operatorname{rank}(\hat{X})$. ■

With this lemma the proof of maximal complementarity is easy.

Lemma 4.1.7 *(de Klerk, Roos, and Terlaky [1997]) For a sequence $\mu_k > 0$, $k \in \mathbb{N}$, with $\mu_k \to 0$, the corresponding solutions (X_{μ_k}, Z_{μ_k}) of (4.1) converge to a maximally complementary, optimal pair (X^*, Z^*) of (PSDP) and (DSDP).*

Proof. From Lemma 4.1.4 we know that the sequence (X_{μ_k}, Z_{μ_k}) converges to an optimal solution (X^*, Z^*). To show that this solution is maximally complementary we have to prove that for arbitrary primal and dual optimal solutions $\hat{X}$ and $\hat{Z}$

$$\operatorname{rank}(X^*) \geq \operatorname{rank}(\hat{X}) \quad \text{and} \quad \operatorname{rank}(Z^*) \geq \operatorname{rank}(\hat{Z}).$$

Replacing X^0 and Z^0 by $\hat{X}$ and $\hat{Z}$ in (4.2) yields

$$\left\langle X_{\mu_k}, \hat{Z} \right\rangle + \left\langle \hat{X}, Z_{\mu_k} \right\rangle = n\mu_k,$$

because $\left\langle \hat{X}, \hat{Z} \right\rangle = 0$. The perturbed complementarity condition $X_{\mu_k} Z_{\mu_k} = \mu_k I$ implies $X_{\mu_k} = \mu_k Z_{\mu_k}^{-1}$ and $Z_{\mu_k} = \mu_k X_{\mu_k}^{-1}$, therefore

$$\left\langle Z_{\mu_k}^{-1}, \hat{Z} \right\rangle + \left\langle \hat{X}, X_{\mu_k}^{-1} \right\rangle = n.$$

We conclude that $\left\langle Z_{\mu_k}^{-1}, \hat{Z} \right\rangle \leq n$ and $\left\langle \hat{X}, X_{\mu_k}^{-1} \right\rangle \leq n$ and apply Lemma 4.1.6 to complete the proof. ■

In other words, this lemma states that the central path converges to a point in the relative interior of the optimal face. This point is called the *analytic center* of the optimal face.

We would like to compute an approximate solution of (4.1), *i.e.*, a solution to

$$F_\mu(X, y, Z) = \begin{pmatrix} \mathcal{A}X - b \\ \mathcal{A}^T y + Z - C \\ XZ - \mu e \end{pmatrix} = 0.$$

Newton's method computes a step direction $(\Delta X, \Delta y, \Delta Z)$ by solving

$$F_\mu + \nabla F_\mu \cdot (\Delta X, \Delta y, \Delta Z)^T = 0.$$

Here, the step direction can be determined by the linearized system

$$\begin{aligned} \mathcal{A}\Delta X &= -(\mathcal{A}X - b) && (4.3) \\ \mathcal{A}^T \Delta y + \Delta Z &= -(\mathcal{A}^T y + Z - C) && (4.4) \\ \Delta X Z + X \Delta Z &= \mu I - XZ. && (4.5) \end{aligned}$$

Unfortunately, X and Z do not commute, $XZ \neq ZX$, and the same is true in the linearization of the perturbed complementarity condition (quite in contrast to the case of linear programming). In general we cannot expect that there exist symmetric ΔX and ΔZ that solve (4.3) to (4.5).

Solving the system for square matrices $\Delta X, \Delta Z \in M_n$ yields a symmetric ΔZ because of (4.4) but (in general) an unsymmetric ΔX. Since the next iterate $X + \alpha \Delta X$ has to be a symmetric positive definite matrix this is a serious problem. A number of approaches have been proposed to get around this difficulty. We present only three (see Todd [1999] for a survey on search directions).

The first approach (Helmberg, Rendl, Vanderbei, and Wolkowicz [1996]; Kojima, Shindoh, and Hara [1997]; Monteiro [1997]) allows ΔX to be unsymmetric in order to guarantee that the system is solvable. The skew-symmetric part of ΔX is then ignored and the symmetric part constitutes the new step direction,

$$\Delta \hat{X} Z + X \Delta Z = \mu I - XZ, \qquad \Delta X = \frac{\Delta \hat{X} + \Delta \hat{X}^T}{2}. \tag{4.6}$$

The second approach is based on the concept of self-scaled barrier functions which to explain is beyond the scope of this text (see Nesterov and Todd [1997]; Nesterov and Todd [1998]). A special scaling point W satisfying $W^{-\frac{1}{2}} X W^{-\frac{1}{2}} = W^{\frac{1}{2}} Z W^{\frac{1}{2}}$ is used to reformulate the complementarity condition,

$$W^{-1} \Delta X W^{-1} + \Delta Z = \mu X^{-1} - Z. \tag{4.7}$$

The (unique) scaling point reads $W = X^{\frac{1}{2}} (X^{\frac{1}{2}} Z X^{\frac{1}{2}})^{-\frac{1}{2}} X^{\frac{1}{2}}$. Any solution to this system is guaranteed to be symmetric.

In a third approach (Alizadeh, Haeberly, and Overton [1998]) the complementarity condition (4.5) is modified so as to allow for symmetric updates only. Consider the linearization of $XZ + ZX - \mu I$,

$$\Delta X Z + X \Delta Z + Z \Delta X + \Delta Z X = 2\mu I - XZ - ZX. \tag{4.8}$$

Symmetrization is implicit, the existence of symmetric solutions ΔX and ΔZ is guaranteed if X and Z are "close" to the central path.

The three search directions presented can be generalized by introducing a symmetrization operator (Zhang [1998])

$$H_P(M) = \frac{1}{2}(PMP^{-1} + (PMP^{-1})^T) \tag{4.9}$$

with a given nonsingular matrix $P \in S_n$. Symmetrizing the complementarity condition by this operator yields

$$H_P(XZ + \Delta X Z + X \Delta Z) = \mu I. \tag{4.10}$$

The choice $P = I$ corresponds to search direction (4.8). For $P = Z^{\frac{1}{2}}$ it is equivalent to (4.6). It can be worked out that (4.7) is obtained by choosing P so that $P^T P = W$ (see Todd, Toh, and Tütüncü [1998]).

The search directions (4.6), (4.7), and (4.8) are currently the most popular in practical implementations. Other interesting choices may show up in the future. The algorithmic framework for all these methods or even hybrid methods is the same.

Algorithm 4.1.8

Input: *$\mathcal{A}$, b, C, and some starting point (X^0, y^0, Z^0) with $X^0 \succ 0$ and $Z^0 \succ 0$ (usually this starting point will have to satisfy some additional conditions).*

1. *Choose μ.*
2. *Compute $(\Delta X, \Delta y, \Delta Z)$ by solving (4.3), (4.4) together with a variant of (4.5).*
3. *Choose some $\alpha \in (0, 1]$ so that $X + \alpha \Delta X$ and $Z + \alpha \Delta Z$ remain positive definite.*
4. *Set $(X, y, Z) := (X + \alpha \Delta X, y + \alpha \Delta y, Z + \alpha \Delta Z)$.*
5. *If $\|\mathcal{A}X - b\|$ and $\left\|\mathcal{A}^T y + Z - C\right\|_{\mathrm{F}}$ and $\langle X, Z \rangle$ are small enough then* **stop**, *else* **goto** *1.*

In order to prove polynomial iteration complexity for a particular scheme of search directions, rather strong restrictions must be imposed on the starting point and the specific choices of μ and α.

4.2 The Feasible XZ–Method

In this section we give a proof of the polynomial iteration complexity using the XZ–search direction (4.6) for a feasible starting point close to the central path. In the next section we will show how such a starting point can be constructed. The line of argument and most of the proofs follow Zhang [1998], who proved an infeasible variant; the arguments have been adapted to match the feasible case and to parallel the analysis of the linear counterpart of Monteiro and Adler [1989a]. Instead of (4.6) we will work with the symmetrization operator H_P of (4.9) for $P = Z^{\frac{1}{2}}$. We will frequently exploit the following property of this operator.

Proposition 4.2.1 $\langle X, Z\rangle = \langle H_P(XZ), I\rangle$

Proof. $\langle H_P(XZ), I\rangle = \frac{1}{2}(\mathrm{tr}(PXZP^{-1} + (PXZP^{-1})^T) = \mathrm{tr}(XZ)$. ■

We describe the neighborhood of the central path by the distance of the matrix $H_{Z^{\frac{1}{2}}}(XZ)$ to its projection onto the ray αI. The projected point is

$$\left\langle H_{Z^{\frac{1}{2}}}(XZ), \frac{1}{\sqrt{n}}I \right\rangle \frac{1}{\sqrt{n}}I = \frac{\langle X, Z\rangle}{n}I.$$

$\frac{\langle X,Z\rangle}{n}$ may be interpreted as the value of the barrier parameter corresponding to X and Z. This value plays an important role in the analysis and we define

$$\mu(X, Z) := \frac{\langle X, Z\rangle}{n}.$$

We say that (X, Z) is in the neighborhood of the central path if it satisfies

$$\left\| H_{Z^{\frac{1}{2}}}(XZ) - \mu(X, Z)I \right\|_{\mathrm{F}} \leq \theta\mu(X, Z) \tag{4.11}$$

for some constant $0 < \theta < 1$. This condition is equivalent to requiring

$$\left\| \Lambda_{Z^{\frac{1}{2}}XZ^{\frac{1}{2}}} - \mu(X, Z)I \right\|_{\mathrm{F}} \leq \theta\mu(X, Z)$$

which implies that the eigenvalues of $Z^{\frac{1}{2}}XZ^{\frac{1}{2}}$ are bounded by

$$(1 - \theta)\mu(X, Z) \leq \lambda_i(Z^{\frac{1}{2}}XZ^{\frac{1}{2}}) \leq (1 + \theta)\mu(X, Z). \tag{4.12}$$

The algorithm proceeds as follows. It starts at a strictly feasible point (X, y, Z) in the neighborhood of the central path and computes a step direction $(\Delta X, \Delta y, \Delta Z)$ for a barrier parameter $\sigma\mu(X, Z)$ with $\sigma < 1$. σ is the factor by which μ is reduced in each iteration. We will see that σ and θ can be chosen independent of the current iterate such that the new point $(X^+, y^+, Z^+) := (X + \Delta X, y + \Delta y, Z + \Delta Z)$ is again a strictly feasible point in the same neighborhood satisfying $\mu(X^+, Z^+) = \sigma\mu(X, Z)$. It remains to iterate till $\langle X, Z\rangle$ is small enough.

Algorithm 4.2.2

Input: *$\mathcal{A}$, b, C, ε and a strictly feasible starting point $(X^{(0)}, y^{(0)}, Z^{(0)})$ which satisfies (4.11) for appropriately chosen parameters $0 < \theta < 1$ and $0 < \sigma < 1$.*

1. *Set $k = 0$.*
2. *Set $\mu_k := \left\langle X^{(k)}, Z^{(k)}\right\rangle / n$.*
3. *Solve the following system for $(\Delta X, \Delta y, \Delta Z)$.*

$$\begin{aligned} \mathcal{A}\Delta X &= 0 && (4.13)\\ \mathcal{A}^T\Delta y + \Delta Z &= 0 && (4.14)\\ H_{Z^{(k)\frac{1}{2}}}(\Delta X Z^{(k)} + X^{(k)}\Delta Z) &= \sigma\mu_k I - H_{Z^{(k)\frac{1}{2}}}(X^{(k)}Z^{(k)}) && (4.15) \end{aligned}$$

4. $(X^{(k+1)}, y^{(k+1)}, Z^{(k+1)}) := (X^{(k)} + \Delta X, y^{(k)} + \Delta y, Z^{(k)} + \Delta Z)$

5. *If* $\left\langle X^{(k+1)}, Z^{(k+1)} \right\rangle < \varepsilon$ *then* **stop**.

6. *Set* $k := k + 1$ *and* **goto** *step 2.*

We postpone the discussion on how to solve (4.13) to (4.15) to Section 4.4 and concentrate on the correctness and polynomial iteration complexity of the algorithm. Correctness hinges on the choice of θ and σ. Can we find $0 < \theta < 1$ and $\sigma < 1$ such that the next iterate is not only feasible but again a point in the neighborhood of the central path? In the following we will derive the conditions on σ and θ from these restrictions on the next iterate. Once the existence of such parameters is established the proof of polynomial complexity will be easy, because correctness implies (see Lemma 4.2.3 below) that $\mu(X^{(k)}, Z^{(k)})$, and therefore $\left\langle X^{(k)}, Z^{(k)} \right\rangle$, is reduced by σ in each iteration.

As in the linear case we denote the current iterate $(X^{(k)}, y^{(k)}, Z^{(k)})$ by (X, y, Z) and the successor $(X^{(k+1)}, y^{(k+1)}, Z^{(k+1)})$ by (X^+, y^+, Z^+). In the same vein we speak of $H(\cdot) = H_{Z^{\frac{1}{2}}}(\cdot)$, $H_+(\cdot) = H_{(Z^+)^{\frac{1}{2}}}(\cdot)$, $\mu = \mu(X, Z)$, and $\mu_+ = \mu(X^+, Z^+)$.

We start with investigating some properties of (X^+, y^+, Z^+). Obviously the new iterate satisfies the linear constraints, because by strict feasibility of (X, y, Z) and equations (4.13) and (4.14),

$$\begin{aligned} \mathcal{A}X^+ = &\quad \mathcal{A}X + \mathcal{A}\Delta X &= b \\ \mathcal{A}^T(y^+) + Z^+ = &\quad \mathcal{A}^T(y) + Z + \mathcal{A}^T(\Delta y) + \Delta Z &= C. \end{aligned}$$

We cannot expect that the nonlinear complementarity condition behaves as nicely, because the linearization (4.15) ignores the bilinear term $\Delta X \Delta Z$. The next lemma shows that this nonlinear part of the update is of no consequence for $\mu_+ = \mu(X^+, Z^+)$ but has a considerable influence on the distance to the central path.

Lemma 4.2.3

$$\langle \Delta X, \Delta Z \rangle = 0 \tag{4.16}$$

$$\mu_+ = \frac{\langle X^+, Z^+ \rangle}{n} = \sigma\mu \tag{4.17}$$

$$H(X^+Z^+) = \mu_+ I + H(\Delta X \Delta Z) \tag{4.18}$$

Proof. $\langle \Delta X, \Delta Z \rangle = 0$ is immediate from Lemma 4.1.3 and the feasibility of (X, Z) and (X^+, Z^+). Now observe that

$$H(X^+Z^+) = H(XZ + \Delta X Z + X \Delta Z) + H(\Delta X \Delta Z) = \sigma\mu I + H(\Delta X \Delta Z)$$

and, using Proposition 4.2.1, the lemma follows from

$$\left\langle X^+, Z^+ \right\rangle = \left\langle H(X^+Z^+), I \right\rangle = \langle \sigma\mu I + H(\Delta X \Delta Z), I \rangle = n\sigma\mu + \langle \Delta X, \Delta Z \rangle .$$

■

Looking at the characterization (4.18) of $H(X^+Z^+)$ in this lemma it is tempting to use $H(X^+Z^+)$ instead of $H_+(X^+, Z^+)$ in (4.11). The task of establishing that (X^+, Z^+) is close to the central path would then reduce to the condition

$$\left\| H(X^+Z^+) - \mu_+ I \right\|_{\mathrm{F}} = \| H(\Delta X \Delta Z) \|_{\mathrm{F}} \leq \theta\mu_+. \tag{4.19}$$

The following lemma states that this is indeed possible. More precisely, $H_+(X^+Z^+)$ satisfies (4.11) if $H(X^+Z^+)$ does.

Lemma 4.2.4 *For arbitrary* $X, Y, Z \in S_n^{++}$

$$\left\| H_{Z^{\frac{1}{2}}}(XZ) - \mu(X,Z)I \right\|_{\mathrm{F}} \leq \left\| H_{Y^{\frac{1}{2}}}(XZ) - \mu(X,Z)I \right\|_{\mathrm{F}}.$$

Proof. To simplify the inequality we first examine $\|H_P(A) - \alpha I\|_{\mathrm{F}}^2$ for nonsingular $P \in M_n$, $A \in M_n$, and some constant $\alpha > 0$,

$$\langle H_P(A) - \alpha I, H_P(A) - \alpha I \rangle = \|H_P(A)\|_{\mathrm{F}}^2 - 2\alpha \langle H_P(A), I \rangle + \alpha^2 n.$$

Since $2\langle H_P(A), I\rangle = \mathrm{tr}(PAP^{-1}) + \mathrm{tr}(P^{-T}A^TP^T) = 2\,\mathrm{tr}(A)$, it suffices to prove

$$\left\| H_{Z^{\frac{1}{2}}}(XZ) \right\|_{\mathrm{F}}^2 \leq \left\| H_{Y^{\frac{1}{2}}}(XZ) \right\|_{\mathrm{F}}^2.$$

Rearranging and comparing the summands of the corresponding trace expressions this can be further reduced to showing

$$\left\| X^{\frac{1}{2}} Z X^{\frac{1}{2}} \right\|_{\mathrm{F}}^2 \leq \left\| (Y^{\frac{1}{2}} X^{\frac{1}{2}}) X^{\frac{1}{2}} Z X^{\frac{1}{2}} (Y^{\frac{1}{2}} X^{\frac{1}{2}})^{-1} \right\|_{\mathrm{F}}^2.$$

$X^{\frac{1}{2}} Z X^{\frac{1}{2}}$ is a positive definite matrix and on the right hand side the same matrix is subject to a similarity transformation. Thus, the matrices on both sides have the same spectrum Λ. By the Schur Theorem there is an orthonormal matrix $Q \in M_n$ such that the right hand side matrix can be written in the form $Q(\Lambda + R)Q^T$ with R a strictly upper triangular matrix. We obtain

$$\left\| Q(\Lambda + R)Q^T \right\|_{\mathrm{F}}^2 = \|\Lambda + R\|_{\mathrm{F}}^2 = \|\Lambda\|_{\mathrm{F}}^2 + \|R\|_{\mathrm{F}}^2 \geq \|\Lambda\|_{\mathrm{F}}^2 = \left\| X^{\frac{1}{2}} Z X^{\frac{1}{2}} \right\|_{\mathrm{F}}^2$$

which is the desired result. ■

Before proving the existence of θ and σ such that $\|H(\Delta X \Delta Z)\|_{\mathrm{F}} \leq \theta\mu_+$ is satisfied we show that $\|H(\Delta X \Delta Z)\|_{\mathrm{F}} \leq \theta\mu_+$ suffices to prove the positive definiteness of X^+ and Z^+. By (4.19) and (4.12) the eigenvalues of $Z^{(+)\frac{1}{2}} X^+ Z^{(+)\frac{1}{2}}$ must be positive. Does this imply the positive semidefiniteness of X^+ and Z^+? Not yet, because both, X^+ and Z^+, might have negative eigenvalues. To exclude this possibility we need a technical lemma. It shows that if $\|H(\Delta X \Delta Z)\|_{\mathrm{F}} \leq \theta\mu_+$ holds then (4.11) is not only satisfied for the point (X^+, Z^+), but for all $(X + \alpha\Delta X, y + \alpha\Delta y, Z + \alpha\Delta Z)$ with $\alpha \in [0,1]$.

Lemma 4.2.5 *Let* $X(\alpha) = X + \alpha\Delta X$ *and* $Z(\alpha) = Z + \alpha\Delta Z$. *If*

$$\|H(X(\alpha)Z(\alpha)) - \mu(X(\alpha), Z(\alpha))I\|_{\mathrm{F}} \leq \theta\mu(X(\alpha), Z(\alpha))$$

is satisfied for $\alpha = 0$ *and* $\alpha = 1$ *then it is satisfied for all* $\alpha \in [0,1]$.

Proof. Using (4.15), (4.18), and (4.17) we first observe that

$$\begin{aligned} H(X(\alpha)Z(\alpha)) &= H(XZ) + \alpha H(\Delta X Z + X \Delta Z) + \alpha^2 H(\Delta X \Delta Z) \\ &= H(XZ) + \alpha(\sigma\mu I - H(XZ)) + \alpha^2 (H(X^+Z^+) - \mu_+ I) \\ &= (1-\alpha^2)H(XZ) + (\alpha - \alpha^2)(\sigma\mu I - H(XZ)) + \alpha^2 H(X^+Z^+) \\ &= (1-\alpha)H(XZ) + (\alpha - \alpha^2)\sigma\mu I + \alpha^2 H(X^+Z^+). \end{aligned}$$

From (4.16) it follows that $\mu(X(\alpha), Z(\alpha)) = (1-\alpha)\mu + \alpha\mu_+$ and together with $\sigma\mu = \mu_+$ from (4.17) we obtain for $(0 \leq \alpha \leq 1)$

$$\begin{aligned} &\|H(X(\alpha)Z(\alpha)) - (1-\alpha)\mu I - \alpha\mu_+ I\|_{\mathrm{F}}^2 = \\ &= \left\| (1-\alpha)(H(XZ) - \mu I) + \alpha^2 (H(X^+Z^+) - \mu_+ I) \right\|_{\mathrm{F}}^2 \\ &= \|(1-\alpha)(H(XZ) - \mu I)\|_{\mathrm{F}}^2 + \left\| \alpha^2 (H(X^+Z^+) - \mu_+ I) \right\|_{\mathrm{F}}^2 + \\ &\quad 2(1-\alpha)\alpha^2 (H(XZ) - \mu I)^T (H(X^+Z^+) - \mu_+ I) \\ &\leq (1-\alpha)^2\theta^2\mu^2 + \alpha^4\theta^2\mu_+^2 + 2(1-\alpha)\alpha^2 \|H(XZ) - \mu I\|_{\mathrm{F}} \left\| H(X^+Z^+) - \mu_+ I \right\|_{\mathrm{F}} \\ &\leq \theta^2((1-\alpha)\mu + \alpha\mu_+)^2. \end{aligned}$$

This guarantees that

$$\|H(X(\alpha)Z(\alpha)) - \mu(X(\alpha), Z(\alpha))I\|_{\mathrm{F}} \leq \theta\mu(X(\alpha), Z(\alpha))$$

for all $\alpha \in [0,1]$. ■

Lemma 4.2.6 *Let $X(\alpha) = X + \alpha\Delta X$ and $Z(\alpha) = Z + \alpha\Delta Z$. If $\|H(\Delta X \Delta Z)\|_{\mathrm{F}} \leq \theta\mu_+$ then $(X(\alpha), y + \alpha\Delta y, Z(\alpha))$ is strictly feasible for all $\alpha \in [0,1]$.*

Proof. We have to prove that $X(\alpha)$ and $Z(\alpha)$ are positive definite for all $\alpha \in [0,1]$. By (4.19), Lemma 4.2.4, and Lemma 4.2.5 it follows from (4.12) that

$$(1-\theta)\mu_\alpha \leq \lambda_i(Z(\alpha)^{\frac{1}{2}}X(\alpha)Z(\alpha)^{\frac{1}{2}}) \leq (1+\theta)\mu_\alpha \tag{4.20}$$

for $i \in \{1, \ldots, n\}$ and $\alpha \in [0,1]$ with $\mu_\alpha = \mu(X(\alpha), Z(\alpha)) = (1-\alpha)\mu + \alpha\mu_+ > 0$. Recall, that X and Z are positive definite and that eigenvalues depend continuously on matrix entries. Assume, for contradiction, that there exists $0 < \hat{\alpha} \leq 1$ such that $X(\hat{\alpha})$ or $Z(\hat{\alpha})$ are not positive definite and let $\hat{\alpha}$ be the smallest $\hat{\alpha} > 0$ of this kind. But then $X(\hat{\alpha})$ or $Z(\hat{\alpha})$ or both have an eigenvalue zero. If $Z(\hat{\alpha})$ has a zero eigenvector v, then v is also a zero eigenvector of $Z(\hat{\alpha})^{\frac{1}{2}}X(\hat{\alpha})Z(\hat{\alpha})^{\frac{1}{2}}$, which is a contradiction to (4.20). If $Z(\hat{\alpha})$ is positive definite and $X(\hat{\alpha})$ has a zero eigenvector v, then $Z(\hat{\alpha})^{-\frac{1}{2}}v$ is a zero eigenvector of $Z(\hat{\alpha})^{\frac{1}{2}}X(\hat{\alpha})Z(\hat{\alpha})^{\frac{1}{2}}$, which is again a contradiction to (4.20). Thus, $X(\alpha)$ and $Z(\alpha)$ must be positive definite for all $\alpha \in [0,1]$. ■

Summing up, we have proved that the next iterate (X^+, y^+, Z^+) is strictly feasible (Lemma 4.2.6) and satisfies (4.11) (Lemma 4.2.4 and (4.19)) provided we can find $\sigma < 1$ and $0 < \theta < 1$ such that

$$\|H(\Delta X \Delta Z)\|_{\mathrm{F}} \leq \sigma\mu\theta.$$

To bound $\|H(\Delta X \Delta Z)\|_{\mathrm{F}}$ in terms of σ and θ we will have to exploit (4.15). We start by multiplying (4.15) from left and right with $\sqrt{2}Z^{\frac{1}{2}}$.

$$2Z\Delta XZ + ZX\Delta Z + \Delta ZXZ = 2(\sigma\mu Z - ZXZ) \quad =: R \tag{4.21}$$

Kronecker products allow to rewrite this as

$$2(Z \otimes Z)\operatorname{vec}(\Delta X) + (ZX \otimes I + I \otimes ZX)\operatorname{vec}(\Delta Z) = \operatorname{vec}(R).$$

In order to simplify notation we define

$$E = 2(Z \otimes Z) \qquad \text{and} \qquad F = ZX \otimes I + I \otimes ZX. \tag{4.22}$$

Together with Lemma 4.2.8 below the following lemma allows to bound $\|H(\Delta X \Delta Z)\|_{\mathrm{F}}$ in terms of the norm of the scaled right hand side of (4.11).

Lemma 4.2.7 *For nonsingular matrices $E, F \in M_k$ let $x, z, r \in \mathbb{R}^k$ such that $Ex + Fz = r$. If $S = EF^T \succ 0$ then for $D = S^{-\frac{1}{2}}F$*

$$\left\|D^{-T}x\right\|^2 + \|Dz\|^2 + 2\langle x, y\rangle = \left\|S^{-\frac{1}{2}}r\right\|^2.$$

Proof. Multiply $Ex + Fz = r$ with $S^{-\frac{1}{2}}$ from the left and observe that $(S^{-\frac{1}{2}}E)D^T = I$. Take the 2-norm on both sides of $D^{-T}x + Dz = S^{-\frac{1}{2}}r$. Squaring both sides completes the proof. ■

In particular, with E and F from (4.22), S may be written as

$$\begin{aligned} S &= 2(Z\otimes Z)(XZ\otimes I + I\otimes XZ) \\ &= 2(ZXZ\otimes Z) + (Z\otimes ZXZ) \\ &= E^{\frac{1}{2}}(Z^{\frac{1}{2}}XZ^{\frac{1}{2}}\otimes I + I\otimes Z^{\frac{1}{2}}XZ^{\frac{1}{2}})E^{\frac{1}{2}} \\ &= E^{\frac{1}{2}}(Q\otimes Q)(\Lambda\otimes I + I\otimes\Lambda)(Q^T\otimes Q^T)E^{\frac{1}{2}}, \end{aligned}$$

with

$$Z^{\frac{1}{2}}XZ^{\frac{1}{2}} = Q\Lambda Q^T. \tag{4.23}$$

In the remainder of this section Λ will always refer to the spectrum of $Z^{\frac{1}{2}}XZ^{\frac{1}{2}}$, λ_i to its i-th eigenvalue, and Q to the matrix containing the corresponding eigenvectors. For later use we define

$$\hat{F} = (Q\otimes Q)(\Lambda\otimes I + I\otimes\Lambda)(Q^T\otimes Q^T) \tag{4.24}$$

and have

$$ES^{-1}E = E^{\frac{1}{2}}\hat{F}E^{\frac{1}{2}}. \tag{4.25}$$

Since $\Lambda\otimes I + I\otimes\Lambda$ is diagonal with diagonal elements $\lambda_i+\lambda_j$, $\hat{F}$ is symmetric and positive definite with eigenvalues $\lambda_i+\lambda_j$. Consequently S is positive definite, as well. The orthogonality of ΔX and ΔZ carries over to the vector representation,

$$\langle \operatorname{vec}(\Delta X), \operatorname{vec}(\Delta Z)\rangle = \langle \Delta X, \Delta Z\rangle = 0.$$

Therefore Lemma 4.2.7 yields the important relation

$$\left\| D^{-T}\operatorname{vec}(\Delta X)\right\|^2 + \|D\operatorname{vec}(\Delta Z)\|^2 = \left\|S^{-\frac{1}{2}}\operatorname{vec}(R)\right\|^2. \tag{4.26}$$

The next lemma establishes a bound on $\|H(\Delta X\Delta Z)\|_{\mathrm{F}}$ in terms of the left hand side of this equation. This will enable us to bound $\|H(\Delta X\Delta Z)\|_{\mathrm{F}}$ independent of ΔX and ΔZ via the right hand side of the equation.

Lemma 4.2.8

$$\|H(\Delta X\Delta Z)\|_{\mathrm{F}} \le \frac{1}{2}\sqrt{\frac{1+\theta}{1-\theta}}\left(\left\| D^{-T}\operatorname{vec}(\Delta X)\right\|^2 + \|D\operatorname{vec}(\Delta Z)\|^2\right)$$

Proof. First we split $\|H(\Delta X\Delta Z)\|_{\mathrm{F}}$ into two independent terms,

$$\begin{aligned} \|H(\Delta X\Delta Z)\|_{\mathrm{F}} &= \frac{1}{2}\left\| Z^{\frac{1}{2}}\Delta X\Delta Z Z^{-\frac{1}{2}} + (Z^{\frac{1}{2}}\Delta X\Delta Z Z^{-\frac{1}{2}})^T\right\|_{\mathrm{F}} \\ &\le \left\| Z^{\frac{1}{2}}\Delta X\Delta Z Z^{-\frac{1}{2}}\right\|_{\mathrm{F}} \\ &= \left\| Z^{\frac{1}{2}}\Delta X Z^{\frac{1}{2}}Z^{-\frac{1}{2}}\Delta Z Z^{-\frac{1}{2}}\right\|_{\mathrm{F}} \\ &\le \left\| Z^{\frac{1}{2}}\Delta X Z^{\frac{1}{2}}\right\|_{\mathrm{F}}\left\| Z^{-\frac{1}{2}}\Delta Z Z^{-\frac{1}{2}}\right\|_{\mathrm{F}}. \end{aligned}$$

Recalling from the proof of Lemma 4.2.7 that $D^{-T} = S^{-\frac{1}{2}}E$ we bound each term separately using (4.24) and (4.25),

$$\begin{aligned} \left\| D^{-T}\operatorname{vec}(\Delta X)\right\|^2 &= \operatorname{vec}(\Delta X)^T ES^{-1}E\operatorname{vec}(\Delta X) \\ &= \operatorname{vec}(\Delta X)^T E^{\frac{1}{2}}\hat{F}^{-1}E^{\frac{1}{2}}\operatorname{vec}(\Delta X) \\ &= 2\operatorname{vec}(Z^{\frac{1}{2}}\Delta X Z^{\frac{1}{2}})^T\hat{F}^{-1}\operatorname{vec}(Z^{\frac{1}{2}}\Delta X Z^{\frac{1}{2}}) \\ &\ge 2\frac{1}{\lambda_{\max}(\hat{F})}\left\| Z^{\frac{1}{2}}\Delta X Z^{\frac{1}{2}}\right\|_{\mathrm{F}}^2 \\ &\ge \frac{1}{(1+\theta)\mu}\left\| Z^{\frac{1}{2}}\Delta X Z^{\frac{1}{2}}\right\|_{\mathrm{F}}^2. \end{aligned}$$

The last line follows from $\lambda_{\max}(\hat{F}) = 2\lambda_{\max}(Z^{\frac{1}{2}}XZ^{\frac{1}{2}})$ which can be bounded by (4.12). Likewise we obtain

$$\left\|D^T \operatorname{vec}(\Delta Z)\right\|^2 \geq (1-\theta)\mu \left\|Z^{-\frac{1}{2}}\Delta X Z^{-\frac{1}{2}}\right\|_{\mathrm{F}}^2.$$

Putting things together we get

$$\begin{aligned}\|H(\Delta X \Delta Z)\|_{\mathrm{F}} &\leq \sqrt{\frac{1+\theta}{1-\theta}}\left\|D^{-T}\operatorname{vec}(\Delta X)\right\| \left\|D^T \operatorname{vec}(\Delta Z)\right\| \\ &\leq \frac{1}{2}\sqrt{\frac{1+\theta}{1-\theta}}\left(\left\|D^{-T}\operatorname{vec}(\Delta X)\right\|^2 + \left\|D^T\operatorname{vec}(\Delta Z)\right\|^2\right).\end{aligned}$$

The last inequality follows from the well known relation $\frac{1}{2}(a-b)^2 \geq 0$. ■

In view of Lemma 4.2.8 and (4.26) it remains to bound $\left\|S^{-\frac{1}{2}}\operatorname{vec}(R)\right\|^2$ to achieve our goal of bounding $\|H(\Delta X \Delta Z)\|_{\mathrm{F}}$ in terms of σ and θ.

Lemma 4.2.9 *Let $\lambda_1, \ldots, \lambda_n$ denote the eigenvalues of $Z^{\frac{1}{2}}XZ^{\frac{1}{2}}$, then*

$$\left\|S^{-\frac{1}{2}}\operatorname{vec}(R)\right\|^2 = \sum_{i=1}^{n}\frac{(\sigma\mu-\lambda_i)^2}{\lambda_i} \leq \frac{\|\Lambda-\sigma\mu I\|_{\mathrm{F}}^2}{(1-\theta)\mu}.$$

Proof. By (4.21) and (4.23) we have

$$R = 2Z^{\frac{1}{2}}(\sigma\mu I - Z^{\frac{1}{2}}XZ^{\frac{1}{2}})Z^{\frac{1}{2}} = 2Z^{\frac{1}{2}}Q(\sigma\mu I-\Lambda)Q^T Z^{\frac{1}{2}}.$$

Applying the vec-operator and rule (1.8) for Kronecker products yields

$$\operatorname{vec}(R) = \sqrt{2}E^{\frac{1}{2}}(Q\otimes Q)\operatorname{vec}(\sigma\mu I-\Lambda).$$

Therefore

$$\begin{aligned}\left\|S^{-\frac{1}{2}}\operatorname{vec}(R)\right\|^2 &= \operatorname{vec}(R)^T S^{-1}\operatorname{vec}(R) \\ &= 2\operatorname{vec}(\sigma\mu I-\Lambda)^T(\Lambda\otimes I + I\otimes\Lambda)^{-1}\operatorname{vec}(\sigma\mu I-\Lambda) \\ &= \sum_{i=1}^{n}\frac{(\sigma\mu-\lambda_i)^2}{\lambda_i}.\end{aligned}$$

The inequality is obtained by bounding the smallest eigenvalue using (4.12). ■

We are ready to specify for which choices of θ and σ the iterates remain feasible and close to the central path.

Theorem 4.2.10 *If the constants $0<\theta<1$ and $0<\sigma$ are chosen such that*

$$\frac{(1+\theta)^{\frac{1}{2}}}{2(1-\theta)^{\frac{3}{2}}}(\theta^2 + n(1-\sigma)^2) \leq \sigma\theta,$$

then $\|H_+(X^+Z^+)-\mu_+ I\|_{\mathrm{F}} \leq \theta\mu_+$ is satisfied in each iteration.

Proof. So far we know from Lemma 4.2.4 that

$$\left\|H_+(X^+Z^+)-\mu_+ I\right\|_{\mathrm{F}} \leq \left\|H(X^+Z^+)-\mu_+ I\right\|_{\mathrm{F}}$$

and from (4.18), Lemma 4.2.8, (4.26), and Lemma 4.2.9 that

$$\left\|H(X^+Z^+)-\mu_+ I\right\|_{\mathrm{F}} = \|H(\Delta X\Delta Z)\|_{\mathrm{F}} \leq \frac{(1+\theta)^{\frac{1}{2}}}{2(1-\theta)^{\frac{3}{2}}\mu}\|\Lambda-\sigma\mu I\|_{\mathrm{F}}^2.$$

To bound the right hand side we use Pythagoras and (4.11),

$$\begin{aligned} \|\Lambda - \sigma\mu I\|_{\mathrm{F}}^2 &\leq \|\Lambda - \mu I\|_{\mathrm{F}}^2 + \|\mu I - \sigma\mu I\|_{\mathrm{F}}^2 \\ &\leq \|H(X,Z) - \mu I\|_{\mathrm{F}}^2 + n(1-\sigma)^2\mu^2 \\ &\leq (\theta^2 + n(1-\sigma)^2)\mu^2 \end{aligned}$$

The Lemma follows if we require that the right hand side is no greater than $\theta\sigma\mu$. ■

We substitute $1 - \delta/\sqrt{n}$ for σ and obtain the condition

$$\frac{(1+\theta)^{\frac{1}{2}}}{2(1-\theta)^{\frac{3}{2}}}(\theta^2 + \delta^2) \leq \theta(1 - \frac{\delta}{\sqrt{n}}). \tag{4.27}$$

One possible choice is $\theta = \delta = 3.2$. This yields the final result.

Theorem 4.2.11 *For $\sigma = 1 - \frac{\delta}{\sqrt{n}}$ with $\delta > 0$ satisfying (4.27) and a starting point (X^0, y^0, Z^0) satisfying (4.11) Algorithm 4.2.2 terminates in $O(\sqrt{n}\log(\langle X^0, Z^0\rangle/\varepsilon))$ iterations.*

Proof. The algorithm terminates when $n\mu_k = n\sigma^k\mu_0 \leq \varepsilon$. We have $k\ln\sigma = k\ln(1-\frac{\delta}{\sqrt{n}}) \leq -k\frac{\delta}{\sqrt{n}}$. Therefore any $\bar{k}$ satisfying

$$\bar{k}\frac{\delta}{\sqrt{n}} \geq -\ln\varepsilon + \ln(n\mu_0)$$

is an upper bound for the number of iterations. Multiplication with $\frac{\sqrt{n}}{\delta}$ proves the theorem. ■

Note, that in contrast to linear programming it does not make sense to give the required precision in terms of the encoding length because of Example 2.3.2. It is surprising that the number of iterations depends on $\sqrt{n}$ as in linear programming. In fact, the number of iterations grows only with the fourth root of the number of variables, because in semidefinite programming the number of variables equals $\binom{n+1}{2}$ (this is connected to the self concordancy parameter of the barrier function $-\log\det(\cdot)$, see Nesterov and Nemirovskii [1994]). We will see in Section 4.4 that the computational cost of each step is considerably higher than in linear programming.

4.3 Centered Starting Points

Algorithm 4.2.2 requires a strictly feasible starting point that is close to the central path. In general such a point is not known and may, in fact, not exist at all. Inspired by the self dual skew-symmetric embedding for linear programming several authors have independently proposed a skew-symmetric self-dual embedding for semidefinite programs. We follow the exposition by de Klerk, Roos, and Terlaky [1997]. The embedding has a centered starting point and a trivial optimal solution. The convergence of the iterates of an interior point algorithm to a maximally complementary solution of the embedding can be exploited to detect whether the original problem is well behaved. In this case the solution also yields the solution to the original problem.

We start with homogenizing the system of optimality conditions derived from strong duality by introducing two artificial variables τ and ρ.

$$\begin{array}{cccccc} & \mathcal{A}X & -\tau b & & & = 0 \\ -\mathcal{A}^T y & & +\tau C & -Z & & = 0 \\ b^T y & -\langle C, X\rangle & & & -\rho & = 0 \\ & X \succeq 0, & \tau \geq 0, & Z \succeq 0, & \rho \geq 0. & \end{array}$$

This system is feasible since it is satisfied by the all zero solution. Assume that there is a solution with $\tau > 0$. Then dividing X, y, and Z by τ yields feasible primal and dual solutions. Furthermore these solutions are optimal because the third constraint of the system above is equivalent to $b^T y \geq \langle C, X\rangle$ which is the opposite of weak duality. On the other hand we know that for feasible

primal and dual solutions it is impossible that $b^T y > \langle C, X \rangle$. Consequently the slack variable ρ must be zero for any feasible solution. This system never has a strictly feasible point. To obtain strictly feasible starting points while maintaining the skew-symmetry of the system two more variables ϑ and σ together with some new constants are introduced. This forms a new semidefinite program.

$$
\begin{array}{llllllll}
\min \vartheta\beta & \text{s.t.} & & & & & & \\
 & \mathcal{A}X & -\tau b & +\vartheta\bar{b} & & & = & 0 \\
-\mathcal{A}^T y & & +\tau C & -\vartheta\bar{C} & -Z & & = & 0 \\
b^T y & -\langle C, X \rangle & & +\vartheta\alpha & & -\rho & = & 0 \\
-\bar{b}^T y & +\langle \bar{C}, X \rangle & -\tau\alpha & & & & -\sigma = & -\beta \\
 & X \succeq 0, & \tau \geq 0, & \vartheta \geq 0, & Z \succeq 0, & \rho \geq 0, & \sigma \geq 0, &
\end{array}
$$

with

$$
\begin{aligned}
\bar{b} &:= -\mathcal{A}I + b \\
\bar{C} &:= C - I \\
\alpha &:= \langle C, I \rangle + 1 \\
\beta &:= n + 2.
\end{aligned}
$$

$\bar{b}$, $\bar{C}$, α, and β are chosen such that the point $y^0 = 0$, $X^0 = Z^0 = I$, $\tau^0 = \vartheta^0 = \rho^0 = \sigma^0 = 1$ yields a strictly feasible point for this semidefinite program. Moreover, after some rearrangements the dual to this program turns out to be exactly this program again. It is self-dual. In particular the complementary semidefinite variable to X is Z and vice versa. The other complementary variable pairs are τ and ρ, and ϑ and σ. Since the constraints are identical, the same starting point can be chosen. Primal and dual variables will be updated in exactly the same way. Therefore it is sufficient to keep just one set of variables. In the KKT system for primal-dual interior point methods we only need the complementarity conditions in addition to the set of constraints above,

$$
\begin{aligned}
XZ &= \mu I \\
\tau\rho &= \mu \\
\vartheta\sigma &= \mu.
\end{aligned}
$$

The additional computational burden is therefore rather small. The strictly feasible point given above satisfies these perturbed complementarity conditions for $\mu = \mu(X^0, Z^0) = 1$. Thus, it is a point on the central path satisfying (4.11) and may be used to start Algorithm 4.2.2. Obviously setting $\sigma = \beta$ and all other variables to zero yields an optimal solution. Why does it make sense to solve this program? The interior point code will deliver a maximally complementary solution. This maximally complementary solution renders this embedding useful.

Theorem 4.3.1 *(de Klerk, Roos, and Terlaky [1997]) Let $(X^*, y^*, Z^*, \rho^*, \sigma^*, \tau^*, \vartheta^*)$ be a maximally complementary optimal solution to the skew-symmetric embedding. Then the following cases can be distinguished:*

(i) $\tau > 0 \Longleftrightarrow$ *Both (PSDP) and (DSDP) are feasible, the duality gap is zero, the optimal values are attained for both problems.*

(ii) $\tau = 0 \wedge \rho > 0 \Longleftrightarrow$ *At least one of the problems has an improving ray.*

(iii) $\tau = 0 \wedge \rho = 0 \Longleftrightarrow$ *Either both are feasible with strictly positive duality gap or the optimal value is not attained for both problems or not both are feasible, none of the problems has an improving ray.*

Proof. As observed above setting $\sigma = \beta$ and all other variables to zero yields an optimal solution. Hence ϑ^* must be zero.

(i) If $\tau^* > 0$, then dividing X^*, y^*, and Z^* by τ^* yields optimal primal and dual solutions. Conversely, let X^*, and (y^*, Z^*) be primal and dual optimal solutions. Set $\vartheta = \rho = 0$ and let $\bar{X} = \tau X^*$, $\bar{y} = \tau y^*$ and $\bar{Z} = \tau Z^*$. Obviously the first three constraints are valid for all choices of τ. The fourth constraint multiplied by -1 reads

$$\langle b - \mathcal{A}I, \bar{y}\rangle - \left\langle C - I, \bar{X}\right\rangle + \tau(\langle C, I\rangle + 1) + \sigma = n + 2.$$

Because of $\langle b, \bar{y}\rangle = \left\langle C, \bar{X}\right\rangle$ and $-\mathcal{A}^T\bar{y} + \tau C = \bar{Z}$ we get

$$\left\langle I, \bar{X}\right\rangle + \left\langle I, \bar{Z}\right\rangle + \tau + \sigma = n + 2.$$

This is feasible for $0 \leq \tau \leq (n+2)/(1 + \operatorname{tr}(X^* + Z^*))$. Therefore any maximally complementary optimal solution must have $\tau^* > 0$ in this case.

(ii) If $\tau^* = 0$ and $\rho > 0$ then $\mathcal{A}X^* = 0$, $\mathcal{A}^T y^* \preceq 0$ and $b^T y^* > \langle C, X^*\rangle$. The last condition implies that at least one of the conditions $b^T y^* > 0$ or $\langle C, X^*\rangle < 0$ must be true. At least one of both problems must be infeasible. We show the other direction for a primal ray $\bar{X} \succeq 0$ with $\mathcal{A}\bar{X} = 0$ and $\left\langle C, \bar{X}\right\rangle < 0$ which is scaled such that $\left\langle I - C, \bar{X}\right\rangle \leq n + 2$. For this $\bar{X}$ the constraints are satisfied choosing for $y = 0$, $Z = 0$, $\tau = \vartheta = 0$, $\rho = -\theta\left\langle C, \bar{X}\right\rangle$, and $\sigma = n + 2 - \left\langle I - C, \bar{X}\right\rangle$. Since $\rho > 0$ any maximally complementary solution must have $\rho > 0$ as well.

(iii) follows from the previous equivalences. ■

We point out that the fourth equality constraint of the skew-symmetric embedding is not only needed for symmetry purposes. As can be seen in the proof it also serves to normalize τ. Without this constraint τ would either be zero or plus infinity.

For practical applications any problem not containing a strictly feasible point in both, primal and dual, programs may be considered ill posed. Small perturbations of the data may lead to large changes of objective value or to the existence of "strictly feasible" solutions. The absence of strictly feasible solutions is usually an indication that the problem formulation is insufficient. In this sense the skew-symmetric embedding offers everything that is asked for in practice. It yields optimal solutions for well posed problems and gives a certificate for infeasibility in most relevant cases.

4.4 Implementational Considerations

In this section we look at some practical aspects of interior point methods in semidefinite programming, mainly drawing from our experience with combinatorial applications (Helmberg, Rendl, Vanderbei, and Wolkowicz [1996]; Helmberg [1994]; Helmberg and Rendl [1998]; Helmberg, Rendl, and Weismantel [2000], see also Chapter 6). We discuss how step direction (4.6) can be computed efficiently and analyze its complexity for some typical settings. Practical algorithms do not stay close to the central path, line searches are necessary to guarantee the positive definiteness of the iterates. We present some empirical rules for implementing line searches and choosing μ. With reasonable computational effort the quality of the step direction can be improved considerably by the predictor-corrector approach. Finally, we demonstrate how the algorithm is generalized to several semidefinite variables of varying dimensions and show that it specializes to the interior point code for linear programming if the dimension of each semidefinite variable is one.

In a slight abuse of notation, we will apply $\mathcal{A}(\cdot)$ to arbitrary square matrices. The assumption, that all coefficient matrices A_i are symmetric, ensures that for any matrix $M \in M_n$

$$\mathcal{A}M = \mathcal{A}M^T,$$

because the skew-symmetric part of M is mapped to zero. In order to compute (4.6) we would like to solve the following system for $\Delta\hat{X} \in M_n$, $\Delta y \in \mathbb{R}^m$, $\Delta Z \in S_n$.

$$\begin{aligned}
\mathcal{A}\Delta\hat{X} &= -(\mathcal{A}X - b) =: F_p && (4.28)\\
\mathcal{A}^T\Delta y + \Delta Z &= -(\mathcal{A}^T y + Z - C) =: F_d && (4.29)\\
\Delta\hat{X}Z + X\Delta Z &= \mu I - XZ. && (4.30)
\end{aligned}$$

We first express $\Delta\hat{X}$ in terms of ΔZ using (4.30), then ΔZ is written in terms of Δy via (4.29). Together with (4.28) this yields a system depending on Δy only.

$$\begin{aligned} \Delta\hat{X} &= \mu Z^{-1} - X - X\Delta Z Z^{-1} \\ \Delta Z &= F_d - \mathcal{A}^T\Delta y \\ \mathcal{A}(X\mathcal{A}^T(\Delta y)Z^{-1}) &= F_p - \mathcal{A}(\mu Z^{-1} - X + XF_dZ^{-1}) = b - \mathcal{A}(\mu Z^{-1} + XF_dZ^{-1}) \end{aligned}$$

Having solved this last equation for Δy we obtain the final step direction $(\Delta X, \Delta y, \Delta Z)$ by back substitution and setting $\Delta X = \frac{\Delta\hat{X}+\Delta\hat{X}^T}{2}$.

Before investigating the computational cost of determining the step direction let us examine the matrix $M = \mathcal{A}(X\mathcal{A}^T(\cdot)Z^{-1})$ in detail. Row i of $My = \mathcal{A}(X\mathcal{A}^T(y)Z^{-1})$ is computed via

$$\langle A_i, X\mathcal{A}^T(y)Z^{-1}\rangle = \left\langle XA_iZ^{-1}, \sum_{j=1}^m y_jA_j \right\rangle = \sum_{j=1}^m y_i \operatorname{tr}(XA_iZ^{-1}A_j).$$

Therefore the ij-th element of M is equal to

$$M_{ij} = \operatorname{tr}(XA_iZ^{-1}A_j). \tag{4.31}$$

Since A_j is symmetric, $\operatorname{tr}(XA_iZ^{-1}A_j) = \operatorname{tr}(Z^{-1}A_iXA_j)$ and consequently $M_{ij} = M_{ji}$. But M is not only symmetric, it is positive semidefinite:

$$\begin{aligned} y^TMy &= \langle y, \mathcal{A}(X\mathcal{A}^T(y)Z^{-1})\rangle = \langle \mathcal{A}^Ty, X\mathcal{A}^T(y)Z^{-1}\rangle \\ &= \left\langle X^{\frac{1}{2}}\mathcal{A}^T(y)Z^{-\frac{1}{2}}, X^{\frac{1}{2}}\mathcal{A}^T(y)Z^{-\frac{1}{2}}\right\rangle \geq 0. \end{aligned}$$

If we assume that $\mathcal{A}(\cdot)$ has full row rank (this can always be achieved) then M is easily seen to be positive definite. In this case there is a unique solution $(\Delta\hat{X}, \Delta y, \Delta Z)$ of (4.28) to (4.30) yielding a unique step direction $(\Delta X, \Delta y, \Delta Z)$ (Zhang [1998] shows that even without this assumption the system can be solved consistently if $\mathcal{A}X = b$ is consistent but we will not delve into this). It can be worked out that $(\Delta X, \Delta y, \Delta Z)$ is the same step direction as in Algorithm 4.2.2. In the following we assume that $M \in S_m^{++}$.

Solving for Δy requires the factorization of M. Since M is positive definite this can be done efficiently in $m^3/3$ arithmetic operations using Cholesky decomposition (see Theorems 1.1.10 and 1.1.14). However, M has to be recomputed in each iteration. An efficient way to build one row of M is to compute XA_iZ^{-1} in $O(n^3)$ once and to determine the single elements via $\langle XA_iZ^{-1}, A_j\rangle$ for $O(n^2)$ each. In total the construction of M requires $O(mn^3 + m^2n^2)$ arithmetic operations. In general one assumes that $m = O(n)$ and so the construction of M is by far the most expensive operation within one iteration.

In many applications the constraint matrices A_i have special structure that can be exploited to speed up the computation of M. In particular in combinatorial applications many constraints can be represented as positive semidefinite rank one matrices, *i.e.*, $A_i = v^{(i)}v^{(i)T}$ with $v^{(i)} \in \mathbb{R}^n$ for $i \in \{1, \ldots, m\}$. In this case it is more efficient to compute, instead of XA_iZ^{-1}, two temporary vectors $v_X = Xv^{(i)}$ and $v_Z = Z^{-1}v^{(i)}$, and to determine the single elements via $v_Z^TA_jv_X = (v_Z^Tv^{(j)})(v^{(j)T}v_X)$. This reduces computation time to $O(mn^2 + m^2n)$. If in addition the vectors $v^{(i)}$ are sparse, the factorization of M is the most expensive operation within each iteration, because we cannot expect M to be sparse. The necessity to store and factorize the dense matrix M limits the size of problems solvable by interior point methods.

Benson, Ye, and Zhang [2000] propose a dual scaling algorithm that offers reasonable possibilities to exploit structure in the dual slack matrix. Unfortunately, their approach cannot avoid that M (in this case $M_{ij} = \operatorname{tr} A_iZ^{-1}A_jZ^{-1}$) is dense, so it is again restricted to a moderate number of constraints.

As in linear programming most practical implementations do not keep the iterates close to the central path. It is therefore necessary to guarantee that the next iterate is again positive

definite. This is usually achieved by a line search for $X + \alpha\Delta X$ and $Z + \alpha\Delta Z$ using Cholesky decomposition. If Cholesky fails the matrix is not positive definite and α is diminished. Typically separate step sizes are used for X and Z, the step size of the last iteration is used as an initial guess. If the initial guess is good α is increased by a factor of $5/4$, say, otherwise it is decreased by $4/5$. This is repeated as long as the matrix is positive definite (and $\alpha < 1$) or not positive definite, respectively. In practical experiments it proved useful not to get too close to the boundary. If α yields a positive semidefinite matrix that is "close" to singular then a step size of $\alpha \cdot 9/10$ seems advisable. Each check for positive definiteness involves $n^3/3$ arithmetic operations and is therefore rather expensive. A strong reduction of μ may not only yield a poor step direction but also decreases the predictability of the correct step size. This may slow down the algorithm considerably. So far there is no good general strategy for choosing μ. A rule on the save side is to choose $\mu/2$ if the minimum of the last step sizes is below $4/5$ and to choose $\mu/10$, otherwise. Several authors prefer to determine the maximal feasible step length for X and ΔX (or Z and ΔZ) via an eigenvalue computation (see Section 6.3.1) but we consider this computationally too expensive.

Most search directions allow the application of the predictor-corrector approach. In a first step, a search direction is computed for $\mu = 0$, the so called affine direction.

$$\begin{aligned} \mathcal{A}\Delta\hat{X} &= -(\mathcal{A}X - b) \\ \mathcal{A}^T\Delta\hat{y} + \Delta\hat{Z} &= -(\mathcal{A}^T y + Z - C) \\ \Delta\hat{X}Z + X\Delta\hat{Z} &= -XZ. \end{aligned}$$

In a second step the information of the predicted point is included in a new right hand side and a second step direction is computed for some choice of μ yielding the so called centering direction.

$$\begin{aligned} \mathcal{A}\Delta\bar{X} &= 0 \\ \mathcal{A}^T\Delta\bar{y} + \Delta\bar{Z} &= 0 \\ \Delta\bar{X}Z + X\Delta\bar{Z} &= \mu I - \Delta\hat{X}\Delta\hat{Z}. \end{aligned}$$

Afterwards both search directions are added for the final step direction

$$(\Delta X, \Delta y, \Delta Z) = \left(\frac{(\Delta\hat{X} + \Delta\bar{X}) + (\Delta\hat{X} + \Delta\bar{X})^T}{2}, \Delta\hat{y} + \Delta\bar{y}, \Delta\hat{Z} + \Delta\bar{Z} \right).$$

This step direction is usually of much better quality than then the canonical step direction. It is not very expensive to compute because the same factorization of M can be used twice, only the right hand side changes. However, its computation involves a few additional $O(n^3)$ operations and therefore predictor-corrector should not be applied for small m, say, $m < 2n$.

Infeasible methods try to cope with infeasibility by including this information in the computation of the step direction. In theory this infeasible approach has polynomial iteration complexity if the starting point satisfies a number of conditions and if μ and α are chosen in conformity with the decrease in infeasibility (Kojima, Shindoh, and Hara [1997]; Zhang [1998]). Infeasibility is detected by observing that the norm of either X or Z gets too large. A precise definition of "too large" is still missing, although there are some promising results in this direction (Nesterov, Todd, and Ye [1999]).

Current comparisons of infeasible methods versus the skew-symmetric embedding show no significant advantage of one above the other, see Toh, Todd, and Tütüncü [1998].

With respect to the different step directions some computational experience is at hand (see Alizadeh, Haeberly, and Overton [1998]; Toh, Todd, and Tütüncü [1998]). (4.6) is faster to compute, but a considerable loss of accuracy is observed as the gap between primal and dual solutions closes. Ususally, a relative precision of 10^{-6} can be achieved. In most cases this suffices for combinatorial applications. (4.7) is a bit slower than (4.6) and is comparable in terms of accuracy. (4.8) takes about twice the time of (4.6) but may allow considerably more accurate solutions (10^{-12}).

Finally, we examine the case of several semidefinite matrix variables. Let the primal problem be given as follows.

$$\begin{array}{ll} \min & \sum_{i=1}^{k} \langle C_i, X_i \rangle \\ \text{s.t.} & \sum_{i=1}^{k} \mathcal{A}_i(X_i) = b \\ & X_i \in S^+_{n_1}, \ldots, X_k \in S^+_{n_k}. \end{array}$$

In this case the KKT-system (4.1) takes the form

$$\begin{array}{rcll} \sum_{i=1}^{k} \mathcal{A}_i(X_i) & = & b & \\ \mathcal{A}_i^T(y) + Z_i & = & C_i & i = 1, \ldots, k \\ X_i Z_i & = & \mu I_{n_i} & i = 1, \ldots, k. \end{array}$$

The linearization of this system can be solved in the same manner as for one semidefinite variable. This leads to a generalized formula for M.

$$M = \sum_{i=1}^{k} \mathcal{A}_i(X_i \mathcal{A}_i^T(\cdot) Z_i^{-1}).$$

Clearly, it is much faster to work with all variables separately than to glue them all together into one big matrix. Although in principle it is possible to choose a different μ for each variable, we recommend to select one μ for all variables so that all converge at approximately the same rate.

Note that if all variables have dimension one, $n_i = 1$ for $i \in \{1, \ldots, k\}$, this is linear programming. In particular for $x, z \in \mathbb{R}^k$ with $x_i = X_i$ and $z_i = Z_i$, M takes the canonical form $A \operatorname{Diag}(x) \operatorname{Diag}(z^{-1}) A^T$, and the algorithm specializes to the canonical interior point approach for linear programming.

4.5 Remarks on the Literature

A detailed account on the primal-dual interior point approach to semidefinite programming is given in Sturm [1997] and on interior point algorithms in general in Nesterov and Nemirovskii [1994]; Ye [1997c].

The first use of the logarithmic barrier function in context with convex programming is attributed to Frisch [1955]. With the classical book of Fiacco and McCormick [1968] and their SUMT algorithm (Sequential Unconstrained Minimization Technique) interior point algorithms became a standard tool in nonlinear optimization.

Many developments in interior point methods for semidefinite programming are based on approaches developed for linear programming before. In 1984 Karmarkar published his seminal paper "A new polynomial-time algorithm for linear programming" (Karmarkar [1984]), proving polynomial complexity for a potential reduction algorithm for linear programming. Gill, Murray, Saunders, Tomlin, and Wright [1986] established the connection of Karmarkar's algorithm to Newton's method and the classical interior point approach of Fiacco and McCormick. The analytic center of a polytope and the central path was introduced by Sonnevend [1985]. By staying close to the central path Renegar [1988] achieved an iteration complexity of $O(\sqrt{n}L)$ for linear programming. Megiddo [1989] paved the way for the primal-dual approach and Kojima, Mizuno, and Yoshise [1989b] developed the first primal-dual algorithm with $O(\sqrt{n}L)$ iteration complexity. Immediately afterwards the same authors Kojima, Mizuno, and Yoshise [1989a] and Monteiro and Adler [1989a]; Monteiro and Adler [1989b] established an $O(n^3 L)$ arithmetic operations bound for this algorithm. Some aspects of the convergence proof presented here are based on Monteiro and Adler [1989a]. The currently best bound in linear programming of $O(n^3 L / \log n)$ is due to Anstreicher [1999].

To the best of our knowledge the first interior point methods designed for semidefinite programming were proposed by Nesterov and Nemirovskii [1994], Jarre [1993], and Alizadeh [1995]. The book of Nesterov and Nemirovskii [1994] contains a special section on semidefinite programming. They apply their framework of strongly self concordant functions to derive pure primal and dual interior point algorithms. Jarre gives a pure dual algorithm using the theory of self concordancy (Nesterov and Nemirovskii [1994]), Alizadeh shows a generic way of adapting pure primal or pure dual interior point algorithms for linear programming to semidefinite programming (the paper also includes a survey on combinatorial applications). Vandenberghe and Boyd [1995] combine a pure primal and a pure dual step direction in one algorithm to achieve a primal-dual behavior.

The first primal-dual algorithms were independently proposed by Helmberg, Rendl, Vanderbei, and Wolkowicz [1996] and Kojima, Shindoh, and Hara [1997]. Both use search direction (4.6), Kojima et al. provide a proof of the polynomial iteration complexity for feasible and infeasible start variants (in fact, they prove this for the class of semidefinite linear complementarity problems). In the revision Kojima, Shindoh, and Hara [1997] generalize the approach to orthogonal skew-symmetric subspaces for ΔX and ΔZ.

Search direction (4.8) was introduced in Alizadeh, Haeberly, and Overton [1994]. Monteiro [1998] proved its polynomial complexity by proving the polynomial complexity of short step path following algorithms for the family of directions (4.10). Other interesting properties of this search direction have been established in Kojima, Shida, and Shindoh [1999]; Alizadeh, Haeberly, and Overton [1998]; Monteiro and Zanjacomo [1997].

In Nesterov and Todd [1997] and Nesterov and Todd [1998] the authors develop a general framework for self scaled cones. The step direction (4.7) for semidefinite programming is a special case of this theory. It was rediscovered in an independent approach by Sturm and Zhang [1999]. Todd, Toh, and Tütüncü [1998] provide computational evidence that this approach is computationally efficient by comparing it to the two other primal-dual methods.

Special cases of the extended version of step direction (4.6) were later rediscovered by Monteiro [1997]; Potra and Sheng [1998b]; Lin and Saigal [1995b] and others. The somewhat simpler analysis of Monteiro inspired Zhang [1998] to define the H_P operator with the corresponding step direction (4.10). The paper includes a proof of polynomiality for an infeasible start method. This proof was adapted here for the semidefinite version of the algorithm of Monteiro and Adler [1989a]. The H_P operator is further investigated in Monteiro and Zhang [1998]; Monteiro [1998]. Todd [1999] gives a survey on various search directions, including many more than cited here.

The concept of maximal complementarity was introduced by de Klerk, Roos, and Terlaky [1997] (and independently by Goldfarb and Scheinberg [1996]). They prove that, for $\mu \to 0$, the central path converges to a point satisfying this property (our discussion in Lemma 4.1.4, Lemma 4.1.6, and Lemma 4.1.7 follows their work). In the same paper they propose the skew-symmetric embedding for semidefinite programming and show Theorem 4.3.1 (this was independently also proposed in Potra and Sheng [1998a]; Nesterov, Todd, and Ye [1999]; Luo, Sturm, and Zhang [1996]). A more detailed analysis of the central path can be found in Goldfarb and Scheinberg [1996].

The predictor-corrector approach has already been in use in Helmberg [1994]; Helmberg, Rendl, Vanderbei, and Wolkowicz [1996]. Proofs of polynomial complexity for predictor-corrector algorithms are given in Kojima, Shida, and Shindoh [1998]; Lin and Saigal [1995b]; Lin and Saigal [1995a]; Potra and Sheng [1996].

Chapter 5

The Spectral Bundle Method

From a theoretical point of view, the interior point methods of Chapter 4 offer everything what one can hope for. They work on all reasonably defined semidefinite programs, they exploit second order information, and the order of the number of iterations is bounded by the fourth root of the number of variables. Still, for practical applications with many constraints the price to pay in a single iteration is often too high: The system matrix M (cf. (4.31)) that arises in computing the step direction is in general a dense positive definite matrix of order m, where m is the number of constraints. If the $\binom{m+1}{2}$ elements of M are stored explicitly and an explicit factorization is computed (e.g., a Cholesky factorization requires $m^3/3$ arithmetic operations), then within current technology the applicability of primal-dual interior point methods is limited to problems with about 7000 constraints on a well equipped work station. This motivates the search for other approaches, that are suitable for large m and allow to exploit problem structure. The spectral bundle method, which we are going to describe in this chapter, offers these features for semidefinite programs that can be cast as eigenvalue optimization problems, at the cost of a poor convergence rate: it is only a first order method.

The spectral bundle method is applicable to eigenvalue optimization problems of the form

$$\text{(E)} \qquad \min_{y \in \mathbb{R}^m} \lambda_{\max}(C - \mathcal{A}^T y) + b^T y.$$

In an extended variant, it allows for upper and lower bounds on y-variables. A large class of semidefinite programs and, in particular, several important relaxations of combinatorial optimization problems can equivalently be formulated as problems of this type, see, *e.g.*, the eigenvalue formulation of the max-cut relaxation (3.11) or the Lovász ϑ-function (3.24).

The maximum eigenvalue function $\lambda_{\max}(\cdot)$ is a nonsmooth convex function and we will employ nonsmooth convex optimization techniques to solve (E). This necessitates the use of some basic concepts and results from convex analysis; we have collected the relevant material in Appendix B. The spectral bundle method is a specialized subgradient method and we explain briefly and informally the main components of such algorithms.

In smooth nonlinear optimization the gradient plays a dominant role. The corresponding object in nonsmooth convex optimization is the subgradient of a convex function f. A subgradient at a point x and the function value $f(x)$ together describe a supporting hyperplane to the function in the point x; in other words, they give rise to a linear function that minorizes f and touches f in x. At a nonsmooth point of f several subgradients exist; the set of all subgradients at x forms the subdifferential $\partial f(x)$ of f at x. An oracle returning $f(x)$ and some subgradient $s \in \partial f(x)$ for given x is all the information needed by so-called subgradient methods for producing a minimizing sequence for f (under some mild regularity conditions).

Subgradient methods for general convex optimization such as the bundle methods of Kiwiel [1990] or Schramm and Zowe [1992] construct a polyhedral cutting plane model of the cost function by maintaining a bundle of accumulated subgradient information. The spectral bundle method improves on these methods by exploiting the semidefinite structure of the maximum eigenvalue function and forms a semidefinite cutting surface model of (E).

In the case of the maximum eigenvalue function, subgradients to $\lambda_{\max}(\cdot)$ in $X = C - \mathcal{A}^T y$ are determined by the eigenvectors to the maximum eigenvalue of X. Extremal eigenvalues and eigenvectors of large structured matrices may be determined by iterative methods. These generate a sequence of vectors converging to an eigenvector by a series of vector matrix multiplications; examples are the power method or the more sophisticated Lanczos method (see, *e.g.*, Saad [1992]; Parlett [1998]). If the structure of the matrices in $C - \mathcal{A}^T y$ allow the matrix vector multiplications to be carried out quickly, then the oracle will be able to deliver function value and subgradient quickly.

Typically, subgradient methods show fast initial convergence but a strong tailing off effect as the iterates approach the optimal solution. The same can be observed for the spectral bundle method. In several combinatorial applications the gap between the best integral solution and the optimal solution of a relaxation is fairly large. In this case it is sometimes more efficient to improve the relaxation than to compute its exact value. Fast initial convergence may be an advantage if the rough guess of the optimal solution yields sufficient information to tighten the relaxation, *e.g.*, by cutting planes.

The chapter is organized as follows. In Section 5.1 we show that eigenvalue optimization problems of the form (E) are equivalent to the duals of semidefinite programs with bounded feasible set; we also investigate the convex structure and subdifferentials of $\lambda_{\max}(X)$ and the cost function of (E). Section 5.2 starts with a description of the general approach of the proximal bundle method of Kiwiel [1990]. We then specialize this method to eigenvalue optimization problems of the form (E). Convergence of the method is proved in Section 5.3. Primal inequality constraints give rise to sign constraints on the dual y-variables. Section 5.4 explains how such bounds on y can be incorporated at negligible additional cost. In each iteration of the spectral bundle method a small quadratic semidefinite subproblem has to be solved. For this we employ an interior point method that we describe in Section 5.5. Due to the structure of the spectral bundle algorithm it is not always necessary to compute the maximum eigenvalue exactly. This inexact evaluation approach is theoretically save, as explained in Section 5.6, and can lead to considerable speedup in practice.

5.1 Eigenvalue Optimization

A symmetric matrix $X \in S_n$ is positive semidefinite if and only if the minimal eigenvalue of X is nonnegative, $\lambda_{\min}(X) \geq 0$. Due to this property, semidefinite programming and eigenvalue optimization over affine sets of matrices are tightly related. In this section we clarify this relation and state a few simple properties of the maximum eigenvalue function.

Consider the primal-dual pair of semidefinite programs

$$\begin{array}{llll} & \max & \langle C, X\rangle & \\ \text{(P)} & \text{s.t.} & \mathcal{A}X = b & \\ & & X \succeq 0, & \end{array} \qquad \begin{array}{llll} & \min & b^T y & \\ \text{(D)} & \text{s.t.} & Z = \mathcal{A}^T y - C & \\ & & Z \succeq 0 & \end{array}$$

and assume that the following holds:

$$\text{There exits a } \bar{y} \in \mathbb{R}^m \text{ with } I = \mathcal{A}^T \bar{y}. \tag{5.1}$$

This assumption allows to reformulate (D) as an eigenvalue optimization problem. In (D) we write the constraint $Z \succeq 0$ as $0 \geq -\lambda_{\min}(Z) = \lambda_{\max}(-Z)$ and lift it into the objective function by means of a Lagrange multiplier $a \geq 0$,

$$\min_y \; a\lambda_{\max}(C - \mathcal{A}^T y) + b^T y. \tag{5.2}$$

Proposition 5.1.1 *If $\mathcal{A}$ satisfies (5.1) then (D) is equivalent to (5.2) for $a = \max\{0, b^T\bar{y}\}$. Furthermore, if (P) is feasible then all its feasible solutions X satisfy $\operatorname{tr} X = a$, the primal optimum is attained and is equal to the infimum of (D).*

Proof. Let $\bar{y}$ satisfy $\mathcal{A}^T\bar{y} = I$. Then, for $y \in \mathbb{R}^m$, the half ray $\{y + \lambda\bar{y} : \lambda \geq \lambda_{\max}(C - \mathcal{A}^T y)\}$ is feasible for (D), because

$$\mathcal{A}^T(y + \lambda\bar{y}) - C = \lambda\mathcal{A}^T\bar{y} + \mathcal{A}^T y - C = \lambda I - (C - \mathcal{A}^T y) \succeq 0.$$

Now consider the cases $b^T\bar{y} < 0$ and $b^T\bar{y} \geq 0$.

If $b^T\bar{y} < 0$ then $a = 0$ and the objective values of (D) and (5.2) tend to minus infinity along any ray $y + \lambda\bar{y}$ with $\lambda \to \infty$.

For $0 \leq b^T\bar{y} = a$ problem (5.2) is constant along directions $\lambda\bar{y}$ for $\lambda \in \mathbb{R}$, because

$$\begin{aligned} &a\lambda_{\max}(C - \mathcal{A}^T(y + \lambda\bar{y})) + b^T(y + \lambda\bar{y}) = \\ &\quad = a\lambda_{\max}(C - \mathcal{A}^T y - \lambda I) + b^T y + \lambda a = a\lambda_{\max}(C - \mathcal{A}^T y) + b^T y. \end{aligned}$$

The choice of $\hat{\lambda} = \lambda_{\max}(C - \mathcal{A}^T y)$ ensures that $\lambda_{\max}(C - \mathcal{A}^T(y + \hat{\lambda}\bar{y})) = 0$. Therefore $y + \hat{\lambda}\bar{y}$ is a feasible solution of (D) with the same objective value. Conversely, for any feasible solution y of (D) we find that $y + \lambda_{\max}(C - \mathcal{A}^T y)\bar{y}$ is also a feasible solution of (D) with identical objective values in (D) and (5.2). This value is not greater than $b^T y$, since $\lambda_{\max}(C - \mathcal{A}^T y) = \lambda_{\max}(-Z) \leq 0$ for feasible y. This proves the equivalence of (D) and (5.2).

Since (D) has strictly feasible solutions, it follows from the Strong Duality Theorem 2.2.5 that the supremum in (P) is attained if (D) is finite. A feasible X of (P) satisfies $\mathcal{A}X - b = 0$ and thus

$$0 = \langle \mathcal{A}X - b, \bar{y}\rangle = \left\langle X, \mathcal{A}^T\bar{y}\right\rangle - b^T\bar{y} = \langle X, I\rangle - a = \operatorname{tr} X - a.$$

■

Observe that (5.1) holds for the dual of several of the semidefinite relaxations discussed in Chapter 3, in particular for max-cut and therefore also for quadratic 0-1 programming (with equality constraints) by means of the scaling of Lemma 3.2.3.

An appropriate dual satisfying (5.1) can, in theory, be set up for any primal problem (P) whose feasible set is bounded. As we will show below, a bounded primal feasible set can be scaled so that it has constant trace on the scaled primal feasible set. If $\operatorname{tr} X = a$ for all feasible solutions of (P) then adding the redundant constraint $\langle I, X\rangle = a$ will produce the desired effect. The latter addition may lead to changes in (D) only if I is not in the span of $\mathcal{A}^T$.

Proposition 5.1.2 *Let $\mathcal{X} = \{X \succeq 0 : \mathcal{A}X = b\}$ be a bounded set. Then there is a regular scaling matrix Q such that $\operatorname{tr} W = a$ for all $W \in \{W \succeq 0 : \mathcal{A}(QWQ^T) = b\} = \{Q^{-1}XQ^{-T} : X \in \mathcal{X}\}$.*

Proof. Let $\bar{X}$ be a point in the relative interior of $\mathcal{X}$, let $P_1 \in \mathbb{R}^k$ be an orthogonal matrix spanning the nonzero eigenspace of $\bar{X}$ and let the columns of $P_2 \in \mathbb{R}^{n-k}$ be an orthonormal basis of the null space of $\bar{X}$. By Lemma 2.2.9 the set $\{P_1 V P_1^T : V \succeq 0\}$ is the minimal face of S_n^+ containing $\mathcal{X}$. Consider the primal dual pair of semidefinite programs derived from the restriction of (P) to this minimal face,

$$\begin{array}{llll} & \max & \langle C, P_1 V P_1^T\rangle & \\ (\mathrm{P}_{P_1}) & \text{s.t.} & \mathcal{A}P_1 V P_1^T = b & \\ & & V \succeq 0. & \end{array} \qquad \begin{array}{llll} & \min & b^T y & \\ (\mathrm{D}_{P_1}) & \text{s.t.} & U = P_1^T(\mathcal{A}^T y - C)P_1 & \\ & & U \succeq 0. & \end{array}$$

Problem (P_{P_1}) is by construction strictly feasible (since there is a $V \succ 0$ with $\bar{X} = P_1 V P_1^T$) and has a bounded feasible set by assumption. By Theorem 2.2.5 the dual optimal solution is attained for all C. Choosing $C = P_1 P_1^T$ it follows that there is a $\bar{y}$ so that $P_1^T(\mathcal{A}^T\bar{y} - P_1 P_1^T)P_1 \succeq 0$ and therefore $P_1^T\mathcal{A}^T\bar{y}P_1 \succeq I$. Factorize $P_1^T\mathcal{A}^T\bar{y}P_1$ into BB^T with $B \succ 0$ and observe that for all $X \in \mathcal{X}$ (X has a representation $X = P_1 V P_1^T$ with $V \succeq 0$),

$$0 = \langle \bar{y}, \mathcal{A}X - b\rangle = \left\langle \mathcal{A}^T\bar{y}, X\right\rangle - b^T\bar{y} = \left\langle P_1^T\mathcal{A}^T\bar{y}P_1, V\right\rangle - b^T\bar{y} = \left\langle BB^T, V\right\rangle - b^T\bar{y}.$$

Therefore, the scaling $V = B^{-T}HB^{-1}$ transforms (P_{P_1}) into a problem in $H \succeq 0$ satisfying $\operatorname{tr} H = b^T\bar{y}$ for all feasible H. The direct projection from $X \in \mathcal{X}$ to H is obtained from $X = P_1 B^{-T} H B^{-1} P_1^T$ and can be extended to a scaling of the positive semidefinite cone $X = QWQ^T \succeq 0$ by the matrix $Q = \left[P_1 B^{-T}, P_2\right]$. ■

Corollary 5.1.3 *Any problem (P) with bounded feasible set admits a dual that guarantees strong duality and satisfies* (5.1).

The construction outlined in the proof of Proposition 5.1.2 is only of theoretic interest. The matrices $\bar{X}$ and $\bar{y}$ of the proof are not readily available in general and scaling typically destroys the structure of the problem.

If a reasonable bound $M > 0$ on the trace of the primal feasible set is known then the following approach may be computationally feasible. Introduce to (P) a slack variable s together with the constraint

$$\left\langle \begin{bmatrix} X & 0 \\ 0 & s \end{bmatrix}, I \right\rangle = M.$$

This ensures (5.1). Since Z will have the same block structure, the additional complexity is neglectable. However, a large M may cause numerical difficulties in computing the optimum.

For simplicity we will assume in the following that $a = 1$ is the correct multiplier and consider the function

$$f(y) = \lambda_{\max}(C - \mathcal{A}^T y) + b^T y. \tag{5.3}$$

We begin our investigations with the maximum eigenvalue function $\lambda_{\max}(\cdot)$.

The most common variational characterization of the maximum eigenvalue of a symmetric matrix uses the Rayleigh-Ritz ratio (see Theorem A.0.4), $\lambda_{\max}(X) = \max_{\|v\|=1} v^T X v$. The maximum is attained for eigenvectors to the maximum eigenvalue of X. Since $v^T X v = \langle X, vv^T \rangle$ and the set

$$\mathcal{W} = \{W \succeq 0 : \operatorname{tr} W = 1\} \tag{5.4}$$

is the convex hull of the set $\{vv^T : \|v\| = 1\}$, the maximum eigenvalue function may equivalently be formulated as a semidefinite program,

$$\lambda_{\max}(X) = \max\{\langle X, W\rangle : W \in \mathcal{W}\}. \tag{5.5}$$

This characterization of $\lambda_{\max}(\cdot)$ as the maximum over a family of linear functions implies that $\lambda_{\max}(\cdot)$ is convex. Since $\mathcal{W}$ is bounded, $\lambda_{\max}(\cdot)$ is also Lipschitz continuous (in fact, it is the support function of $\mathcal{W}$ and therefore sublinear). The subgradients of $\lambda_{\max}(\cdot)$ at X, *i.e.*, the linear forms W satisfying the subgradient inequality $\lambda_{\max}(Y) \geq \lambda_{\max}(X) + \langle W, Y - X\rangle$ for all $Y \in S_n$, are the matrices

$$\begin{array}{rcl} W \in \operatorname{Argmax}\{\langle X, W\rangle : W \in \mathcal{W}\} & = & \{W \in \mathcal{W} : \langle X, W\rangle = \lambda_{\max}(X)\} \\ & = & \left\{PVP^T : \operatorname{tr} V = 1, V \succeq 0\right\}, \end{array} \tag{5.6}$$

where the columns of P form an orthonormal basis of the eigenspace to the maximum eigenvalue of X. We may also view the set of maximizers as the convex hull of the dyadic products of the normalized eigenvectors to the maximum eigenvalue of X. Thus, any eigenvector to the maximum eigenvalue of X gives rise to a subgradient of $\lambda_{\max}$ at X. The subdifferential of $\lambda_{\max}$ at X is the set of subgradients at X. It is denoted by $\partial\lambda_{\max}(X)$ and is precisely the set described in (5.6).

Characterization (5.5) allows to reformulate f of (5.3) as

$$f(y) = \max_{W \in \mathcal{W}} \left\langle C - \mathcal{A}^T y, W \right\rangle + b^T y = \max_{W \in \mathcal{W}} \langle C, W\rangle + \langle b - \mathcal{A}W, y\rangle. \tag{5.7}$$

Since f is the max over the linear functions

$$f_W(y) := \langle C, W\rangle + \langle b - \mathcal{A}W, y\rangle \tag{5.8}$$

the subdifferential of f at y is the set

$$\begin{array}{rcl} \partial f(y) & = & \left\{b - \mathcal{A}W : W \in \mathcal{W}, \left\langle C - \mathcal{A}^T y, W \right\rangle = \lambda_{\max}(C - \mathcal{A}^T y)\right\} \\ & = & \left\{\nabla f_W : W \in \partial\lambda_{\max}(C - \mathcal{A}^T y)\right\}. \end{array} \tag{5.9}$$

Observe that the set of all subgradients over all y is bounded because $\mathcal{W}$ is so. A point y_* is a minimizer of the convex function f if and only if $0 \in \partial f(y_*)$, or equivalently,

$$y_* \in \operatorname{Argmin} f \Longleftrightarrow \exists W_* \in \mathcal{W} : \mathcal{A}W_* = b \text{ and } \left\langle C - \mathcal{A}^T y_*, W_* \right\rangle = \lambda_{\max}(C - \mathcal{A}^T y_*). \tag{5.10}$$

Replacing in (5.7) the set $\mathcal{W}$ by a subset $\widehat{\mathcal{W}} \subseteq \mathcal{W}$ gives rise to a function minorizing f,

$$f_{\widehat{\mathcal{W}}}(y) := \max_{W \in \widehat{\mathcal{W}}} f_W(y) \leq f_{\mathcal{W}}(y) = f(y) \quad \text{for all } \widehat{\mathcal{W}} \subseteq \mathcal{W}, y \in \mathbb{R}^m. \tag{5.11}$$

5.2 The Algorithm

The spectral bundle method is a specialization of the proximal bundle method of Kiwiel [1990] to eigenvalue optimization problems of the form (E). We start with an intuitive explanation of the general principle of this method.

The Proximal Bundle Method

The basic assumption in subgradient methods is that a convex objective function f is given by a first order oracle, *i.e.*, a subroutine that computes for an input point $\bar{y}$ the objective value $f(\bar{y})$ and a subgradient $\bar{g} \in \partial f(\bar{y})$. By definition, subgradients satisfy the subgradient inequality

$$f(y) \geq f(\bar{y}) + \langle \bar{g}, y - \bar{y} \rangle \qquad \forall y \in \mathbb{R}^m.$$

They may therefore be used to construct a cutting plane model $\hat{f}$ of f minorizing f on $\mathbb{R}^m$. For example, let $y^1, \ldots, y^k$ denote a set of points for which the oracle has been evaluated to $f(y^i)$ and g^i for $i = 1, \ldots, k$, then a possible cutting plane model is

$$\hat{f}^k(y) = \max_{i=1,\ldots,k} f(y^i) + \left\langle g^i, y - y^i \right\rangle.$$

The linear approximations $f(y^i) + \left\langle g^i, y - y^i \right\rangle$ of f will be of reasonable quality only in the vicinity of y^i. Therefore we concentrate on a neighborhood around the last successful iterate $\hat{y}^k$ and determine the next trial point y^{k+1} as the minimizer of

$$f^k(y) = \hat{f}^k(y) + \frac{u}{2} \left\| y - \hat{y}^k \right\|^2.$$

The quadratic term $\left\| y - \hat{y}^k \right\|^2$ ensures that the minimum of this *augmented model* is finite and unique. The *weight* $u > 0$ allows to control to some extent the distance of y^{k+1} to the so called *stability center* $\hat{y}^k$ (u may also be interpreted as a Lagrange multiplier to a trust region constraint $\left\| y - \hat{y}^k \right\|^2 \leq R$).

If the function value $f(y^{k+1})$ at the new candidate shows reasonable progress in comparison to the decrease predicted by the model value $\hat{f}^k(y^{k+1})$, *i.e.*, for some parameter $\kappa \in (0, 1)$

$$f(\hat{y}^k) - f(y^{k+1}) \geq \kappa \left[f(\hat{y}^k) - \hat{f}^k(y^{k+1}) \right],$$

then we make a *descent step* by moving the stability center to $\hat{y}^{k+1} = y^{k+1}$. Otherwise we improve the cutting plane model by adding the new cutting plane $f(y^{k+1}) + \left\langle g^{k+1}, \cdot - y^{k+1} \right\rangle$ but keep the old stability center, $\hat{y}^{k+1} = \hat{y}^k$. This is called a *null step*. These steps are iterated till the decrease predicted by the cutting plane model is small relative to the function value.

In the following we specialize this general approach to problems of the form (E).

The Oracle

The oracle is assumed to deliver, for a given matrix $C - \mathcal{A}^T y^k$, the maximum eigenvalue and a matrix

$$W_S^k \in \operatorname*{Argmax}_{W \in \mathcal{W}} \langle C - \mathcal{A}^T y^k, W \rangle .$$

In practice W_S^k is typically computed by iterative methods (see Section 5.6) that deliver a normalized eigenvector v^k to the maximum eigenvalue. We will use the slightly more general W_S^k for the formal statement of the algorithm but assume W_S^k to be the dyadic product of an eigenvector, $W_S^k = v^k (v^k)^T$, in some updating formulas.

The Cutting Plane Model

The decisive step in exploiting the structure of (E) is the choice of an appropriate cutting plane model for f. In view of (5.11), a cutting plane model may be obtained by restricting $\mathcal{W}$ to some smaller set. Our particular choice for iteration k is

$$\widehat{\mathcal{W}}^k = \left\{ P_k V P_k^T + \alpha \overline{W}_k : \operatorname{tr} V + \alpha = 1, V \in S^+_{r_k}, \alpha \geq 0 \right\}, \tag{5.12}$$

where $P_k \in M_{n,r_k}$ is an orthonormal matrix and $\overline{W}_k \in \mathcal{W}$. The corresponding cutting plane (or rather cutting surface) model is $f_{\widehat{\mathcal{W}}^k}$ (cf. (5.11)). The character of this model is clarified in the following proposition.

Proposition 5.2.1 *For $\widehat{\mathcal{W}}^k$ of (5.12) and $f_{\widehat{\mathcal{W}}^k}$ defined as in (5.11)*

$$f_{\widehat{\mathcal{W}}^k}(y) = \max \left\{ \lambda_{\max}(P_k^T (C - \mathcal{A}^T y) P_k), \left\langle C - \mathcal{A}^T y, \overline{W}_k \right\rangle \right\} + b^T y \leq f(y). \tag{5.13}$$

Proof. The right hand side inequality is the same as in (5.11). To prove the left hand equation we exploit that the maximum of a linear function over a closed bounded convex set is equal to the maximum over the extreme points of the set. Determining $f_{\widehat{\mathcal{W}}^k}(y)$ for fixed y is a linear program over the semidefinite set $\widehat{\mathcal{W}}^k$ (cf. (5.11) and (5.8)). Since $\widehat{\mathcal{W}}^k$ is the convex hull of $\overline{W}_k$ and the set $\left\{ P_k V P_k^T : \operatorname{tr} V = 1, V \succeq 0 \right\}$ we may consider these two sets separately and take the maximum of the results. Now, $f_{\overline{W}_k}(y) = \left\langle C - \mathcal{A}^T y, \overline{W}_k \right\rangle + b^T y$ and

$$\begin{aligned} \max_{W \in \left\{ P_k V P_k^T : \operatorname{tr} V = 1, V \succeq 0 \right\}} f_W(y) &= \max_{\{V \succeq 0 : \operatorname{tr} V = 1\}} \left\langle C - \mathcal{A}^T y, P_k V P_k^T \right\rangle + b^T y \\ &= \max_{\{V \succeq 0 : \operatorname{tr} V = 1\}} \left\langle P_k^T (C - \mathcal{A}^T y) P_k, V \right\rangle + b^T y \\ &= \lambda_{\max}(P_k^T (C - \mathcal{A}^T y) P_k) + b^T y \end{aligned}$$

where the last equation follows from (5.5) for the set $\mathcal{W}' = \{V \succeq 0 : \operatorname{tr} V = 1\}$. ∎

The proposition shows that, for small r_k, the value of the cutting plane model can be determined efficiently. It also hints at useful choices for P_k: In order to obtain a large value of $f_{\widehat{\mathcal{W}}^k}$ in the vicinity of the current candidate y^k, the matrix P_k should span the eigenspaces of the largest eigenvalues of $C - \mathcal{A}^T y^k$.

Without $\overline{W}_k$ the set $\widehat{\mathcal{W}}^k$ corresponds to a $\binom{r_k+1}{2}$ dimensional face of the semidefinite cone (Theorem 1.2.9), which might be too small to contain any W_* of (5.10). The matrix $\overline{W}_k$ allows $\widehat{\mathcal{W}}^k$ to reach into the interior of S_n^+ without significantly increasing the cost of computing the next trial point.

The Augmented Model

It will be convenient to express the augmented model in terms of the augmented Lagrangian

$$L^k(y, W) := f_W(y) + \frac{u}{2} \left\| y - \hat{y}^k \right\|^2 = \left\langle C - \mathcal{A}^T y, W \right\rangle + b^T y + \frac{u}{2} \left\| y - \hat{y}^k \right\|^2, \tag{5.14}$$

then

$$f^k(y) := \max_{W \in \widehat{\mathcal{W}}^k} L^k(y, W) = f_{\widehat{\mathcal{W}}^k}(y) + \frac{u}{2} \left\| y - \hat{y}^k \right\|^2. \tag{5.15}$$

In the dual problem to $\min_y f^k(y)$,

$$\max_{W \in \widehat{\mathcal{W}}^k} \min_y L^k(y, W), \tag{5.16}$$

the inner minimization over y is unconstrained. For fixed $W \in \widehat{\mathcal{W}}^k$ the corresponding optimal y can be determined explicitly,

$$y^k_{\min}(W) := \hat{y}^k + \frac{1}{u}(\mathcal{A}W - b) = \hat{y}^k - \frac{1}{u}\nabla f_W. \tag{5.17}$$

General theorems about convex duality establish easily that strong duality holds for these duals, but it is instructive to prove this fact directly via semidefinite duality.

Lemma 5.2.2 *Let L^k be as defined in* (5.14). *Then*

$$\min_y \max_{W \in \widehat{\mathcal{W}}^k} L^k(y, W) = L^k(y^{k+1}, W^{k+1}) = \max_{W \in \widehat{\mathcal{W}}^k} \min_y L^k(y, W) \tag{5.18}$$

with $y^{k+1} = y^k_{\min}(W^{k+1})$ unique, and W^{k+1} an optimal solution of

$$\text{(QSP)} \qquad \begin{array}{ll} \min & \frac{1}{2u}\|b - \mathcal{A}W\|^2 - \langle W, C - \mathcal{A}^T\hat{y}^k\rangle - b^T\hat{y}^k \\ \text{s.t.} & W = P_k V P_k^T + \alpha \overline{W}_k \\ & \operatorname{tr} V + \alpha = 1 \\ & V \succeq 0, \alpha \geq 0. \end{array}$$

Proof. By Proposition 5.2.1, $f_{\widehat{\mathcal{W}}^k}$ may be rewritten as

$$f_{\widehat{\mathcal{W}}^k}(y) = \min\ \lambda + b^T y \ \text{ s.t. } \ \lambda I \succeq P_k^T(C - \mathcal{A}^T y)P_k,\ \lambda \geq \langle C - \mathcal{A}^T y, \overline{W}_k \rangle.$$

In order to model $\frac{u}{2}\|y - \hat{y}\|^2 = \frac{u}{2}(y^T y - 2\langle y, \hat{y}\rangle + \hat{y}^T\hat{y})$ we introduce a scalar d for the quadratic term and use (2.7),

$$d \geq \frac{u}{2} y^T y \quad \Longleftrightarrow \quad \begin{bmatrix} d & y^T \\ y & \frac{2}{u} I \end{bmatrix} \succeq 0.$$

This allows to reformulate $\min_y f^k(y)$ as the dual semidefinite program

$$\begin{array}{ll} \min & \lambda + \langle b - u\hat{y}, y\rangle + d + \frac{u}{2}\hat{y}^T\hat{y} \\ \text{s.t.} & U = \lambda I + P_k^T(\mathcal{A}^T y - C)P_k \succeq 0 \\ & \beta = \lambda + \langle \mathcal{A}^T y - C, \overline{W}_k\rangle \geq 0 \\ & G = \begin{bmatrix} d & y^T \\ y & I \end{bmatrix} \succeq 0. \end{array}$$

Denote by V, α, and H the dual variables to U, β, and G, respectively, to obtain the primal problem

$$\begin{array}{ll} \max & \langle C, P_k V P_k^T\rangle + \langle C, \overline{W}_k\rangle \alpha - \frac{2}{u}\sum_{i=2}^{n+1} h_{ii} + \frac{u}{2}\hat{y}^T\hat{y} \\ \text{s.t.} & \langle I, V\rangle + \alpha = \operatorname{tr} V + \alpha = 1 \\ & h_{11} = 1 \\ & h_{1i} = \frac{1}{2}[b - u\hat{y} - \mathcal{A}(P_k V P_k^T) - \alpha\mathcal{A}\overline{W}_k]_i \qquad \text{for } i = 2, \ldots, n+1 \\ & V \succeq 0, \alpha \geq 0, H \succeq 0. \end{array}$$

Both primal and dual problems are strictly feasible, so strong duality holds by Theorem 2.2.5 and the optimal value is attained on both sides. The 2×2 submatrices $\begin{bmatrix} h_{11} & h_{1i} \\ h_{1i} & h_{ii} \end{bmatrix}$ of H must be

positive semidefinite for $i = 2, \ldots, n+1$ and therefore $h_{ii} \geq h_{1i}^2$. Because $-h_{ii}$ appears in the cost function, and because the lower bound h_{1i}^2 is attained on all h_{ii} simultaneously for $H = hh^T \succeq 0$ with $h = [1, h_{12}, \ldots, h_{1n}]^T$, it follows that $h_{ii} = h_{1i}^2$ in all optimal solutions. Therefore H can be eliminated from the primal problem and the primal cost function simplifies to

$$\begin{aligned}\left\langle C, P_k V P_k^T + \alpha \overline{W}_k \right\rangle - \tfrac{1}{2u} \left\| b - u\hat{y} - \mathcal{A}(P_k V P_k^T + \alpha \overline{W}_k) \right\|^2 + \tfrac{u}{2} \hat{y}^T \hat{y} = \\ = -\tfrac{1}{2u} \left\| b - \mathcal{A}(W) \right\|^2 - \left\langle b - \mathcal{A}(W), \hat{y} \right\rangle + \left\langle C, W \right\rangle ,\end{aligned}$$

with $W = P_k V P_k^T + \alpha \overline{W}_k$. The primal problem is now in the form that results from substituting the expression for $y_{\min}^k(W)$ of (5.17) into (5.16). It remains to switch the optimization direction and the sign in the cost function of the primal problem to obtain (QSP). Uniqueness of y^{k+1} follows from the strict convexity of (5.15). ■

Observe that (QSP) is a quadratic semidefinite programming problem in small dimension if r_k of (5.12) is small. It can be solved efficiently via interior point methods (see Section 5.5) and the optimal solution W^{k+1} gives rise to the new candidate point $y^{k+1} = y_{\min}^k(W^{k+1})$.

Updating the Model

We add new subgradient information $W_S = vv^T \in \mathcal{W}$ to the model by adding the new eigenvector v as orthonormalized column to P, thereby increasing the number of columns r. At the same time it is important to keep r bounded so that (QSP) is still efficiently solvable. If r grows too large, part of the subspace spanned by P has to be eliminated from P and its contribution to the solution has to be incorporated in $\overline{W}$.

Before we proceed to this aggregation step, we have to localize the "important" information in $\widehat{\mathcal{W}}^k$ that has to be saved to $\widehat{\mathcal{W}}^{k+1}$. The decisive piece of information is W^{k+1} and the cutting plane $f_{W^{k+1}}$ induced by it (cf. (5.8)). Indeed, in Lemma 5.2.2, (5.18) implies that (y^{k+1}, W^{k+1}) is a saddle point of L^k,

$$\max_{W \in \widehat{\mathcal{W}}^k} L^k(y^{k+1}, W) \leq L^k(y^{k+1}, W^{k+1}) \leq \min_y L^k(y, W^{k+1}). \tag{5.19}$$

Thus, if $W^{k+1} \in \widehat{\mathcal{W}}^{k+1}$ then the right hand side inequality ensures that after a null step the value of the augmented value cannot decrease. This is true, because in a null step the stability center $\hat{y}^{k+1} = \hat{y}^k$ remains unchanged, and therefore also L^{k+1},

$$L^k(y, W^{k+1}) = L^{k+1}(y, W^{k+1}) \leq f^{k+1}(y) \quad \text{for all } y \in \mathbb{R}^m \text{ if } \hat{y}^k = \hat{y}^{k+1}. \tag{5.20}$$

We will see in the next section that

$$W^{k+1}, W_S^{k+1} \in \widehat{\mathcal{W}}^{k+1} \text{ with } \widehat{\mathcal{W}}^{k+1} \text{ of the form (5.12)} \tag{5.21}$$

is all that is needed to guarantee convergence.

The minimal choice satisfying (5.21) for $W_S^{k+1} = v^{k+1}(v^{k+1})^T$ is $\overline{W}_{k+1} = W^{k+1}$ and $P_{k+1} = v^{k+1}$, so $r_{k+1} = 1$ would suffice in theory. In practice we would like to preserve the "most important" subspace within the subspace spanned by the columns of P_k in order to accumulate the relevant eigenspace without recomputing the entire spectrum in each evaluation. It is not possible to predict which subspace will be especially helpful in the future, but W^{k+1} provides some indication on the subspace that has been important in the last computation. Let V_* and α_* denote the optimal solutions to (QSP) that give rise to W^{k+1}, and let $Q\Lambda Q^T = V_*$ denote an eigenvalue decomposition of V_* with $Q^T Q = I$ and $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_{r_k})$, $\lambda_1 \geq \ldots \geq \lambda_{r_k} \geq 0$. Then

$$W^{k+1} = P_k V_* P_k^T + \alpha_* \overline{W}_k = (P_k Q) \Lambda (P_k Q)^T + \alpha_* \overline{W}_k. \tag{5.22}$$

We see that the first columns of $P_k Q$, corresponding to the large eigenvalues of Λ, carry more information about W^{k+1} then the last columns. Taking this as an indication that the first columns

of P_kQ are more important for the optimization process, we split Q into two parts, $Q = [Q_1, Q_2]$, Q_1 carrying the eigenvectors to the large eigenvalues of V_*. Analogously, we split Λ into two smaller diagonal matrices Λ_1 and Λ_2. The new matrix P_{k+1} will hold an orthonormal basis of the space spanned by the columns of P_kQ_1 and v^{k+1},

$$P_{k+1} = \operatorname{orth}\left[P_kQ_1, v^{k+1}\right]. \tag{5.23}$$

The orthonormalization operator orth could be implemented by a QR-factorization of the matrix $\left[P_kQ_1, v^{k+1}\right]$. The remaining columns P_kQ_2 are easily added to the new aggregate matrix,

$$\overline{W}_{k+1} = \frac{(P_kQ_2)\Lambda_2(P_kQ_2)^T + \alpha_*\overline{W}_k}{\operatorname{tr}\Lambda_2 + \alpha_*}. \tag{5.24}$$

Proposition 5.2.3 *For $W_S^{k+1} = v^{k+1}(v^{k+1})^T \in \mathcal{W}$, update formulas (5.23) and (5.24) ensure that P_{k+1} is orthonormal, $\overline{W}_{k+1} \in \mathcal{W}$, and that (5.21) is satisfied for $\widehat{\mathcal{W}}^{k+1}$ of (5.12).*

Proof. The first two properties follow from the construction, we show (5.21). Since the columns of P_{k+1} form an orthonormal basis of the space spanned by P_kQ_1 and v^{k+1}, there is a normalized vector q such that $v^{k+1} = P_{k+1}q$. Then $v^{k+1}(v^{k+1})^T = P_{k+1}qq^TP_{k+1}^T \in \widehat{\mathcal{W}}^{k+1}$. Now let $V_* = Q_1\Lambda_1Q_1^T + Q_2\Lambda_2Q_2$ and α_* be as defined in the construction of the update formulas. Again, there is an orthonormal matrix $\bar{Q}$ with $P_kQ_1 = P_{k+1}\bar{Q}$. Define the matrix $\bar{V} = \bar{Q}\Lambda_1\bar{Q}^T \succeq 0$ and the scalar $\bar{\alpha} = \operatorname{tr}\Lambda_2 + \alpha_* \geq 0$. We have $\operatorname{tr}\bar{V} + \bar{\alpha} = \operatorname{tr}\Lambda_1 + \operatorname{tr}\Lambda_2 + \alpha_* = 1$ by the feasibility of (V_*, α_*) for (QSP) and $\widehat{\mathcal{W}}^{k+1} \ni P_{k+1}\bar{V}P_{k+1}^T + \bar{\alpha}\overline{W}_{k+1} = (P_kQ_1)\Lambda_1(P_kQ_1)^T + (P_kQ_2)\Lambda_2(P_kQ_2)^T + \alpha_*\overline{W}_k = P_kV_*P_k^T + \alpha_*\overline{W}_k = W^{k+1}$ by (5.24) and (5.22). ■

The Stopping Criterion

We would like the algorithm to stop if the objective value is relatively close to the optimal value $\min_y f(y)$. Unfortunately, no lower bound is available for this value. The cutting plane model $f_{\widehat{\mathcal{W}}^k}$ minorizes f, but we do not know its minimizer. In the augmented model the quadratic term mimics a trust region constraint for the cutting plane model. Therefore we we may view y^{k+1} as the minimizer of $f_{\widehat{\mathcal{W}}^k}$ over a ball. If the weight u is reasonably small, then $f_{\widehat{\mathcal{W}}^k}(y^{k+1}) = f_{W^{k+1}}(y^{k+1})$ (cf. (5.19)) gives a lower bound on f over a ball of reasonable size. If the gap between $f(\hat{y}^k)$ and $f_{W^{k+1}}(y^{k+1})$ is small,

$$f(\hat{y}^k) - f_{W^{k+1}}(y^{k+1}) < \varepsilon(|f(\hat{y}^k)| + 1), \tag{5.25}$$

then we cannot expect good progress of the algorithm within the trust region and terminate.

We are now ready to state the algorithm.

Algorithm 5.2.4 *(Spectral Bundle Method)*

Input: $y^0 \in \mathbb{R}^m$, $\varepsilon \geq 0$, $\kappa \in (0,1)$, *a weight* $u > 0$.

1. *Set $k = 0$, $\hat{y}^0 = y^0$, compute $f(y^0)$ and $\widehat{\mathcal{W}}^0$.*
2. *(Trial point finding). Compute W^{k+1} and $y^{k+1} = y_{\min}^k(W^{k+1})$ (Lemma 5.2.2).*
3. *(Stopping criterion). If $f(\hat{y}^k) - f_{W^{k+1}}(y^{k+1}) \leq \varepsilon(|f(\hat{y}^k)| + 1)$ then* **stop**.
4. *(Evaluation). Find $W_S^{k+1} \in \operatorname{Argmax}_{W\in\mathcal{W}}\langle C - \mathcal{A}^Ty^{k+1}, W\rangle$ and determine $f(y^{k+1})$.*
5. *(Descent test). If $f(\hat{y}^k) - f(y^{k+1}) \geq \kappa[f(\hat{y}^k) - f_{W^{k+1}}(y^{k+1})]$ then set $\hat{y}^{k+1} = y^{k+1}$ (descent step); otherwise set $\hat{y}^{k+1} = \hat{y}^k$ (null step).*
6. *(Model updating). Choose a $\widehat{\mathcal{W}}^{k+1} \supset \{W^{k+1}, W_S^{k+1}\}$ of the form (5.12).*
7. *Increase k by one and* **goto** *2.*

Remark 5.2.5

1. *In practice the weight u is adapted during the algorithm for efficiency reasons. For example, a sequence of descent steps would indicate that u should be decreased in order to allow for larger steps. Similarly, if each descent step is preceded by a long sequence of null steps then a larger u might be more efficient. Such updates do not endanger convergence if u is nondecreasing during null steps and nonincreasing as well as bounded from below by some $\underline{u} > 0$ for descent steps (see Kiwiel [1990]; this paper also includes the description of a reasonable update rule).*

2. *The algorithmic approach may be interpreted in terms of the dual problem (D) of page 72. Optimizing in (QSP) over a cutting plane model defined by $W = PVP^T$, $V \succeq 0$ with $P^TP = I$ corresponds to relaxing in (D) the constraint $Z \succeq 0$ to $P^TZP \succeq 0$ while the quadratic term ensures that the solutions remain bounded. The solution Z_P of this modified problem will usually have some negative eigenvalues. The corresponding eigenvectors are used to update P. Thus the spectral bundle method may be viewed as a mechanism for updating this relaxation till Z remains positive semidefinite and we may move on to the better y that is now feasible. Within this framework, assumption (5.1) makes it possible to push the solution of the relaxation back into the feasible set of (D).*

5.3 Convergence Analysis

We will prove convergence of Algorithm 5.2.4 for $\varepsilon = 0$. The proof is divided into the following steps. After a technical lemma we prove optimality of the final y^k if the algorithm stops after a finite number of iterations. Then we show that for constant stability center $\hat{y}$ (if the descent test is omitted) the algorithm converges to the optimal solution of $\min_y f(y) + \frac{u}{2}\|y - \hat{y}\|^2$ and that the gap between model value $f_{W^k}(y^k)$ and function value $f_{W_S^k}(y^k)$ converges to zero. In the next step this will imply that either a descent step is triggered after a finite number of iterations or $\hat{y}^k$ is optimal. Finally, we prove that in the case of an infinite number of descent steps the sequence of the $f(\hat{y}^k)$ satisfies $f(\hat{y}^k) \downarrow \inf_y f(y)$.

The following two relations state, for L^k, the simple facts that the linear part of a quadratic can be removed by shifting the origin to the minimizer of the quadratic and that the minimum of a function is smaller than an upper bound on one of its function values.

Proposition 5.3.1

$$L^k(y, W) = L^k(y^k_{\min}(W), W) + \frac{u}{2}\left\|y - y^k_{\min}(W)\right\|^2, \tag{5.26}$$

$$L^k(y^{k+1}, W^{k+1}) \leq f(\hat{y}^k). \tag{5.27}$$

Proof. Using (5.14) and $y^k_{\min}(W) - \hat{y}^k = \frac{1}{u}(\mathcal{A}W - b)$ from (5.17) we obtain

$$\begin{aligned} L^k(y, W) &= \left\langle C - \mathcal{A}^T y, W\right\rangle + b^T y + \frac{u}{2}\left\|y - y^k_{\min}(W) + y^k_{\min}(W) - \hat{y}^k\right\|^2 \\ &= \langle C, W\rangle + \langle b - \mathcal{A}(W), y\rangle \\ &\quad + \frac{u}{2}\left\|y - y^k_{\min}(W)\right\|^2 - \left\langle b - \mathcal{A}W, y - y^k_{\min}(W)\right\rangle + \frac{u}{2}\left\|y^k_{\min}(W) - \hat{y}^k\right\|^2 \\ &= \langle C, W\rangle + \left\langle b - \mathcal{A}(W), y^k_{\min}(W)\right\rangle + \frac{u}{2}\left\|y^k_{\min}(W) - \hat{y}^k\right\|^2 + \frac{u}{2}\left\|y - y^k_{\min}(W)\right\|^2. \end{aligned}$$

The last line is the right hand side of (5.26). To see (5.27) we need Lemma 5.2.2 and (5.11),

$$L^k(y^{k+1}, W^{k+1}) = \min_y L^k(y, W^{k+1}) \leq L^k(\hat{y}^k, W^{k+1}) = f_{W^{k+1}}(\hat{y}^k) \leq f(\hat{y}^{k+1}).$$

■

Now we show that for $\varepsilon = 0$ the stopping criterion identifies optimal solutions correctly.

Lemma 5.3.2 *$f(\hat{y}^k) \geq f_{W^{k+1}}(y^{k+1})$ and if $f(\hat{y}^k) = f_{W^{k+1}}(y^{k+1})$ then $\hat{y}^k$ is optimal.*

Proof. Since $W^{k+1} \in \widehat{\mathcal{W}}^k$ by step 6 of Algorithm 5.2.4, we have $f_{W^{k+1}}(y^{k+1}) \leq f_{\widehat{\mathcal{W}}^k}(y^{k+1}) \leq f_{\widehat{\mathcal{W}}^k}(y^{k+1}) + \frac{u}{2}\|y^{k+1} - \hat{y}^k\|^2 = L^k(y^{k+1}, W^{k+1}) \leq f(\hat{y}^k)$ by (5.27). If equality holds then the same relations imply $\|y^{k+1} - \hat{y}^k\|^2 = 0$. Since $y^{k+1} = y^k_{\min}(W^{k+1})$, (5.17) and (5.9) yield $0 = u(y^{k+1} - \hat{y}^k) = -\nabla f_{W^{k+1}} \in \partial f(y^k)$. ■

This handles the case of a finite number of iterations.

Next, we analyze the asymptotic behavior of null steps. The null step mechanism ensures with its update strategy that the iterates converge to the minimizer of $f(\cdot) + \frac{u}{2}\|\cdot - \hat{y}^k\|^2$. The proof hinges on three basic observations. First, the minimal value of the augmented model is bounded from above by $f(\hat{y}^k)$ via (5.27). Second, the element $W^{k+1} \in \widehat{\mathcal{W}}^{k+1}$ forces the augmented model to increase whenever it moves away from y^{k+1} by (5.26). Finally, the new subgradient $W_S^{k+1} \in \widehat{\mathcal{W}}^{k+1}$ increases the value of the augmented model in y^{k+1} itself. In combination with the boundedness of the iterates and the subgradients this will force $f_{W^{k+1}}(y^{k+1})$ and $f_{W_S^{k+1}}(y^{k+1})$ together, *i.e.*, the model converges to the true function at the point of interest.

Lemma 5.3.3 *Assume that, starting with iteration K, the algorithm is run without the descent test, $\hat{y} = \hat{y}^K = \hat{y}^{K+1} = \ldots$. Then*

$$\varepsilon_k := f_{W_S^k}(y^k) - f_{W^k}(y^k) \to 0 \tag{5.28}$$

$$y^k \to \operatorname{argmin}_y \max_{W \in \mathcal{W}} L^k(y, W) = \operatorname{argmin}_y f(y) + \tfrac{u}{2}\|y - \hat{y}\|^2 \tag{5.29}$$

Proof. Note that the functions $L^K = L^{K+1} = \ldots$ (5.14) and $y^K_{\min} = y^{K+1}_{\min} = \ldots$ (5.17) are identical for all $k \geq K$. The $y^k = y^K_{\min}(W^k)$ remain bounded for $k \geq K$ because $\max_{W \in \mathcal{W}} \|y^K_{\min}(W)\|$ is bounded by the compactness of $\mathcal{W}$ (see (5.4)). Since $W^k \in \widehat{\mathcal{W}}^k$ by (5.21) and $W^{k+1} = \operatorname{argmax}_{W \in \widehat{\mathcal{W}}^k} L^k(y^{k+1}, W)$ by (5.19), we obtain from (5.26) and (5.27) for all $k \geq K$

$$L^K(y^k, W^k) + \frac{u}{2}\left\|y^{k+1} - y^k\right\|^2 = L^K(y^{k+1}, W^k) \leq L^K(y^{k+1}, W^{k+1}) \leq f(\hat{y}). \tag{5.30}$$

So there exists an $f_* \in \mathbb{R}$ with $L^K(y^k, W^k) \uparrow f_* \leq f(\hat{y})$. In addition $\left\|y^{k+1} - y^k\right\|^2 \to 0$. By (5.21) $W_S^k \in \widehat{\mathcal{W}}^k$ and therefore (5.19) yields

$$f_{W_S^k}(y^{k+1}) \leq \max_{W \in \widehat{\mathcal{W}}^k} f_W(y^{k+1}) = f_{W^{k+1}}(y^{k+1}). \tag{5.31}$$

Then, using $f_{W_S^k}(y^k) \geq f_{W^k}(y^k)$ by step 4 of the algorithm, the linearity of $f_{W_S^k}$, (5.31), Cauchy-Schwarz, and the definition of L^k (5.14), we find

$$\begin{aligned}
0 \leq \varepsilon_k := f_{W_S^k}(y^k) - f_{W^k}(y^k) &= f_{W_S^k}(y^{k+1}) - f_{W^k}(y^k) + \left\langle \nabla f_{W_S^k}, y^k - y^{k+1} \right\rangle \\
&\leq f_{W^{k+1}}(y^{k+1}) - f_{W^k}(y^k) + \left\|\nabla f_{W_S^k}\right\| \left\|y^k - y^{k+1}\right\| \\
&= L^K(y^{k+1}, W^{k+1}) - L^K(y^k, W^k) - \frac{u}{2}\left\|y^{k+1} - \hat{y}\right\|^2 + \frac{u}{2}\left\|y^k - \hat{y}\right\|^2 + \left\|\nabla f_{W_S^k}\right\| \left\|y^k - y^{k+1}\right\|.
\end{aligned}$$

From (5.30) it follows that $L^K(y^{k+1}, W^{k+1}) - L^K(y^k, W^k) \downarrow 0$. Furthermore, the last term satisfies $\|\nabla f_{W_S^k}\| \|y^k - y^{k+1}\| \to 0$ because of the boundedness of $\nabla f_{W_S^k}$ (see (5.9)) and $\|y^{k+1} - y^k\|^2 \to 0$. Finally, $\frac{u}{2}\|y^k - \hat{y}\|^2 - \frac{u}{2}\|y^{k+1} - \hat{y}\|^2 \to 0$ because the y^k are bounded and $\|y^{k+1} - y^k\|^2 \to 0$. This proves $f_{W_S^k}(y^k) - f_{W^k}(y^k) \to 0$.

In order to verify (5.29) let $y_* \in \mathbb{R}^m$ and $W_* \in \operatorname{Argmax}_{W \in \mathcal{W}} L^K(y_*, W)$ satisfy $L^K(y_*, W_*) = \min_y f(y) + \frac{u}{2}\|y - \hat{y}\|^2$. Since the quadratic term is added to the *convex* function f, we have $f(y^k) + \frac{u}{2}\|y^k - \hat{y}\|^2 \geq L^K(y_*, W_*) + \frac{u}{2}\|y^k - y_*\|^2$. On the other hand, $y^k = y^K_{\min}(W^k)$ yields

$$L^K(y_*, W_*) \geq L^K(y_*, W^k) \geq L^K(y^k, W^k) = L^K(y^k, W_S^k) - \varepsilon_k.$$

We obtain

$$f(y^k) + \frac{u}{2}\left\|y^k - \hat{y}\right\|^2 - L^K(y^k, W_S^k) + \varepsilon_k \geq \frac{u}{2}\left\|y^k - y_*\right\|^2. \tag{5.32}$$

Since $L^K(y^k, W_S^k) = \max_{W\in\mathcal{W}} L^K(y^k, W) = f(y^k) + \frac{u}{2}\left\|y^k - \hat{y}\right\|^2$, the claim follows from $\varepsilon_k \to 0$. ■

We proceed to the case that no descent steps occur in spite of the presence of the descent test. By the previous lemma and the descent test this can only happen if the gap between $f(\hat{y}^k)$ and $f_{W^{k+1}}(y^{k+1})$ decreases to zero with the same rate as the gap between $f(\hat{y}^k)$ and $f_{W_S^{k+1}}(y^{k+1})$. Because of the quadratic term in the augmented model this forces the y^k to converge to $\hat{y}^k$.

Lemma 5.3.4 *If in Algorithm 5.2.4 no descent steps occur starting with iteration K then $\hat{y}^K \in$ Argmin f.*

Proof. Let $\hat{y} = \hat{y}^K = \hat{y}^{K+1} = \ldots$. By assumption the descent test of Algorithm 5.2.4 is violated for all $k \geq K$ and therefore

$$f_{W_S^k}(y^k) - f_{W^k}(y^k) > (1-\kappa)[f(\hat{y}) - f_{W^k}(y^k)] \geq 0, \tag{5.33}$$

where nonnegativity follows from Lemma 5.3.2. By (5.28) this shows $f_{W^k}(y^k) \to f(\hat{y})$. On the other hand, $W^k \in \mathcal{W}$, the definition of L^k (5.14), and (5.26) imply

$$f(\hat{y}) \geq L^k(\hat{y}, W^k) = L^k(y^k, W^k) + \frac{u}{2}\left\|\hat{y} - y^k\right\|^2 = f_{W^k}(y^k) + u\left\|\hat{y} - y^k\right\|^2.$$

Thus $y^k \to \hat{y}$. Furthermore, by (5.17), $\nabla f_{W^k} = u(\hat{y} - y^k) \to 0$. Now (5.9) yields for all $y \in \mathbb{R}^m$

$$f(y) \geq f_{W^k}(y^k) + \left\langle \nabla f_{W^k}, y - y^k \right\rangle \to f(\hat{y}) + \left\langle 0, y - y^k \right\rangle = f(\hat{y}).$$

This proves $\hat{y} \in \operatorname{Argmin}_y f(y)$. ■

In the following we may concentrate on the case of infinitely many descent steps. In order to simplify notation we drop all iterates corresponding to null steps (indices k that do not satisfy $\hat{y}^k = y^k$) and equip the remaining iterates and the corresponding W^k with a new index h so that for all h

$$y^{h+1} = y^h_{\min}(W^{h+1}), \tag{5.34}$$

$$f(y^h) - f(y^{h+1}) \geq \kappa[f(y^h) - f_{W^{h+1}}(y^{h+1})] > 0. \tag{5.35}$$

If $f(y^h) \downarrow \inf f$ then there is nothing to prove. Therefore we consider the case that the $f(y^h)$ remain bounded from below by $f(\bar{y})$ for some $\bar{y} \in \mathbb{R}^m$.

Lemma 5.3.5 *If*

$$f(y^h) \geq f(\bar{y}) \qquad \text{for some } \bar{y} \in \mathbb{R}^m \text{ and all } h \tag{5.36}$$

then the y^h converge to a minimizer of f.

Proof. By (5.36) and (5.11),

$$f(y^h) \geq f(\bar{y}) \geq f_{W^{h+1}}(y^{h+1}) + \left\langle \nabla f_{W^{h+1}}, \bar{y} - y^{h+1} \right\rangle,$$

and by (5.34) and (5.17),

$$\nabla f_{W^{h+1}} = u(y^h - y^{h+1}). \tag{5.37}$$

Therefore the distance of y^{h+1} to $\bar{y}$ can be bounded by

$$\begin{aligned}
\left\|\bar{y} - y^{h+1}\right\|^2 &= \left\|\bar{y} - y^h + y^h - y^{h+1}\right\|^2 \\
&\leq \left\|\bar{y} - y^h\right\|^2 + 2\left\langle \bar{y} - y^h, y^h - y^{h+1} \right\rangle + 2\left\langle y^h - y^{h+1}, y^h - y^{h+1} \right\rangle \\
&= \left\|\bar{y} - y^h\right\|^2 + 2\left\langle \bar{y} - y^{h+1}, \nabla f_{W^{h+1}}/u \right\rangle \\
&\leq \left\|\bar{y} - y^h\right\|^2 + \frac{2}{u}[f(y^h) - f_{W^{h+1}}(y^{h+1})].
\end{aligned}$$

Iterating this argument yields for $h > H$

$$\left\|\bar{y} - y^h\right\|^2 \leq \left\|\bar{y} - y^H\right\|^2 + \frac{2}{u}\sum_{i=H}^{\infty}[f(y^i) - f_{W^{i+1}}(y^{i+1})]. \tag{5.38}$$

Because the sum over all descents $f(y^h) - f(y^{h+1})$ cannot exceed $f(y^0) - f(\bar{y})$, we obtain from (5.35)

$$\kappa\sum_{h=0}^{\infty}[f(y^h) - f_{W^{h+1}}(y^{h+1})] \leq \sum_{h=0}^{\infty}[f(y^h) - f(y^{h+1})] \leq f(y^0) - f(\bar{y}). \tag{5.39}$$

This together with (5.38) shows that the y^h are bounded and have an accumulation point $\tilde{y}$. We may replace $\bar{y}$ with $\tilde{y}$ in (5.38) and choose H so that the right hand side of (5.38) is smaller than some arbitrary $\varepsilon > 0$. This shows that $y^h \to \tilde{y}$ and, via (5.37), $\nabla f_{W^{h+1}} \to 0$. Eq. (5.39) implies $f(y^h) - f_{W^{h+1}}(y^{h+1}) \to 0$ and therefore $f_{W^{h+1}}(y^{h+1}) \to f(\tilde{y})$. Now $f(y) \geq f_{W^{h+1}}(y^{h+1}) + \left\langle \nabla f_{W^{h+1}}, y - y^{h+1}\right\rangle \to f(\tilde{y})$ for all $y \in \mathbb{R}^m$ completes the proof. ■

We are ready to state the main theorem.

Theorem 5.3.6 *Either $\hat{y}^k \to \bar{y} \in \operatorname{Argmin}_{y\in\mathbb{R}^m} f(y)$, or $\operatorname{Argmin}_{y\in\mathbb{R}^m} f(y) = \emptyset$ and $\|\bar{y}\| \to \infty$. In both cases $f(\hat{y}^k) \downarrow \inf_{y\in\mathbb{R}^m} f$.*

Proof. Lemma 5.3.4 shows $\hat{y}^k \to \bar{y} \in \operatorname{Argmin}_y f(y)$ for a finite number of descent steps, Lemma 5.3.5 does the same for an infinite number of descent steps satisfying (5.36). Otherwise there is no $\bar{y} \in \mathbb{R}^m$ so that $f^{\hat{y}^k} \geq f(\bar{y})$ for all k and thus $f(\hat{y}^k) \to \inf f$ by the definition of inf. Since f is continuous, boundedness of the y^k would imply attainment of the inf and therefore (5.36). ■

Remark 5.3.7 *The proof of convergence illustrates that (5.21) is only needed to force the null steps to produce a descent step over time. After the descent step, to $\hat{y}^{k+1}$ say, the algorithm could be restarted with $y^0 = y^{k+1}$ without endangering convergence.*

It is instructive to relate the W^k to the optimal solutions of the primal problem (P) via Proposition 5.1.1.

Theorem 5.3.8 *Let $\mathcal{X}_* := \operatorname{Argmax}_X \{\langle C, X\rangle : \mathcal{A}X = b, \operatorname{tr} X = 1, X \succeq 0\}$ denote the set of optimal solutions of the primal semidefinite program corresponding to $\min_y f(y)$ and assume that $\operatorname{Argmin}_y f(y) \neq \emptyset$. If k is finite with $k = K$ on termination then $W^{K+1} \in \mathcal{X}_*$. If there is an infinite number of iterations and no descent steps occur after iteration K then all accumulation points of W^k lie in $\mathcal{X}_*$. Otherwise all accumulation points of the W^h giving rise to descent steps (see (5.34)) lie in $\mathcal{X}_*$.*

Proof. Following the proof of Proposition 5.1.1 it can be worked out that $\min_y f(y)$ is equivalent to the dual of the primal-dual pair of semidefinite programs.

$$(\mathrm{P}_\lambda) \qquad \begin{array}{ll} \max & \langle C, X\rangle \\ \text{s.t.} & \mathcal{A}X = b \\ & \operatorname{tr} X = 1 \\ & X \succeq 0, \end{array} \qquad\qquad (\mathrm{D}_\lambda) \qquad \begin{array}{ll} \min & \lambda + b^T y \\ \text{s.t.} & Z = \lambda I + \mathcal{A}^T y - C \\ & Z \succeq 0. \end{array}$$

First consider the case of finite k with $k = K$ on termination. By assumption, $\varepsilon = 0$ and by step 3 of Algorithm 5.2.4, $f(\hat{y}^K) = f_{W^{K+1}}(y^{K+1})$. Then the proof of Lemma 5.3.2 shows $\nabla f_{W^{K+1}} = b - \mathcal{A}W^{K+1} = 0$ and with $W^{K+1} \in \widehat{\mathcal{W}}^K \subseteq \mathcal{W}$ it follows from (5.4) that W^{K+1} is feasible for (P_λ). Furthermore, $\left\langle C, W^K\right\rangle = \left\langle C, W^K\right\rangle + \left\langle b - \mathcal{A}W^{K+1}, y^{K+1}\right\rangle = \left\langle C - \mathcal{A}^T y^{K+1}, W^{K+1}\right\rangle + b^T y^{K+1} = f_{W^{K+1}}(y^{K+1}) = f(\hat{y}^K)$, proving $W^{K+1} \in \mathcal{X}_*$.

Now assume that the number or iterations is infinite and there are no descent steps after iteration K. By (5.12) and (5.4), all $W^{k+1} \in \widehat{\mathcal{W}}^k \subseteq \mathcal{W}$ and $\mathcal{W}$ is compact. Therefore the W^k have accumulation points and these are contained in $\mathcal{W}$. Denote by S a subsequence with $W^k \xrightarrow{S} W_* \in \mathcal{W}$. The proof of Lemma 5.3.4 shows that $y^k \xrightarrow{S} \tilde{y} \in \operatorname{Argmin}_y f(y)$,

$f_{W^k}(y^k) \xrightarrow{S} f(\tilde{y})$, and $\nabla f_{W^k} = b - \mathcal{A}W^k \xrightarrow{S} 0$. Then by the continuity of $\mathcal{A}$ it follows that $b - \mathcal{A}W_* = 0$ and $f_{W^k}(y^k) = \langle C, W^k \rangle + \langle b - \mathcal{A}W^k, y^k \rangle \xrightarrow{S} \langle C, W^k \rangle + \langle 0, \tilde{y} \rangle = f(\tilde{y})$, thus $W_* \in \mathcal{X}_*$.

For an infinite number of descent steps observe that, by assumption, $\operatorname{Argmin}_y f(y) \neq \emptyset$ and therefore the setting of Lemma 5.3.5 applies. The proof can be completed as in the null step case. ■

Cutting Plane Models

In constructing $\widehat{\mathcal{W}}^{k+1}$ from $\widehat{\mathcal{W}}^k$, (5.21) is all that is needed to guarantee convergence. The minimal choice within the framework (5.12) is $\overline{W}_{k+1} = W^{k+1}$ and $P_{k+1} = v^{k+1}$ where v^{k+1} is an eigenvector to the maximum eigenvalue of $C - \mathcal{A}^T y^{k+1}$. One should not, however, be tempted to believe that aggregation is restricted to $\overline{W}$.

In order to illustrate the power of aggregation in P let us consider a second extreme choice: Use P_k exclusively and ignore $\overline{W}_k$ (set α to zero in (5.12)). In this case we have to provide sufficiently many columns r in P so that we can reconstruct W^{k+1}. The trivial upper bound on r is n, but a much better bound is available via Lemma 2.3.1 if m is small.

Lemma 5.3.9 *If* $\widehat{\mathcal{W}}^k = \{P_k V P_k^T : \operatorname{tr} V = 1, V \in S_r^+\}$ *with* $P_k \in M_{n,r}$, $P_k^T P_k = I$, *and* $W_S^{k+1} = v^{k+1}(v^{k+1})^T$ *then* (5.21) *can be ensured whenever* $r \geq \bar{r} + 1$ *where* $\bar{r} \in \mathbb{N}$ *is the largest number satisfying* $\binom{\bar{r}+1}{2} \leq m + 1$.

Proof. By Lemma 5.2.2 and the uniqueness of y^{k+1}, all optimal solutions $W^{k+1} = P_k V^* P_k^T$ of (QSP) can be characterized as the set of optimal solutions of the following semidefinite program,

$$\begin{array}{ll} \max & \langle P_k^T(C - \mathcal{A}^T\hat{y}^k)P_k, V \rangle \\ \text{s.t.} & \mathcal{A}(P_k V P_k^T) = b + u(y^{k+1} - \hat{y}^k) \\ & \operatorname{tr} V = 1 \\ & V \succeq 0. \end{array}$$

The new constraint with respect to (QSP) is derived from the uniqueness condition on y^{k+1}, $y^k_{\min}(W^{k+1}) = y^{k+1}$ (5.17), and replaces the quadratic cost term that is now constant. So the new program has the same optimal solutions as (QSP). It is a linear program with strictly feasible dual due to the constraint $\operatorname{tr} V = 1$ (Proposition 5.1.1), therefore the optimal solution is attained in a face of dimension 0. The number of constraints is $m + 1$. By Lemma 2.3.1, the rank of matrices contained in 0 dimensional faces of the feasible set is at most $\bar{r}$ with $\binom{\bar{r}+1}{2} \leq m + 1$. Thus, the program has an optimal solution V_* of rank at most $\bar{r}$. Let $V_* = Q\Lambda Q^T$ denote an eigenvalue decomposition of V_* with $Q \in M_{n,\bar{r}}$ and $Q^T Q = I$ and Λ the diagonal matrix of eigenvalues $\lambda_1 \geq \ldots \geq \lambda_{\bar{r}} > 0$. Let $P_{k+1} = \operatorname{orth}\left[P_k Q, v^k\right]$. Then P_{k+1} has at most $\bar{r} + 1$ columns and $\widehat{\mathcal{W}}^{k+1}$ contains $W^{k+1} = P_k Q \Lambda (P_k Q)^T$ as well as $v^{k+1}(v^{k+1})^T$ by following the line of argument in the proof of Proposition 5.2.3. This completes the proof. ■

Various other choices of sets $\widehat{\mathcal{W}}^k$ may turn out useful in particular applications. For example, in the case of several primal matrices one may wish to use a model with several matrices V and $\overline{W}$ such as

$$\widehat{\mathcal{W}}^k = \left\{ \sum_{i=1}^{n_V} P_k^i V_i (P_k^i)^T + \sum_{i=1}^{n_{\overline{W}}} \alpha_i \overline{W}_k^i : \operatorname{tr} \sum_{i=1}^{n_V} V_i + \sum_{i=1}^{n_{\overline{W}}} \alpha_i = 1, \quad V_i \succeq 0 \text{ for } i = 1, \ldots, n_V, \right.$$
$$\left. \alpha_i \geq 0 \text{ for } i = 1, \ldots, n_{\overline{W}} \right\},$$

with appropriately defined P_k^i and $\overline{W}_k^i$.

The main design criterion for a model will be the efficient computability of the corresponding quadratic semidefinite subproblem. This will have to take into account the trade off between the quality of the model, which strongly influences the number of iterations required to achieve the desired precision, and the cost of evaluating f itself.

5.4 The Spectral Bundle Method with Bounds

So far we have assumed that y is unconstrained. We have exploited this in (5.17) in computing the unconstrained minimizer of $\min_y L^k(y,W)$. The latter step made it possible to reduce the number of variables in the subproblem (QSP) of Lemma 5.2.2 from m to $\binom{r+1}{2}$ (with r a choosable parameter).

With respect to the primal problem (P) of Section 5.1, coordinate y_i of y is unconstrained if the i-th primal constraint is an equality constraint. Thus, the spectral bundle algorithm 5.2.4 is designed for primal equality constraints only. This restriction, paired with the requirement of constant trace on the primal feasible set, rules out many applications of practical importance. For example, several semidefinite relaxations contain inequalities in their basic formulation, *e.g.*, quadratic 0-1 knapsack and max-k-cut. The standard technique of combinatorial optimization of improving initial relaxations by cutting planes relies on primal inequality constraints.

In this section we will show how the spectral bundle method can be extended to primal inequality constraints (we cannot do without assumption (5.1), though). In the eigenvalue problem this corresponds to requiring $y_i \geq 0$ for some of the coordinates. In order to keep notation simple, we assume in this section only that $y \in Y = \mathbb{R}^m_+$, but it is not difficult to extend this approach to a combination of unconstrained and bounded variables (bounded in the sense that $l_i \leq y_i \leq u_i$ with $l_i, u_i \in \mathbb{R}$). In particular, we develop the approach for the problem

$$\min_{y \geq 0} \lambda_{\max}(C - \mathcal{A}^T y) + b^T y.$$

Restated in terms of f (cf. (5.3)) and the indicator function $\imath_Y$ of Y ($\imath_Y(y) = 0$ if $y \in Y$, ∞ otherwise), the problem reads

$$\min_{y \in \mathbb{R}^m} f_Y \qquad \text{with} \quad f_Y(y) := f(y) + \imath_Y(y).$$

Again, we set up linear minorants for f_Y,

$$f_{W,\eta}(y) := \langle C - \mathcal{A}^T y, W \rangle + \langle b - \eta, y \rangle \tag{5.40}$$

where $W \in \mathcal{W}$, with $\mathcal{W}$ as in (5.4), are the subgradients of $\lambda_{\max}(\cdot)$ and $\eta \in \mathbb{R}^m_+$ are the Lagrange multipliers to the nonnegativity constraints on y (or subgradients of $\imath_Y$). With this notation and (5.5),

$$f_Y(y) = \sup_{W \in \mathcal{W}, \eta \in \mathbb{R}^m_+} f_{W,\eta}(y).$$

The subdifferential of f_Y at $y \in Y$ is

$$\partial f_Y(y) = \left\{ \nabla f_{W,\eta} : W \in \mathcal{W}, \eta \in \mathbb{R}^m_+, f_{W,\eta}(y) = f(y) \right\}. \tag{5.41}$$

Similar to (5.11) we describe cutting plane models of f_Y by

$$f_{\widehat{\mathcal{W}},\eta}(y) := \max_{W \in \widehat{\mathcal{W}}} f_{W,\eta}(y) \leq f_Y(y) \qquad \text{for all } \widehat{\mathcal{W}} \subseteq \mathcal{W}, \eta \in \mathbb{R}^m_+, y \in \mathbb{R}^m. \tag{5.42}$$

For reasons that will become clear soon, we do not extend η into a set in $f_{\widehat{\mathcal{W}},\eta}$. We employ $\widehat{\mathcal{W}}^k$ of (5.12) and define, as in (5.14), the augmented Lagrangian with respect to $\hat{y}^k \in Y$,

$$L^k(y; W, \eta) := f_{W,\eta}(y) + \frac{u}{2} \left\| y - \hat{y}^k \right\|^2. \tag{5.43}$$

Following the unconstrained case, we would like to compute the next trial point y^{k+1} as the minimizer of the augmented problem

$$\min_{y \in \mathbb{R}^m} \sup_{W \in \widehat{\mathcal{W}}^k, \eta \in \mathbb{R}^m_+} L^k(y; W, \eta) \tag{5.44}$$

via the dual problem

$$\max_{W\in\widehat{\mathcal{W}}^k,\eta\in\mathbb{R}^m_+}\min_{y\in\mathbb{R}^m} L^k(y;W,\eta) \tag{5.45}$$

by exploiting the unconstrained inner minimization over y in 5.45,

$$y^k_{\min}(W,\eta) := \hat{y}^k + \frac{1}{u}(\mathcal{A}W - b + \eta) = \hat{y}^k - \frac{1}{u}\nabla f_{W,\eta}. \tag{5.46}$$

Indeed, a saddle point exists for (5.44) and (5.45)[1], but (5.45) still contains more than m variables due to η, so the advantage in efficiency is lost.

We regain efficiency by following a slightly weaker approach. Instead of solving (5.45) directly, we approximate its solution by a sequence of "coordinatewise" optimization steps. First, fix $\hat{\eta}$ and solve

$$\max_{W\in\widehat{\mathcal{W}}^k}\min_{y\in\mathbb{R}^m} L^k(y;W,\hat{\eta}) \tag{5.47}$$

for an optimal W^+. The latter step is identical to the situation described in Lemma 5.2.2 with b replaced by $b-\hat{\eta}$. Rewritten with $\hat{\eta}$ the Lemma reads

Lemma 5.4.1 *Let L^k be as defined in (5.43) and $\hat{\eta}\in\mathbb{R}^m_+$, then*

$$\min_y \max_{W\in\widehat{\mathcal{W}}^k} L^k(y;W,\hat{\eta}) = L^k(y^{+\frac{1}{2}};W^+,\hat{\eta}) = \max_{W\in\widehat{\mathcal{W}}^k}\min_y L^k(y;W,\hat{\eta}) \tag{5.48}$$

with $y^{+\frac{1}{2}} = y^k_{\min}(W^+,\hat{\eta})$ unique, and W^+ an optimal solution of

$$(\mathrm{QSP}_{\hat{\eta}}) \qquad \begin{array}{ll} \min & \frac{1}{2u}\|b-\hat{\eta}-AW\|^2 - \langle W, C-\mathcal{A}^T\hat{y}^k\rangle - \langle b-\hat{\eta},\hat{y}^k\rangle \\ \text{s.t.} & W = P_kVP_k^T + \alpha\overline{W}_k \\ & \operatorname{tr}V+\alpha = 1 \\ & V\succeq 0, \alpha\geq 0. \end{array}$$

Proof. Replace b with $b-\hat{\eta}$ in Lemma 5.2.2. ∎

Next we keep W^+ fixed and solve

$$\max_{\eta\in\mathbb{R}^m_+}\min_y L^k(y;W^+,\eta) \tag{5.49}$$

for an optimal η^+. For this problem the optimal solution is particularly easy to compute.

Lemma 5.4.2 *Let L^k be as defined in (5.43) and let $W^+\in\widehat{\mathcal{W}}^k$. Then*

$$\min_{y\in\mathbb{R}^m}\sup_{\eta\in\mathbb{R}^m_+} L^k(y,W^+,\eta) = L^k(y^+,W^+,\eta^+) = \max_{\eta\in\mathbb{R}^m_+}\min_{y\in\mathbb{R}^m} L^k(y,W^+,\eta) \tag{5.50}$$

with

$$\eta^+ = \eta^k_{\max}(W^+) := \max\left\{0, -u\left[\hat{y}^k + \frac{1}{u}(\mathcal{A}W^+ - b)\right]\right\} \tag{5.51}$$

and (cf. (5.46))

$$y^+ = y^k_{\min}(W^+,\eta^+)\geq 0 \quad \textit{satisfying} \quad \langle y^+,\eta^+\rangle = 0. \tag{5.52}$$

Proof. We first show that (y^+,W^+,η^+) is a solution of the right hand side problem of (5.50). Substituting $y^k_{\min}(W^+,\eta)$ of (5.46) for y in (5.49) we obtain

$$(\mathrm{QP}) \quad \min_{\eta\in\mathbb{R}^m_+}\frac{1}{2u}\|\eta\|^2 + \left\langle \hat{y}^k + \frac{1}{u}(AW^+ - b),\eta\right\rangle - \langle W^+, C\rangle - \langle b-\mathcal{A}W,\hat{y}^k\rangle + \frac{1}{2u}\left\|b-\mathcal{A}W^+\right\|^2.$$

[1] See Rockafellar [1970], Theorem 37.6.

This problem is separable convex in η, *i.e.*, the optimal solution is obtained by solving (QP) for each coordinate of η independently. The unconstrained first order optimality conditions for η_i read

$$\frac{1}{u}\eta_i + \hat{y}^k + \frac{1}{u}(\mathcal{A}W^+ - b) = 0.$$

In combination with the sign constraint $\eta_i \geq 0$ and the convexity of the cost function of (QP) this yields (5.51). Now (5.52) is easily verified by direct computation.

In (5.50) the value of the left hand side problem is at least the value of the right hand side problem because the $\inf_y$ of the $\sup_\eta$ is always greater or equal to the $\sup_\eta$ of the $\inf_y$. Thus it suffices to exhibit a solution y for the left hand side that attains $L(y^+, W^+, \eta^+)$. This solution is y^+: Since $y^+ \geq 0$, the inner sup is attained for any $\bar{\eta}$ with $\langle y^+, \bar{\eta}\rangle = 0$ and in particular for η^+. ■

In general the point (W^+, η^+) will not be the optimal solution of (5.45) because W^+ is usually not the best choice within $\widehat{\mathcal{W}}^k$ for the new η^+. The gap $f_{\widehat{\mathcal{W}}^k,\eta^+}(y^+) - f_{W^+,\eta^+}(y^+)$ is easy to compute (see Proposition 5.2.1) and provides a good indicator for the quality of this approximate solution.

Lemma 5.4.3 *Let y^+, W^+, and η^+ be given as in Lemmas 5.4.1 and 5.4.2. Then $f_{W^+,\eta^+}(y^+) \leq f_{\widehat{\mathcal{W}}^k,\eta^+}(y^+)$. If equality holds then (y^+, W^+, η^+) is a saddle point of* (5.44) *and* (5.45).

Proof. The inequality follows from $W^+ \in \widehat{\mathcal{W}}^k$ (see Lemma 5.4.1) and the definition of $f_{\widehat{\mathcal{W}}^k,\eta^+}$ in (5.42). If equality holds then $L^k(y^+, W^+, \eta^+) = L^k(y^k_{min}(W^+, \eta^+), W^+, \eta^+)$ is a lower bound for (5.45). In order to see that it is an upper bound for (5.44) observe that the inner sup is separable into $\sup_\eta$ and $\max_{\widehat{\mathcal{W}}^k}$. The supremum over $\eta \in \mathbb{R}^m_+$ is attained for η^+ because $y^+ \geq 0$ and $\langle y^+, \eta^+\rangle = 0$ (cf. (5.52)). The maximum over $W \in \widehat{\mathcal{W}}^k$ is attained for W^+ by assumption. ■

If this gap is relatively large, say, for some parameter $\kappa_M \in (0, \infty]$

$$f_{\widehat{\mathcal{W}}^k,\eta^+}(y^+) - f_{W^+,\eta^+}(y^+) \geq \kappa_M[f(\hat{y}^k) - f_{W^+,\eta^+}(y^+)], \tag{5.53}$$

then we iterate these coordinatewise maximization steps by setting $\hat{\eta} = \eta^+$ and continue with (5.47). Otherwise we set $y^{k+1} = y^+$, $W^{k+1} = W^+$, and $\eta^{k+1} = \eta^+$ and continue as in the standard spectral bundle method.

Algorithm 5.4.4 *(Spectral Bundle Method with Bounds)*

Input: $y^0 \in \mathbb{R}^n$, $\varepsilon \geq 0$, $\kappa \in (0,1)$, $\kappa_M \in (0, \infty]$, *a weight* $u > 0$.

1. *Set $k = 0$, $\hat{y}^0 = y^0$, compute $f(y^0)$ and $\widehat{\mathcal{W}}^0$, set $\eta^0 = \eta^0_{\max}(W)$ for some $W \in \widehat{\mathcal{W}}^0$.*
2. *(Trial point finding). Set $\hat{\eta} = \eta^k$.*
   (a) *Find $W^+ \in \operatorname{Argmax}_{W \in \widehat{\mathcal{W}}^k} \min_{y \in \mathbb{R}^m} L^k(y; W, \hat{\eta})$ (see Lemma 5.4.1).*
   (b) *Set $\eta^+ = \eta^k_{\max}(W^+)$, $y^+ = y^k_{\min}(W^+, \eta^+)$ (see Lemma 5.4.2).*
   (c) *(Stopping criterion). If $f(\hat{y}^k) - f_{W^+,\eta^+}(y^+) \leq \varepsilon(|f(\hat{y}^k)| + 1)$ then* **stop**.
   (d) *If $f_{\widehat{\mathcal{W}}^k,\eta^+}(y^+) - f_{W^+,\eta^+}(y^+) > \kappa_M[f(\hat{y}^k) - f_{W^+,\eta^+}(y^+)]$ then set $\hat{\eta} = \eta^+$ and* **goto** *(a).*
   (e) *Set $y^{k+1} = y^+$, $W^{k+1} = W^+$, $\eta^{k+1} = \eta^+$.*
3. *(Evaluation). Find $W_S^{k+1} \in \operatorname{Argmax}_{W \in \mathcal{W}} \langle C - \mathcal{A}^T y^{k+1}, W\rangle$ and determine $f(y^{k+1})$.*
4. *(Descent test). If $f(\hat{y}^k) - f(y^{k+1}) \geq \kappa[f(\hat{y}^k) - f_{W^{k+1},\eta^{k+1}}(y^{k+1})]$, then set $\hat{y}^{k+1} = y^{k+1}$, $\eta^{k+1} = \eta^{k+1}_{\max}(W_S^{k+1})$ (descent step); otherwise set $\hat{y}^{k+1} = \hat{y}^k$ (null step).*
5. *(Model updating). Choose a $\widehat{\mathcal{W}}^{k+1} \supset \{W^{k+1}, W_S^{k+1}\}$ of the form* (5.12).
6. *Increase k by one and* **goto** *2.*

Remark 5.4.5

1. *Observe that we allow $\kappa_M = \infty$. For this choice the inner loop in step 2 is executed exactly once. We will show below that just one update of the Lagrange multipliers η suffices to guarantee convergence. In practice, κ_M should be selected smaller than $1-\kappa$ in order to avoid null steps due to model imprecision. Indeed, suppose $y^{k+1} = y^+$ is accepted as candidate even though $f_{\widehat{\mathcal{W}}^k,\eta^+}(y^+) - f_{W^+,\eta^+}(y^+) > (1-\kappa)[f(\hat{y}^k) - f_{W^+,\eta^+}(y^+)]$. Then the value of cutting plane model $f_{\widehat{\mathcal{W}}^k}(y^+)$ (a lower bound on $f(y^+)$) ensures a null step in step 4.*

2. *With respect to the primal semidefinite program (P) of Section 5.1, the Lagrange multipliers η may be interpreted as primal slack variables that transform inequalities $\langle A_i, X\rangle \leq b_i$ into equalities $\langle A_i, X\rangle + \eta_i = b_i$.*

3. *With respect to Remark 5.2.5, 2, the slack variables η_i, together with X, form a larger block matrix that does not satisfy (5.1) in the dual. However, due to the special structure, it is still possible to find back into the feasible region via Lemma 5.4.2.*

The proof of convergence follows the same line of argument that was used for the standard spectral bundle method. The first technical result matches Proposition 5.3.1.

Proposition 5.4.6

$$L^k(y; W, \eta) = L^k(y^k_{\min}(W,\eta), W, \eta) + \frac{u}{2}\left\|y - y^k_{\min}(W)\right\|^2 \tag{5.54}$$

$$L^k(y^{k+1}; W^{k+1}, \eta^{k+1}) \leq f(\hat{y}^k) \tag{5.55}$$

Proof. Analogous to the proof of Proposition 5.3.1. ■

We show that for $\varepsilon = 0$ the stopping criterion identifies optimal solutions correctly.

Lemma 5.4.7 *$f(\hat{y}^k) \geq f_{W^{k+1},\eta^{k+1}}(y^{k+1})$ and if $f(\hat{y}^k) = f_{W^{k+1},\eta^{k+1}}(y^{k+1})$ then $\hat{y}^k \in Y$ is optimal.*

Proof. Together with $y^{k+1} \geq 0$ and $\left\langle \eta^{k+1}, y^{k+1}\right\rangle = 0$ (cf. (5.52)) the proof may be completed as for Lemma 5.3.2. ■

We first prove the lemma corresponding to Lemma 5.3.3 for $\kappa_M = \infty$, *i.e.*, the inner loop in step 2 is executed exactly once per outer iteration. The proof is identical to the proof of Lemma 5.3.3 up to the fact that we consider basically two subgradient steps in one. Once this result is established, the proof for general $\kappa_M \in (0, \infty]$ will follow easily.

Lemma 5.4.8 *Let $\kappa_M = \infty$ and assume that, starting with iteration K, Algorithm 5.4.4 is run without descent steps, $\hat{y} = \hat{y}^K = \hat{y}^{K+1} = \ldots$. Then*

$$\varepsilon_k := f_{W^k_S,\eta^k}(y^k) - f_{W^k,\eta^k}(y^k) \to 0, \tag{5.56}$$

$$y^k \to \operatorname*{argmin}_y \max_{W\in\mathcal{W},\eta\in\mathbb{R}^m_+} L^k(y; W, \eta) = \operatorname*{argmin}_{y\in Y} f(y) + \frac{u}{2}\left\|y - \hat{y}\right\|^2. \tag{5.57}$$

Proof. Note that the functions $L^K = L^{K+1} = \ldots$ (5.43) and $y^K_{\min} = y^{K+1}_{\min} \ldots$ (5.46) are identical for all $k \geq K$. The $\eta^k = \eta^k_{\max}(W^k)$ (5.51) and therefore also the $y^k = y^K_{\min}(W^k, \eta^k)$ remain bounded for $k \geq K$ because of the compactness of $\mathcal{W}$ (see (5.4)). Since $\kappa_M = \infty$, the inner loop in step 2 of Algorithm 5.4.4 is executed only once, and so $\hat{y} = y^k$ and $y^+ = y^{k+1}$. Let $y^{k+\frac{1}{2}} = y^K_{\min}(W^{k+1}, \eta^k)$ denote the intermediate y after computing W^{k+1}, but before computing η^{k+1}. Since $W^k \in \widehat{\mathcal{W}}^k$ by (5.21) and $W^{k+1} = \operatorname{argmax}_{W\in\widehat{\mathcal{W}}^k} L^k(y^{k+\frac{1}{2}}, W, \eta^k)$ by (5.48), we obtain from (5.54) for all $k \geq K$

$$\begin{aligned} L^K(y^k, W^k, \eta^k) + \frac{u}{2}\left\|y^{k+\frac{1}{2}} - y^k\right\|^2 &= L^K(y^{k+\frac{1}{2}}, W^k, \eta^k) \\ &\leq L^K(y^{k+\frac{1}{2}}, W^{k+1}, \eta^k). \end{aligned} \tag{5.58}$$

Likewise, $\eta^k \in \mathbb{R}^m_+$ and $\eta^{k+1} = \operatorname{argmax}_{\eta\in\mathbb{R}^m_+} L^k(y^{k+1}, W^{k+1}, \eta)$ by (5.50) imply via (5.54)

$$\begin{aligned} L^K(y^{k+\frac{1}{2}}, W^{k+1}, \eta^k) + \frac{u}{2}\left\|y^{k+1} - y^{k+\frac{1}{2}}\right\|^2 &= L^K(y^{k+1}, W^{k+1}, \eta^k) \\ &\le L^K(y^{k+1}, W^{k+1}, \eta^{k+1}) \le f(\hat{y}). \end{aligned} \tag{5.59}$$

The last inequality $L^K(y^{k+1}, W^{k+1}, \eta^{k+1}) \le f(\hat{y})$ follows from (5.55). So there exists an $f_* \in \mathbb{R}$ with $L^K(y^k, W^k, \eta^k) \uparrow f_* \le f(\hat{y})$. In addition, $\left\|y^{k+1} - y^k\right\|^2 \to 0$. By (5.21) $W^k_S \in \widehat{\mathcal{W}}^k$, and therefore (5.48) yields

$$f_{W^k_S,\eta^k}(y^{k+\frac{1}{2}}) \le \max_{W\in\widehat{\mathcal{W}}^k} f_{W,\eta^k}(y^{k+\frac{1}{2}}) = f_{W^{k+1},\eta^k}(y^{k+\frac{1}{2}}). \tag{5.60}$$

Then, using $f_{W^k_S,\eta^k}(y^k) \ge f_{W^k,\eta^k}(y^k)$ by step 3 of the algorithm, the linearity of $f_{W^k_S,\eta^k}$, (5.60), Cauchy-Schwarz, and (5.43), we find

$$\begin{aligned} 0 \le \varepsilon_k &:= f_{W^k_S,\eta^k}(y^k) - f_{W^k,\eta^k}(y^k) \\ &= f_{W^k_S,\eta^k}(y^{k+\frac{1}{2}}) - f_{W^k,\eta^k}(y^k) + \left\langle \nabla f_{W^k_S,\eta^k}, y^k - y^{k+\frac{1}{2}}\right\rangle \\ &\le f_{W^{k+1},\eta^k}(y^{k+\frac{1}{2}}) - f_{W^k,\eta^k}(y^k) + \left\|\nabla f_{W^k_S,\eta^k}\right\| \left\|y^k - y^{k+\frac{1}{2}}\right\| \\ &= L^K(y^{k+\frac{1}{2}}; W^{k+1}, \eta^k) - L^K(y^k; W^k, \eta^k) - \frac{u}{2}\left\|y^{k+\frac{1}{2}} - \hat{y}\right\|^2 + \frac{u}{2}\left\|y^k - \hat{y}\right\|^2 \\ &\quad + \left\|\nabla f_{W^k_S,\eta^k}\right\| \left\|y^k - y^{k+\frac{1}{2}}\right\|. \end{aligned}$$

From (5.58) and (5.59) we obtain $L^K(y^{k+1}; W^{k+1}, \eta^k) - L^K(y^k; W^k, \eta^k) \downarrow 0$. Next, observe that $\|\nabla f_{W^k_S,\eta^k}\|\|y^k - y^{k+\frac{1}{2}}\| \to 0$ because of the boundedness of $\nabla f_{W^k_S}$ (due to the boundedness of η^k and $\mathcal{W}$) and $\|y^{k+\frac{1}{2}} - y^k\|^2 \to 0$. Finally, $\frac{u}{2}\|y^k - \hat{y}\|^2 - \frac{u}{2}\|y^{k+\frac{1}{2}} - \hat{y}\|^2 \to 0$ because of the boundedness of y^k and $\|y^{k+\frac{1}{2}} - y^k\|^2 \to 0$. This proves $f_{W^k_S,\eta^k}(y^k) - f_{W^k,\eta^k}(y^k) \to 0$.

In order to verify (5.57), let $y_* \in Y$ and $W_* \in \operatorname{Argmax}_{W\in\mathcal{W}} L^K(y_*, W, 0)$ satisfy $L^K(y_*, W_*, 0) = \min_{y\in Y} f(y) + \frac{u}{2}\|y - \hat{y}\|^2$. By the quadratic term added to the convex function f, we have $f(y^k) + \frac{u}{2}\|y^k - \hat{y}\|^2 \ge L^K(y_*, W_*) + \frac{u}{2}\|y^k - y_*\|^2$. On the other hand, $y^k = y^K_{\min}(W^k, \eta^k)$ implies

$$L^K(y_*, W_*, 0) \ge L^K(y_*, W^k, 0) \ge L^K(y_*, W^k, \eta^k) \ge L^K(y^k, W^k, \eta^k) = L^K(y^k, W^k_S, \eta^k) - \varepsilon_k.$$

We obtain

$$f(y^k) + \frac{u}{2}\left\|y^k - \hat{y}\right\|^2 - L^K(y^k, W^k_S, \eta^k) + \varepsilon_k \ge \frac{u}{2}\left\|y^k - y_*\right\|^2.$$

Since $L^K(y^k, W^k_S, \eta^k) = \max_{W\in\mathcal{W}} L^K(y^k, W, \eta^k) = f(y^k) + \frac{u}{2}\left\|y^k - \hat{y}\right\|^2$ the claim follows from $\varepsilon_k \to 0$. ■

Next, for $\kappa_M \in (0, \infty]$, consider the case that the inner loop in step 2 is infinite.

Corollary 5.4.9 *If, for $\kappa_M \in (0, \infty]$, in iteration K Algorithm 5.4.4 always proceeds to 2(a) in step 2(d), then the sequences $f_{\widehat{\mathcal{W}}^k,\eta_+}(y^+) - f_{W^+,\eta^+} \to 0$ and $y^+ \to \operatorname{argmin}_{y\in Y} f_{\widehat{\mathcal{W}}^k} + \frac{u}{2}\|y - \hat{y}^k\|^2$. In this case $\hat{y}^K \in \operatorname{Argmin} f_Y$ and $f_{W^+,\eta^+}(y^+) \to f(\hat{y}^k)$.*

Proof. Observe, that the behavior of this algorithm is the same as if $\varepsilon_M = \infty$, no descent steps are accepted, the set $\mathcal{W}$ is set to $\widehat{\mathcal{W}}^k$, and in the model updating step 5 the set $\widehat{\mathcal{W}}^k$ is left unchanged. In this situation the proof of Lemma 5.4.8 applies verbatim. This shows $y^+ \to \operatorname{argmin}_{y\in Y} f_{\widehat{\mathcal{W}}^k} + \frac{u}{2}\|y - \hat{y}^k\|^2$ and $f_{\widehat{\mathcal{W}}^k,\eta^+}(y^+) - f_{W^+,\eta^+} \to 0$. The convergence of the latter proves $f_{W^+,\eta^+} \to f(\hat{y}^k)$ by the condition in step 2(d) of the algorithm with $\kappa_M > 0$. But (5.54) implies $f(\hat{y}) \ge L^K(\hat{y}, W^+, \eta^+) = L^K(y^+, W^+, \eta^+) + \frac{u}{2}\left\|\hat{y} - y^+\right\|^2 = f_{W^+,\eta^+}(y^{k+1}) + u\left\|\hat{y} - y^{k+1}\right\|^2$ and therefore $y^{k+1} \to \hat{y}$. ■

Finally, it remains to show (5.56) and (5.57) in the case that all inner loops of step 2 are finite.

Corollary 5.4.10 *If $\kappa_M \in (0,\infty]$ and $k \to \infty$ then (5.56) and (5.57) hold.*

Proof. In the proof of Lemma 5.4.8 use instead of indices $k+\frac{1}{2}$ and $k+1$ the data of the first iterate of the inner loop in iteration $k+1$. ■

The necessary modifications in lemmas 5.3.4 and 5.3.5 are straight forward. Therefore, Theorem 5.3.6 also applies to Algorithm 5.4.4.

5.5 The Quadratic Semidefinite Subproblem

In every iteration of the spectral bundle algorithm 5.2.4, the quadratic semidefinite subproblem (QSP) of Lemma 5.2.2 has to be solved (necessary modifications for inequalities are discussed at the end of this section). We do this in two steps. For implementational efficiency, we eliminate W in (QSP) and express the quadratic cost function in terms of V and α. The computation of these cost parameters is the first step. In the second step we solve the reduced quadratic semidefinite programming problem by an interior point approach.

In order to simplify notation, we drop the indices k of P, $\overline{W}$, $\hat{y}$, r, $\widehat{\mathcal{W}}$, L, $y_{\min}(\cdot)$, etc., in this section. After the substitution of W by $PVP^T+\alpha\overline{W}$ the quadratic cost function of (QSP) reads

$$\frac{1}{2u}\left\|\mathcal{A}(PVP^T+\alpha\overline{W})\right\|^2+\frac{1}{2u}\|b\|^2+\left\langle PVP^T+\alpha\overline{W},\mathcal{A}^T(\hat{y}-\frac{1}{u}b)-C\right\rangle-b^T\hat{y}.$$

Using the svec operator of Section 1.3, we obtain a reduced problem in V and α,

$$\begin{array}{lll}\min & \frac{1}{2}\begin{bmatrix}\operatorname{svec}V\\ \alpha\end{bmatrix}^T\begin{bmatrix}Q_{11} & q_{12}\\ q_{12}^T & q_{22}\end{bmatrix}\begin{bmatrix}\operatorname{svec}V\\ \alpha\end{bmatrix}+\begin{bmatrix}c_1\\ c_2\end{bmatrix}^T\begin{bmatrix}\operatorname{svec}V\\ \alpha\end{bmatrix}+d\\ \text{s.t.} & \alpha+s_I^T\operatorname{svec}V=1\\ & \alpha\geq 0, V\succeq 0,\end{array}\tag{5.61}$$

where, after some technical linear algebra,

$$\begin{array}{rcl}Q_{11} &=& \frac{1}{u}\sum_{i=1}^m \operatorname{svec}(P^TA_iP)\operatorname{svec}(P^TA_iP)^T\\ q_{12} &=& \frac{1}{u}\operatorname{svec}(P^T\mathcal{A}^T(\mathcal{A}\overline{W})P)\\ q_{22} &=& \frac{1}{u}\left\langle\mathcal{A}\overline{W},\mathcal{A}\overline{W}\right\rangle\\ c_1 &=& -\operatorname{svec}(P^T[\mathcal{A}^T(\frac{1}{u}b-\hat{y})+C]P)\\ c_2 &=& -(\left\langle\frac{1}{u}b-\hat{y},\mathcal{A}\overline{W}\right\rangle+\left\langle C,\overline{W}\right\rangle)\\ d &=& \left\langle b,\frac{1}{2u}b-\hat{y}\right\rangle\\ s_I &=& \operatorname{svec}I_r.\end{array}\tag{5.62}$$

Several remarks are in order.

Remark 5.5.1

1. *This problem in $\binom{r+1}{2}+1$ variables can be solved efficiently by interior point methods provided r is not too large, say, around 30. Since convergence is guaranteed even for $r=1$, the spectral bundle algorithm can be run for problems with a huge number m of variables (or constraints in terms of the primal problem (P)).*

2. *If storing $\overline{W}$ as a matrix is impossible due to memory restrictions, then it suffices to store and update the m-vector $\mathcal{A}\overline{W}$ and the scalar $\left\langle C,\overline{W}\right\rangle$. Indeed, only these terms appear in the computation of the cost coefficients. Due to the linearity of $\mathcal{A}$ and $\langle C,\cdot\rangle$, the update of $\mathcal{A}\overline{W}$ and $\left\langle C,\overline{W}\right\rangle$ corresponding to (5.24) is straight forward.*

3. *Almost all calculations involve the projected matrices $P^T A_i P \in S_r$. They are computed only once for each solution of* (5.61)*, and only one such matrix must be kept in memory if the values of Q_{11} to c_2 are accumulated.*

4. *The most expensive operation in computing the cost coefficients is the accumulation of Q_{11}, which involves the summation of m dyadic products of $\binom{r+1}{2}$-vectors in $O(mr^4)$ operations. For rather small r but sufficiently large m, this operation needs more time than solving the reduced quadratic semidefinite subproblem.*

Like in standard convex quadratic programming we may construct a dual to (5.61). To this end we introduce a Lagrange multiplier λ for the equality constraint, a dual slack matrix $U \succeq 0$ as complementary variable to V, and a dual slack scalar $\beta \geq 0$ as complementary variable to α. The dual to (5.61) reads

$$\begin{aligned} \max \quad & -\tfrac{1}{2} \begin{bmatrix} \operatorname{svec} V \\ \alpha \end{bmatrix}^T \begin{bmatrix} Q_{11} & q_{12} \\ q_{12}^T & q_{22} \end{bmatrix} \begin{bmatrix} \operatorname{svec} V \\ \alpha \end{bmatrix} - \lambda + d \\ \text{s.t.} \quad & \begin{bmatrix} \operatorname{svec} U \\ \beta \end{bmatrix} = \begin{bmatrix} Q_{11} & q_{12} \\ q_{12}^T & q_{22} \end{bmatrix} \begin{bmatrix} \operatorname{svec} V \\ \alpha \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + \lambda \begin{bmatrix} s_I \\ 1 \end{bmatrix} \\ & U \succeq 0, \beta \geq 0. \end{aligned} \tag{5.63}$$

Although the cost function does not look too familiar, it turns out to be (the negative of) the value of the augmented model at $y_{\min}(W)$ with $W = PVP^T + \alpha\overline{W}$, *i.e.*, $-L(y_{\min}(W), W)$, if λ is chosen as small as possible. For later reference we use the iteration index k within the statement of the lemma.

Lemma 5.5.2 *Let, for iteration k of Algorithm 5.2.4, (V, α) be a feasible point of* (5.61) *and (V, α, λ) a feasible point of* (5.63)*. Denote the negative of the primal and dual cost functions by*

$$q_p^k(V, \alpha) := -\frac{1}{2} \begin{bmatrix} \operatorname{svec} V \\ \alpha \end{bmatrix}^T \begin{bmatrix} Q_{11} & q_{12} \\ q_{12}^T & q_{22} \end{bmatrix} \begin{bmatrix} \operatorname{svec} V \\ \alpha \end{bmatrix} - \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}^T \begin{bmatrix} \operatorname{svec} V \\ \alpha \end{bmatrix} - d \tag{5.64}$$

and

$$q_d^k(V, \alpha, \lambda) := \frac{1}{2} \begin{bmatrix} \operatorname{svec} V \\ \alpha \end{bmatrix}^T \begin{bmatrix} Q_{11} & q_{12} \\ q_{12}^T & q_{22} \end{bmatrix} \begin{bmatrix} \operatorname{svec} V \\ \alpha \end{bmatrix} + \lambda - d. \tag{5.65}$$

Then, for $\tilde{W} = P_k^T V P_k + \alpha\overline{W}_k$ and $\tilde{y} = y_{\min}^k(\tilde{W})$,

$$q_p^k(V, \alpha) = L^k(\tilde{y}, \tilde{W}) \leq \max_{W \in \widehat{\mathcal{W}}^k} L^k(\tilde{y}, W) \leq q_d^k(V, \alpha, \lambda) \tag{5.66}$$

and $q_p^k(V, \alpha) = q_d^k(V, \alpha, \lambda)$ if and only if (V, α) is an optimal solution of (5.61) *and (V, α, λ) is an optimal solution of* (5.63)*.*

Proof. For notational convenience we work without iteration counters. The left hand side equality in (5.66) follows by the equivalence of (5.61) to (QSP) of Lemma 5.2.2 and the construction of (QSP) there. The middle inequality follows from $\tilde{W} \in \widehat{\mathcal{W}}$ and the choice of $\tilde{y}$. By the definition of L (5.14), the right hand side inequality is equivalent to

$$f_{\widehat{\mathcal{W}}}(\tilde{y}) + \frac{u}{2} \|\tilde{y} - \hat{y}\|^2 \leq q_d(V, \alpha, \lambda). \tag{5.67}$$

By expanding the coefficients of (5.62) in the slack variable U of (5.63) and by rearranging terms, we obtain with the definition of $y_{\min}$ (5.17),

$$U = -P^T \left[C + \mathcal{A}^T \left(\hat{y} + \frac{1}{u}(b - \mathcal{A}\tilde{W}) \right) \right] P + \lambda I = -P^T \left[C + \mathcal{A}^T \tilde{y} \right] P + \lambda I \succeq 0.$$

Thus, by the feasibility of (V, α, λ)

$$\lambda \geq \lambda_{\max}(P^T \left[C + \mathcal{A}^T\tilde{y}\right] P). \tag{5.68}$$

Likewise, the feasibility of slack variable β yields

$$\lambda \geq \left\langle C - \mathcal{A}^T\tilde{y}, \overline{W}\right\rangle. \tag{5.69}$$

The quadratic and the constant term of the cost function q_d expand to

$$\begin{aligned} &\frac{1}{2u}\left\|\mathcal{A}(\tilde{W})\right\|^2 - \left\langle b, \frac{1}{2u}b - \hat{y}\right\rangle = \\ &= \left\langle b, \hat{y} + \frac{1}{u}(\mathcal{A}\tilde{W} - b)\right\rangle + \frac{1}{2u}\left\|\mathcal{A}(\tilde{W})\right\|^2 - \left\langle b, \frac{1}{u}(\mathcal{A})\right\rangle + \frac{1}{2u}\|b\|^2 \\ &= \langle b, \tilde{y}\rangle + \frac{1}{2u}\left\|b - \mathcal{A}(\tilde{W})\right\|^2 = \langle b, \tilde{y}\rangle + \frac{u}{2}\|\tilde{y} - \hat{y}\|^2, \end{aligned} \tag{5.70}$$

where the last equation follows from (5.17) with $\tilde{y} = y_{\min}(\tilde{W})$. Thus, by (5.70), (5.68), and (5.5),

$$q_d(V, \alpha, \lambda) \geq \max\left\{\lambda_{\max}(P^T \left[C + \mathcal{A}^T\tilde{y}\right] P), \left\langle C - \mathcal{A}^T\tilde{y}, \overline{W}\right\rangle\right\} + \langle b, \tilde{y}\rangle + \frac{u}{2}\|\tilde{y} - \hat{y}\|^2.$$

Comparing this to (5.13) we see that (5.67) holds and therefore (5.66) is proved. Finally, the equivalence of $q_p(V, \alpha) = q_d(V, \alpha, \lambda)$ to optimality is due to the duality result of Lemma 5.2.2. ■

We solve (5.61) by a primal-dual interior point method (see Chapter 4). For a barrier parameter $\mu > 0$, the system reads

$$\begin{array}{rcrcrcrcrcrcl} F_U & := & Q_{11}\operatorname{svec} V & + & q_{12}\alpha & + & c_1 & + & \lambda s_I & - & \operatorname{svec}(U) & = & 0 \\ F_\beta & := & q_{12}^T\operatorname{svec} V & + & q_{22}\alpha & + & c_2 & + & \lambda & - & \beta & = & 0 \\ F_1 & := & -s_I^T\operatorname{svec} V & - & \alpha & + & 1 & & & & & = & 0 \\ & & & & & & & & & & UV & = & \mu I \\ & & & & & & & & & & \alpha\beta & = & \mu. \end{array}$$

A step direction $(\Delta\alpha, \Delta\beta, \Delta U, \Delta V, \Delta\lambda)$ may be determined via the linearized system

$$\begin{array}{rcrcrcrcrcl} Q_{11}\operatorname{svec} dV & + & q_{12}\Delta\alpha & - & \Delta\lambda s_I & - & \operatorname{svec}\Delta U & = & -F_U \\ q_{12}^T\operatorname{svec} dV & + & q_{22}\Delta\alpha & - & \Delta\lambda & - & \Delta\beta & = & -F_\beta \\ -s_I^T\operatorname{svec} dV & - & \Delta\alpha & & & & & = & -F_1 \\ (U \otimes_s V^{-1})\operatorname{svec} dV & & & & & + & \operatorname{svec}\Delta U & = & \mu\operatorname{svec} V^{-1} - \operatorname{svec} U \\ & & (\beta/\alpha)\Delta\alpha & & & + & \Delta\beta & = & \mu\alpha^{-1} - \beta \end{array}$$

In the current context we prefer the linearization $(U \otimes_s V^{-1})\operatorname{svec}\Delta V + \operatorname{svec}\Delta U$, because for this choice the system is easy to solve for ΔV with relatively little computational work per iteration. The final system for ΔV reads

$$\begin{aligned} &(Q_{11} + U \otimes_s V^{-1} + (\tfrac{\beta}{\alpha} + q_{22})s_I s_I^T - q_{12}s_I^T - s_I q_{12}^T)\operatorname{svec} dV = \\ &= \mu\ \operatorname{svec}(V^{-1}) - \operatorname{svec}(U) - F_U - F_1 q_{12} - (\mu\alpha^{-1} - \beta - \tfrac{\beta}{\alpha}F_1 - q_{22}F_1)s_I. \end{aligned} \tag{5.71}$$

It can be worked out that the system matrix is positive definite (because $U \otimes_s V^{-1} \succ 0$ by Proposition 1.3.3, 6, it suffices to show that $Q_{11} + q_{22}s_I s_I^T - q_{12}s_I^T - s_I q_{12}^T \succeq 0$ using $\begin{bmatrix} Q_{11} & q_{12} \\ q_{12} & q_{22} \end{bmatrix} \succeq 0$). The main work per iteration is the factorization of this matrix (with $V \in S_r$ this is $\binom{r+1}{2}^3/3$) and it is not possible to do much better, since Q_{11} has to be inverted at some point. Because of the strong dominance of the factorization it pays to employ a predictor corrector approach, but we will not delve into this here.

For $V \in S_r$ a strictly feasible primal starting point is

$$\begin{aligned} V^0 &= I/(r+1) \\ \alpha^0 &= 1/(r+1). \end{aligned}$$

A strictly feasible dual starting point can be constructed by choosing λ^0 sufficiently positive such that

$$U^0 = \operatorname{svec}^{-1}(Q_{11}\operatorname{svec}(V^0) + \alpha^0 q_{12} + c_1) + \lambda^0 I \succ 0 \tag{5.72}$$

$$\beta^0 = q_{12}^T \operatorname{svec} V + q_{22}\alpha + c_2 + \lambda^0 > 0. \tag{5.73}$$

By carefully choosing the barrier parameter and the step length in the line search, it is possible to set up a feasible primal-dual interior point algorithm that provably converges in polynomial time to a prespecified relative precision with respect to the gap in the starting point (see Kojima, Shindoh, and Hara [1997]). Here, we prefer to state our implementational choices. For the strictly feasible primal-dual starting pair we compute the first $\mu = (\langle U, V\rangle + \alpha\beta)/(r+1)$, compute the step direction $(\Delta\alpha, \Delta\beta, \Delta U, \Delta V, \Delta\lambda)$ as indicated above , perform a line search with line search parameter $0 < \delta \leq 1$ so that $(\alpha + \delta\Delta\alpha, \beta + \delta\Delta\beta, U + \delta\Delta U, V + \delta\Delta V, \lambda + \delta\Delta\lambda)$ is again strictly feasible, move to this new point, compute a new μ by

$$\mu = \min\left\{\mu_{\text{old}}, \gamma\frac{\langle U, V\rangle + \alpha\beta}{r+1}\right\} \text{ with } \gamma = \begin{cases} 1 & \text{if } \delta \leq \frac{1}{5} \\ \frac{5}{10} - \frac{4}{10}\delta^2 & \text{if } \delta > \frac{1}{5}, \end{cases}$$

and iterate.

The choice of the stopping criterion requires some care. Observe that all iterates of the interior point algorithm satisfy the requirements of Lemma 5.5.2 and thus (5.66) is valid throughout. Denote the optimal value of the augmented problem by

$$l_* = \max_{W \in \widehat{\mathcal{W}}^k} L^k(y(W), W) \tag{5.74}$$

In computing the solution of (QSP) we should guarantee two properties. First, the accuracy has to be reasonable with respect to the gap between $f(\hat{y}) - l_*$ (which is nonnegative by (5.27)). We can model this by requiring that $q_d - q_p \leq \varepsilon_q(f(\hat{y}) - q_p)$ for some appropriately chosen $\varepsilon_q \in (0,1)$. And second, for the proof of convergence it is of vital importance that (5.30) and (5.31) hold after a null step. Verifying these two conditions directly is computationally too expensive. However, with

$$l_{old} = \begin{cases} -\infty & \text{if } \hat{y}^k = y^k \\ L(y^k, W^k) & \text{otherwise,} \end{cases} \tag{5.75}$$

testing for $q_d - q_p \leq \frac{1}{2}(q_d - l_{old})$ will have a similar effect.

Lemma 5.5.3 *Given the same assumptions and definitions as in Lemma 5.5.2. If $\hat{y}^k \neq y^k$ and*

$$q_d^k(V, \alpha, \lambda) - q_p^k(V, \alpha) \leq \tfrac{1}{2}\left[q_d^k(V, \alpha, \lambda) - L^k(y^k, W^k)\right], \tag{5.76}$$

then $\tilde{W} = P_k V P_k^T + \alpha\overline{W}_k$ and $\tilde{y} = y_{\min}^k(\tilde{W})$ satisfy

$$L^k(y^k, W^k) + \frac{u}{4}\left\|\tilde{y} - y^k\right\|^2 \leq L^k(\tilde{y}, \tilde{W}) \tag{5.77}$$

and

$$f_{W_S^k}(\tilde{y}) \leq f_{\bar{W}}(\tilde{y}) + L^k(\tilde{y}, \tilde{W}) - L^k(y^k, W^k). \tag{5.78}$$

Proof. Let $q_p = q_p^k(V, \alpha)$ and $q_d = q_d^k(V, \alpha, \lambda)$. We first prove (5.77). By (5.26), $W^k \in \widehat{\mathcal{W}}^k$, and (5.66),

$$L^k(y^k, W^k) + \frac{u}{2}\left\|\tilde{y} - y^k\right\|^2 = L^k(\tilde{y}, W^k) \leq \max_{W \in \widehat{\mathcal{W}}^k} L^k(\tilde{y}, W) \leq q_d.$$

Condition (5.76) implies $q_p \geq \frac{1}{2}[q_d + L^k(y^k, W^k)]$ and so (5.77) follows from (5.66) and

$$2L^k(y^k, W^k) + \frac{u}{2}\left\|\tilde{y} - y^k\right\|^2 \leq q_d + L^k(y^k, W^k) \leq 2q_p = 2L^k(\tilde{y}, \tilde{W}).$$

Now we turn to (5.78). Observe that by (5.66), (5.14), and $W_S^k \in \widehat{\mathcal{W}}^k$,

$$q_d - q_p \geq \max_{W \in \widehat{\mathcal{W}}^k} L^k(\tilde{y}, W) - L^k(\tilde{y}, \tilde{W}) = \max_{W \in \widehat{\mathcal{W}}^k} f_W(\tilde{y}) - f_{\tilde{W}}(\tilde{y}) \geq f_{W_S^k}(\tilde{y}) - f_{\tilde{W}}(\tilde{y}).$$

Since, by (5.76) and (5.66), $q_d - q_p \leq q_p - L^k(y^k, W^k) = L^k(\tilde{y}, \tilde{W}) - L^k(y^k, W^k)$, this proves (5.78). ■

The next lemma proves that appropriate (V, α, λ) can be found in finite (for interior point methods, polynomial) time if $f(\hat{y}^k) > l_*^k$.

Lemma 5.5.4 *Let $\varepsilon_q \in (0,1)$, l_*^k and l_{old}^k be as defined in* (5.74) *and* (5.75). *Suppose that in step 2 of Algorithm 5.2.4 we use a routine that generates a sequence $(V^h, \alpha^h, \lambda^h)$, feasible for* (5.61) *and* (5.63), *with $q_p^k(V^h, \alpha^h) \to l_*^k$ and $q_d^k(V^h, \alpha^h, \lambda^h) \to l_*^k$, where q_p^k and q_d^k are as defined in* (5.64) *and* (5.65). *Then there is a finite index h with*

$$q_d^k(V^h, \alpha^h, \lambda^h) - q_p^k(V^h, \alpha^h) \leq \min\left\{\varepsilon_q[f(\hat{y}^k) - q_p^k(V^h, \alpha^h)], \tfrac{1}{2}\left[q_d^k(V^h, \alpha^h, \lambda^h) - l_{old}^k\right]\right\}, \quad (5.79)$$

unless $f(\hat{y}^k) = l_*^k$. In the latter case $\hat{y}^k \in \operatorname{Argmin}_{y \in Y} f(y)$.

Proof. In order to simplify notation, we fix k and drop this iteration index where possible. Denote by $q_p^h = q_p^k(V^h, \alpha^h)$ and $q_d^h = q_d^k(V^h, \alpha^h, \lambda^h)$. By (5.66), $q_p^h \leq l_*$ and $q_d^h \geq l_*$. We first show

$$l_{old} < l_*. \quad (5.80)$$

For $y^k = \hat{y}^k$ it holds by definition of l_{old} (cf. (5.75)). For $y^k \neq \hat{y}^k$ this follows from $l_{old} = L^k(y^k, W^k)$ (by definition of l_{old}), (5.30) and $L^k(y^k, W_S^k) > L(y^k, W^k)$. To see the latter, observe that $L^k(y^k, W_S^k) \leq L(y^k, W^k)$ would have resulted in a descent step in iteration $k-1$ by step 5 of Algorithm 5.2.4 and the definition of L^k (5.14).

Now we investigate the case $f(\hat{y}) > l_*$. First, $\varepsilon_q[f(\hat{y}^k) - q_p^h] \geq \varepsilon_q[f(\hat{y}^k) - l_*] =: \varepsilon_1 > 0$, because $q_p^h \leq l_*$. Next, $\frac{1}{2}[q_d^h - l_{old}] > \frac{1}{2}[l_* - l_{old}^k] =: \varepsilon_2 > 0$, because $q_d^h \geq l_*$. Since $q_d^h - q_p^h \to 0$, there is an index $\hat{h}$ so that $q_d^h - q_p^h < \min\{\varepsilon_1, \varepsilon_2\}$.

Finally, consider the case $f(\hat{y}) = l_*$. We first show optimality. If $l_* = f(\hat{y}^k)$ then for $W_* \in \operatorname{Argmax}_{W \in \widehat{\mathcal{W}}^k} L(y_{\min}^k(W), W)$ (5.26) and (5.27) yield $y_{\min}^k = \hat{y}^k$ and by (5.17) we conclude $\nabla f_{W^*} = 0$. We proceed to the failure of the stopping criterion. Since $q_d^h \geq l_*$ we have $q_d^h - q_p^h \geq f(\hat{y}^k) - q_p^h$ and the first term of the min in (5.79) cannot be satisfied for any h. In order to satisfy the second term we need $\frac{1}{2}(q_d^h - l_{old}) \geq q_d^h - q_p^h$ but because $q_d^h \geq l_* = f(\hat{y}^k)$, we obtain $q_d^h - q_p^h > \varepsilon_q[f(y^k) - q_p^h]$ and in this case the second term is never satisfied if the first is not. ■

Of course, the infinite loop in the case $f(\hat{y}^k) = l_*^k$ could be avoided by some theoretic construction, but practical implementations of such methods include several other termination criteria that take care of this situation as well, *e.g.*, they terminate when some maximum precision is reached or a certain iteration limit is exceeded.

Theorem 5.5.5 *Assume that in step 2 of Algorithm 5.2.4 we use a method for computing W^{k+1} that satisfies the assumptions of Lemma 5.5.4 and uses the stopping criterion* (5.79). *Then Theorem 5.3.6 holds.*

Proof. We only have to verify (5.28), all other results follow from (5.28) as before. Eq. (5.28) depends on (5.30) and (5.31). Replacing these two by (5.77) and (5.78), respectively, the proof of (5.28) can be completed by the same steps as before. ■

Remark 5.5.6

Note that the termination criterion (5.79) *provides an efficient automatism for ensuring the required precision in the model computation. If the cutting plane model still changes a lot then the precise optimizer is not needed, a rough guess is sufficient. This strategy may significantly reduce the time spent in the interior point code.*

In practice, the interior point method turns out to be numerically robust and allows to compute the solution almost to machine precision. Due to the update rule for P_k (5.23) *(and the small number of columns in comparison to the number needed) the minimal eigenvalue of* V *is typically well bounded away from zero in the optimal solution. Then the inversion of* V *and the factorization in solving* (5.71) *involve positive definite matrices with bounded condition number and are therefore numerically stable.*

Inequality Constraints and Inner Iterations

The Lagrange multipliers $\hat{\eta}$ of the sign constraints on y are easily included in the interior point method above. Indeed, by replacing b with $b-\hat{\eta}$, (QSP) of Lemma 5.2.2 allows to compute ($\text{QSP}_{\hat{\eta}}$) of Lemma 5.4.1. It is therefore not surprising that all results on computing (QSP) can be extended to ($\text{QSP}_{\hat{\eta}}$).

The inner iterations in step 2 of Algorithm 5.4.4 offer further possibilities for increasing efficiency. Note that repeated inner iterations do not affect P_k but only $\hat{\eta}$. The changes in $\hat{\eta}$ influence the linear (but not the quadratic!) coefficients c_1, c_2, and d of (5.62), because they contain b. They can be updated efficiently. If the inner iteration yields rather small changes in $\hat{\eta}$, then the optimal solution will also change only slightly. This can be exploited in restarting strategies.

A restart procedure for reoptimizing (5.61) in the inner loop might do the following. Let $(\hat{V}, \hat{\alpha}, \hat{U}, \hat{\beta}, \hat{\lambda})$ be the solution of the latest subproblem, and let Δc_1 and Δc_2 be the increments of c_1 and c_2 due to the change in $\hat{\eta}$. Choose a starting point $(V^0, \alpha^0, U^0, \beta^0, \lambda^0)$ as follows. Since $(V, \alpha) = \frac{1}{r+1}(I, 1)$ is the analytic center of the primal feasible set, any point on the line segment $\xi(\hat{V}, \hat{\alpha}) + (1-\xi)\frac{1}{r+1}(I, 1)$ with $\xi \in [0, 1]$ is primal feasible. Moreover, it should be close to the central path if the new solution is relatively close to the old one. We choose ξ by projecting $(1 - \frac{\|(\Delta c_1, \Delta c_2)\|}{\|(c_1, c_2)\|})^{1/2}$ on $[0.9, 0.99999]$. The change from $(\hat{V}, \hat{\alpha})$ to (V^0, α^0) in (5.72) and (5.73) determines the changes in the dual variables up to a diagonal shift that can be applied through λ. This shift is chosen so that the changes in U and β are positive definite. Thus, U^0 (β^0) is positive definite, but not too far from $\hat{U}$ ($\hat{\beta}$). Numerical experiments reveal that this restarting heuristic reduces the number of iterations by one third up to one half, depending on the size of the changes in $\hat{\eta}$.

5.6 Eigenvalue Computation and Inexact Evaluation

Assume that the structure of a symmetric matrix $A \in S^n$ (the matrix we have in mind is $Z = C - \mathcal{A}^T y$) is such that the product of the matrix with a vector can be computed quickly. As of today, the best approach for computing a few extremal eigenvalues of such a matrix is the Lanczos method. A detailed description of this method is beyond the scope of this text, we will outline only the main ideas (the reader is referred to Golub and van Loan [1989] for an introduction and to Saad [1992]; Parlett [1998] for an in-depth treatment of Lanczos methods).

We start with a much simpler method, the *power* method. For a given (normalized) starting vector q_1 it generates in step i the vector $v_i = Av_{i-1} = A^{i-1}q_1$ ($i = 1, 2, \ldots$, and $A^0 = I$). If $|\lambda_{\max}(A)| > |\lambda_{\min}(A)|$ and q_1 has some component in the eigenspace of the maximum eigenvalue of A (a random vector from $\mathbb{R}^n$ satisfies this requirement with probability one), then the sequence of normalized vectors $v_i / \|v_i\|$ has an accumulation point in the eigenspace of A. To see this, let $A = P\Lambda P^T$ denote an eigenvalue decomposition of A with $P^T P = I$ and $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$, $\lambda_{\max}(A) = \lambda_1 = \cdots \lambda_h > \lambda_{h+1} \geq \cdots \geq \lambda_n$. Then $A^i = P\Lambda^i P^T$ and since $\lambda_{\max}(A)^i$ grows faster

than any other eigenvalue of A, any component of q_1 in the direction of the maximal eigenspace will dominate in v_i eventually.

In contrast to the power method, the Lanczos method produces in iteration i ($i = 1, \ldots, n$) the normalized vector v_i that maximizes $v^T Av / \|v\|^2$ for vectors v in the subspace spanned by the vectors $A^{j-1}q_1$ for $j = 1, \ldots, i$. This subspace is called *Krylov* subspace and is denoted by

$$\mathcal{K}(A, q_1, i) = \operatorname{span}\left\{q_1, Aq_1, \ldots, A^{i-1}q_1\right\}.$$

In order to find this best vector efficiently, the Lanczos method constructs a special orthonormal basis $\mathcal{Q}_k = \{q_1, \ldots, q_i\}$ of $\mathcal{K}(A, q_1, i)$ as follows. Let $Q_i = [q_1, \ldots, q_i]$. Conceptually, the basis vector q_{i+1} is computed by orthonormalizing Aq_i with respect to all previous vectors,

$$q_{i+1} = \frac{\bar{q}_{i+1}}{\|q_{i+1}\|} \quad \text{with} \quad \bar{q}_{i+1} = Aq_i - Q_i Q_i^T Aq_i \in \mathcal{K}(A, q_1, i+1). \tag{5.81}$$

In fact, for $i \geq 3$, the projection step $Q_i Q_i^T Aq_i$ of the orthogonalization involves only the two vectors q_i and q_{i-1}: q_i is in the orthogonal complement of $\mathcal{K}(A, q_1, i-1)$ which contains Aq_j^T for $j = 1, \ldots, i-2$, hence $q_j^T Aq_i = 0$. For the two remaining terms we set $\alpha_i = q_i^T Aq_i$ and $\beta_{i-1} = q_{i-1}^T Aq_i$. The β_i may be computed by multiplying the right equation of (5.81) from the right by q_{i+1}^T. Since $q_{i+1}^T Q_i = 0$ this yields $\beta_i = q_{i+1}^T Aq_i = q_{i+1}^T \bar{q}_{i+1} = \|\bar{q}_{i+1}\|$. Thus, the vectors q_i, called *Lanczos vectors*, can be computed by the following algorithm.

Algorithm 5.6.1
Input: *the starting vector q_1, a routine for computing Av for some $v \in \mathbb{R}^n$.*

1. *Set $\beta_0 = 1$, $q_0 = 0$, $i = 1$.*
2. *Compute $w = Aq_i$.*
3. *Set $\alpha_i = w^T q_i$, $\bar{q}_{i+1} = w - \alpha_i q_i - \beta_{i-1} q_{i-1}$, $\beta_i = \|\bar{q}_{i+1}\|$.*
4. *If $\beta_i = 0$ then **stop**.*
5. *Set $q_{i+1} = \bar{q}_{i+1}/\beta_i$, increase i by 1, and **goto** 2.*

The projection of A onto the Krylov subspace $\mathcal{K}(A, q_1, i)$ gives rise to the tridiagonal matrix

$$T_i = Q_i^T A Q_i = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & \cdots & 0 \\ \beta_1 & \alpha_2 & \beta_2 & \ddots & \vdots \\ 0 & \beta_2 & \alpha_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \beta_{i-1} \\ 0 & \cdots & 0 & \beta_{i-1} & \alpha_i \end{bmatrix} \in S_i.$$

For this tridiagonal matrix an eigenvalue decomposition $T_i = \bar{Q}_i \Lambda_i \bar{Q}_i$ can be computed in $O(i^2)$ by the Symmetric QR Algorithm (see Golub and van Loan [1989], Algorithm 8.2.2). The eigenvector $\bar{v}$ of $\lambda_{\max}(T_i)$ yields the vector $v_i = Q_i \bar{v}$ maximizing $v^T Av / \|v\|^2$ for $v \in \mathcal{K}(A, q_1, i)$.

In order to study the convergence properties of this value, observe that $v \in \mathcal{K}(A, q_1, i)$ is equivalent to $v = \sum_{j=1}^{i} \pi_j A^{j-1} q_1$ for $\pi_j \in \mathbb{R}$, $j = 1, \ldots, i$, in other words, there is a polynomial $p(\cdot)$ of degree $i-1$ so that $v = p(A)q_1$. With $A = P\Lambda P^T$ as above, we may write $p(A) = Pp(\Lambda)P^T$. Hence, the vector v_i corresponds to a polynomial of degree $i-1$ that maximizes

$$\frac{v^T T_i v}{v^T v} = \frac{q_1^T p(A) A p(A) q_1}{q_1^T p(A)^2 q_1} = \frac{q_1^T P p(\Lambda)^2 \Lambda P^T q_1}{q_1^T P p(\Lambda)^2 P^T q_1} = \frac{\sum_{j=1}^{n} \lambda_j p(\lambda_j)^2 (P_{\cdot,j}^T q_1)^2}{\sum_{j=1}^{n} p(\lambda_j)^2 (P_{\cdot,j}^T q_1)^2}.$$

This expression equals the maximum eigenvalue, if the coefficients $p(\lambda_j)^2 (P_{\cdot,j}^T q_1)^2$ of λ_j are positive for some of the $\lambda_{\max}(A) = \lambda_1 = \cdots = \lambda_h$ and zero for $\lambda_{h+1}, \ldots, \lambda_n$. Therefore, a starting vector

q_1 whose component within the maximal eigenspace is large will help to speed up convergence. Furthermore, a polynomial whose value is large in $\lambda_{\max}(A) = \lambda_1 = \cdots = \lambda_h$ but small on the interval $[\lambda_n, \lambda_{h+1}]$ will be available for small i if the spread $\lambda_{h+1} - \lambda_n$ is small in comparison to the separation $\lambda_{\max}(A) - \lambda_{h+1}$, *i.e.*, a few matrix vector multiplications will suffice to produce reasonable approximations to $\lambda_{\max}(A)$ and its eigenvector.

After at most n matrix vector multiplications the Krylov subspace spans an invariant subspace (further multiplications with A will not enlarge it). If the starting vector q_1 has some component in the maximal eigenspace of A, then, in theory, the method stops with $\lambda_{\max}(A)$ after at most n iterations. In practice, however, the number of columns of Q_i has to be kept small for reasons of computational efficiency. This is usually achieved by restarting the process after a certain number of iterations. Various strategies for restarting are available, *e.g.*, one might use the vector $Q_i\bar{v}$, where $\bar{v}$ is the eigenvector to $\lambda_{\max}(T_i)$, as a new starting vector (Saad [1992]) or employ the implicit restarting approach of Sorensen [1992]. The restriction of the number of columns of Q together with numerical difficulties due to generic cancellations appearing in the Lanczos algorithm may lead to iteration numbers much larger than n if the maximum eigenvalue is not well separated.

In the first few iterations of the bundle algorithm the maximum eigenvalue is in general well separated and the maximum eigenvector of the previous iterate is a reasonable starting vector. As the algorithm proceeds, more and more eigenvalues cluster at $\lambda_{\max}(A)$. This clustering is a generic property of optimal solutions in semidefinite programming which is due to the facial structure of the semidefinite feasible set described in Lemma 2.3.1. In consequence, it gets more and more difficult to compute good approximations via the Lanczos method. At this stage the eigenvalue routine often is the bottleneck of the algorithm.

The situation can be improved somewhat by stopping early in the case that the current estimate $v_i^T T_i v_i$ ensures a null step in Step 5 of Algorithm 5.2.4. Essentially, the Lanczos method produces successively better lower estimates

$$\lambda_{\max}(T_i) \leq \lambda_{\max}(A) = \lambda_{\max}(C - \mathcal{A}^T y)$$

for the maximum eigenvalue and therefore for the true function value $\lambda_{\max}(C - \mathcal{A}^T y) + b^T y$. If the current estimate $\lambda_{\max}(T_i)$ is already good enough to prove that the current y will result in a null step, then the corresponding (normalized) eigenvector v of $\lambda_{\max}(T_i)$ gives rise to a cutting plane via $W_S^{k+1} = vv^T$ that improves the cutting plane model sufficiently. This may lead to considerable speed-up, because in iterative methods eigenvalues may converge significantly faster than eigenvectors, see del Corso [1997]. The true eigenvectors of A are only needed if we have to prove that $\lambda_{\max}(T_i)$ is indeed close to an eigenvalue of A. We provide a rigorous analysis of this inexact evaluation approach for Algorithm 5.2.4, but the analysis can be extended to Algorithm 5.4.4 by the same arguments.

For some parameter $\bar{\kappa} \in (\kappa, 1)$ suppose that in Algorithm 5.2.4 steps 4 and 5 are replaced by

4'. (Descent test). Find $W_S^{k+1} \in \mathcal{W}$ such that either

(a) $f(\hat{y}^k) - f_{W_S^{k+1}}(y^{k+1}) \leq \bar{\kappa}\left[f(\hat{y}^k) - f_{W^{k+1}}(y^{k+1})\right]$, or

(b) $f_{W_S^{k+1}}(y^{k+1}) = f(y^{k+1})$ and $f(\hat{y}^k) - f(y^{k+1}) \geq \kappa\left[f(\hat{y}^k) - f_{W^{k+1}}(y^{k+1})\right]$.

In case (a), set $\hat{y}^{k+1} = \hat{y}^k$ (null step), otherwise set $\hat{y}^{k+1} = y^{k+1}$ (descent step).

Observe that the threshold for accepting null steps in (a) is lower than the threshold for rejecting serious steps in (b). The purpose of this interval is to ensure finite termination in the eigenvalue routine.

Lemma 5.6.2 *Let $\lambda = \lambda_{\max}(C - \mathcal{A}^T y)$ and suppose that the eigenvalue routine produces sequences $v_i^T(C - \mathcal{A}^T y)v_i = \underline{\lambda}_i \to \lambda$ and $\overline{\lambda}_i \to \lambda$ with $\underline{\lambda}_i \leq \lambda \leq \overline{\lambda}_i$ for $i = 1, 2, \ldots$. Suppose further that for $W_S^{k+1} = v_i v_i^T$ the condition $f_{W_S^{k+1}} = f(y^{k+1})$ is considered true if, for some parameter $\varepsilon_\lambda \in (0,1)$, the gap $\overline{\lambda}_i - \underline{\lambda}_i \leq \varepsilon_\lambda(\overline{\kappa} - \kappa)[f(\hat{y}^k) - f_{W^{k+1}}(y^{k+1})]$. Then there is a finite $\hat{\imath}$ so that $W_S^{k+1} = v_i v_i^T$ satisfies (a) or (b) of step 4'.*

Proof. Let $\underline{\mu} = f(\hat{y}^k) - \overline{\kappa}\left[f(\hat{y}^k) - f_{W^{k+1}}(y^{k+1})\right] - b^T y^{k+1}$ denote the lower bound on the eigenvalue estimate for being accepted as null step and let $\overline{\mu} = f(\hat{y}^k) - \kappa\left[f(\hat{y}^k) - f_{W^{k+1}}(y^{k+1})\right] - b^T y^{k+1}$ denote the upper bound, above which no descent steps are accepted. Then step 4' (a) is satisfied if $\underline{\lambda}_i \geq \underline{\mu}$ and 4' (b) is satisfied if $\overline{\lambda}_i - \underline{\lambda}_i \leq \varepsilon_\lambda(\overline{\mu} - \underline{\mu})$ and $\overline{\lambda}_i \leq \overline{\mu}$. Because of the stopping criterion in step 3 of Algorithm 5.2.4 we obtain $f(\hat{y}^k) > f_{W^{k+1}}(y^{k+1})$ and therefore $\overline{\mu} - \underline{\mu} > 0$. Since $\overline{\lambda}_i - \underline{\lambda}_i \to 0$ by assumption, there exists an $\hat{\imath}$ so that $\overline{\lambda}_i - \underline{\lambda}_i \leq \varepsilon_\lambda(\overline{\mu} - \underline{\mu})$. Furthermore $\underline{\lambda}_i \geq \underline{\mu}$ or $\overline{\lambda}_i \leq \overline{\mu}$, because otherwise $\overline{\lambda}_i - \underline{\lambda}_i > \overline{\mu} - \underline{\mu}$ yields a contradiction to the choice of $\hat{\imath}$. ■

In the case of a descent step one has to set $f(y^{k+1}) := \overline{\lambda}_i + b^T y^{k+1}$ (instead of the value $f_{W_S^{k+1}}(y^{k+1})$ generated by $W_S^{k+1} = v_i v_i$) in order to ensure correctness of the algorithm. If 4' (a) and 4' (b) are satisfied simultaneously, the descent step is selected.

Unfortunately, the Lanczos method itself does not produce an upper bound $\bar{\lambda}_i$. Such a bound could, in theory, be guessed and verified by Cholesky decomposition, but for practical applications this is computationally too expensive. The finite termination property for this class of eigenvalue routines should not hide the main purpose of the approach, namely to save time by stopping eigenvalue computations early in the case of null steps.

Turning to the proof of convergence we note that Proposition 5.3.1 and Lemma 5.3.2 are not affected by the changes, since $f(\hat{y}^k)$ is the result of a descent step.

For Lemma 5.3.3 the proof remains valid till (5.32) with (5.32) included. To complete the proof assume that $L^K(y^k, W_S^k) < \max_{W \in \mathcal{W}} L^K(y^k, W)$ for infinitely many k. We will show that in this case $\hat{y} \in \operatorname{Argmin} f$ (and therefore also $\hat{y} \in \operatorname{argmin} f(y) + \frac{u}{2}\|y - \hat{y}\|^2$) with $y^k \to \hat{y}$. By step 4'(a) and Lemma 5.3.2,

$$f_{W_S^k}(y^k) - f_{W^k}(y^k) \geq (1 - \bar{\kappa})[f(\hat{y}) - f_{W^k}(y^k)] \geq 0$$

is true for an infinite number of steps. Thus, the proof can completed as in Lemma 5.3.4. This proves (5.29) for inexact evaluation.

The proof of Lemma 5.3.4 remains unchanged except for replacing κ by $\bar{\kappa}$. With respect to Lemma 5.3.5 we note that (5.35) is valid in the new setting as well, because $f_{W_S^{k+1}}(y^{k+1}) = f(\hat{y}^{k+1})$ in the case of a descent step $y^{k+1} = \hat{y}^{k+1}$, and descent steps satisfy $f(\hat{y}^k) - f(y^{k+1}) \geq \kappa[f(\hat{y}^k) - f_{W^{k+1}}(y^{k+1})]$ as before. Therefore no changes are required in the proof of the lemma.

We conclude that Theorem 5.3.6 remains valid for inexact evaluation, with steps 4 and 5 of Algorithm 5.2.4 replaced by step 4'.

Theorem 5.6.3 *For Algorithm 5.2.4 with steps 4 and 5 replaced by step 4' the following holds. Either* $\hat{y}^k \to \bar{y} \in \operatorname{Argmin}_{y \in Y} f(y)$, *or* $\operatorname{Argmin}_{y \in Y} f(y) = \emptyset$ *and* $\|\bar{y}\| \to \infty$. *In both cases* $f(\hat{y}^k) \downarrow \inf_{y \in Y} f$.

The Lanczos method offers further useful information. Typically, all of the large eigenvalues of T_i are close to large eigenvalues of A. Thus, in addition to the eigenvector to $\lambda_{\max}(T_i)$ the eigenvectors to the other large eigenvalues of T_i can also be used to construct good cutting planes for the model at no extra cost.

5.7 Remarks on the Literature

Lewis and Overton [1996] is an excellent survey on eigenvalue optimization. The general reference for convex optimization with subgradient methods is Hiriart-Urruty and Lemaréchal [1993a], Hiriart-Urruty and Lemaréchal [1993b].

The spectral bundle method builds on the framework of the proximal bundle method of Kiwiel [1990] which employs aggregation of linear cutting planes to a linear aggregate cutting plane as suggested by Kiwiel [1983]. The bundle trust method of Schramm and Zowe [1992] is a related but different general convex minimization method in that it uses a trust region approach instead of the weight u.

The spectral bundle method was introduced in Helmberg and Rendl [2000] and marks the first use of a nonpolyhedral cutting plane model in bundle methods. This also applies to the conic

aggregation step, that makes full use of the nonpolyhedral structure of the cutting plane model. The spectral bundle method was extended to include bounds and inexact evaluation by Helmberg and Kiwiel [1999]. A similar simple updating scheme for Lagrange multipliers was, to the best of our knowledge, not in use in bundle methods before. Kiwiel [1995] suggested inexact function evaluation and subgradient computations; the criterion used here differs slightly from Kiwiel [1995] and exploits the special structure of the problem. The exposition of the chapter follows Helmberg and Kiwiel [1999] and profits from connected discussions with K. C. Kiwiel. In particular, the inexact stopping criterion for the subproblem of Lemma 5.5.3 ensuring finite convergence of the interior point method and the associated proof of convergence of Lemma 5.5.4 is published here for the first time.

Specialized routines for eigenvalue optimization first appeared for a restricted class of problems in Cullum, Donath, and Wolfe [1975]. This was later generalized by Polak and Wardi [1982] and, in more implementable form, by Kiwiel [1986]. These 'classical' algorithms require in each iteration the computation of all eigenvectors to eigenvalues within an ε-distance of the maximum eigenvalue and use these to construct an enlargement of the subdifferential. Close to the optimal solution the number of eigenvalues is at least as large as the multiplicity of the maximum eigenvalue in the optimal solution. The approximation to the element of minimal norm within these enlargements is again computed by a subgradient method.

Inspired by the work of Fletcher [1985], Overton [1992] developed a quadratically convergent method. Each step is computed from a complete spectral decomposition of the matrix and a guess of the exact multiplicity of the maximum eigenvalue in the optimal solution. The information is used to construct a tangential approximation to the smooth manifold of matrices having the same multiplicity of the maximum eigenvalue and employs a quadratic model within this affine subspace. The final system is closely related to the system for determining the step direction in primal-dual algorithms. The analysis of the method was completed independently by Overton and Womersly [1995] and Shapiro and Fan [1995].

In recent work, Oustry [1999a] reinterprets the algorithm of Overton within the framework of the $\mathcal{U}$-Lagrangian introduced by Lemaréchal, Oustry, and Sagastizábal [2000]. In these terms, the smooth manifold corresponds to the ridge of the convex eigenvalue function and the second order method uses a quadratic model of the ridge along the tangent direction of the ridge in the current point. Oustry [1999b] embeds the second order method in a first order method to ensure global convergence. Again, for global convergence the approach relies on the spectrum of all eigenvalues within ε-distance of the maximum eigenvalue to construct an enlargement of the subdifferential and makes use of the entire spectral information to obtain local quadratic convergence. As in the classical methods, the element of minimal norm within the enlargement of the subdifferential is computed by a subgradient method and a polyhedral cutting plane model is used. Helmberg and Oustry [2000] present an overview over these methods and discuss some possibilities for combining the approaches.

Chapter 6

Cutting Plane Algorithms in Semidefinite Programming

In linear programming, cutting plane methods belong to the most successful techniques for solving combinatorial optimization problems. Given a linear cost function and a set of feasible integral points, one computes an optimal solution of a linear relaxation of the convex hull of this set. If the solution is not feasible for the convex hull, the separation problem must be solved, *i.e.*, find inequalities that are valid for the convex hull, but cut off the current point. Upon success, the process is iterated for the tightened relaxation. The approach requires a profound understanding of the relevant polyhedral structure and hinges on the availability of good separation routines as well as a method that is able to solve a sequence of related linear relaxations efficiently. There is a vast literature on cutting plane methods; a bibliographic overview on implementational approaches is presented by Caprara and Fischetti [1997], for polyhedral studies see Aardal and Weismantel [1997]. Linear programs arising from combinatorial applications may be extremely large scale, having up to several millions of variables. For some problems cutting planes are hard to find, for others there are way too many to include all of them. In the second case one is faced with the problem of selecting a "good" subset. The latter problem is not well understood and it seems likely that there is no satisfactory answer. In the design of cutting plane algorithms algorithmic decisions are therefore often based on heuristic considerations involving the problem at hand. These decisions may work well in one case but may fail for a slight modification of the problem data.

In semidefinite programming, cutting plane algorithms are still a rare sight, even though theory indicates superior quality of semidefinite over linear relaxations for several problems of practical interest. One reason for this lack of interest is, that for the main tool for solving semidefinite relaxations, interior point methods, it is neither known how to make efficient use of structural properties of a given problem, nor is there a convincing warm-start strategy that allows to re-optimize a slightly changed problem quickly. Therefore, current computational studies based on interior point methods are restricted to small problems of moderate practical relevance.

With the development of the spectral bundle method an algorithm is available that is capable of exploiting structural properties of given problem data. Furthermore, it offers attractive possibilities for warm-starts. Even though the final convergence rate of the spectral bundle method is poor, its strong initial convergence may render it an attractive choice for problems that do not require solutions of high accuracy.

In this chapter we discuss, on basis of particular case studies, some possibilities for implementing cutting plane algorithms for semidefinite relaxations with interior point methods (Section 6.1) and the spectral bundle method (Section 6.2). In both cases, the underlying methods are still under development and computational experience is limited. Cutting plane methods in semidefinite programming are faced with the same difficulties as in linear programming concerning algorithmic decisions such as the separation and selection of cutting planes. It is therefore impossible to ar-

rive at definite conclusions, but we try to locate potential advantages and disadvantages of these approaches. In addition, we present computational results[1] in order to illustrate the practical behavior of our algorithms as well as the quality of some of the semidefinite relaxations described in Chapter 3.

On basis of sensitivity analysis of the optimal solution of a relaxation it is sometimes possible to determine the correct value of an integral variable in all optimal integral solutions. This technique, called *fixing* of variables, is of high importance in branch and cut approaches. In semidefinite relaxations some bounds on variables are implicitly contained in the semidefiniteness constraint. The sensitivity analysis has to recover the dual information corresponding to these bounds from the dual semidefinite matrix variable. This is the topic of Section 6.3.

6.1 An Interior Point Approach

In developing an interior point cutting plane algorithm numerous algorithmic decisions have to be made. The most prominent are

- the choice of an initial relaxation of the combinatorial optimization problem,
- the cutting planes to separate and the separation algorithms to use,
- the selection of a subset of inequalities if too many violated inequalities have been found,
- the warm-start strategy after the addition of inequalities,
- the elimination of inequalities that are not tight,
- and the selection of an interior point method.

These decisions are typically problem dependent, they strongly influence each other, and it is often impossible to justify them by theoretical arguments. None the less the decisions have to be made and we hope that sharing our experience on two particular problems will help to facilitate these decisions in future applications. The algorithmic approaches should not be understood as recommendations, but rather as a discussion of alternatives that may be helpful in one case but may fail in another. The two algorithms are the cutting plane algorithm for max-cut as described by Helmberg and Rendl [1998] and the cutting plane algorithm for the quadratic 0-1 knapsack problem of Helmberg, Rendl, and Weismantel [2000], see also Helmberg and Weismantel [1998]. The two combinatorial problems are closely related, because the quadratic 0-1 knapsack problem can be interpreted as the simplest case of a constrained max-cut problem, *i.e.*, max-cut with one linear constraint, see Section 3.2. Yet the underlying polyhedra differ substantially: the structure of the max-cut polytope is determined by the dimension only, whereas the quadratic 0-1 knapsack polyhedron depends, in addition, on the data of the linear constraint. For our algorithms this rendered several successful approaches for max-cut useless for quadratic 0-1 programming.

Since both algorithms use the same interior point code, we discuss this choice and some of its consequences in advance. Then we will turn to the cutting plane algorithm for max-cut and report on our choices with respect to the issues listed above. Finally, we will do the same for the quadratic 0-1 knapsack problem.

The Interior Point Method

Both algorithms use the same feasible primal-dual interior point method employing the XZ-step direction (4.6) that is implemented as described in Section 4.4. A primal-dual method was selected because at that time numerical experiments indicated better convergence properties of the primal-dual approach in connection with inequalities. Due to the special structure of the max-cut and the quadratic 0-1 programming problems the use of feasible methods is possible. Since, in addition, the

[1] All times are computed on a Sun Ultra 10 with a 299 MHz SUNW,UltraSPARC-IIi CPU.

currently best bounds on the number of iterations are known for feasible path-following methods the choice of a feasible method seems natural.

The convergence theory of feasible primal-dual path-following interior point methods requires the iterates to remain in "the neighborhood" of the central path (cf. (4.11)), hence the name *path-following*. The central path is a smooth curve from the analytic center of the feasible set to the analytic center of the optimal face of the feasible set. The neighborhood is a wide region in the vicinity of the analytic center and shrinks along the central path to a single point in the optimal solution. In cutting plane algorithms it is of fundamental importance that the basic operation, adding a few cutting planes and reoptimizing, can be carried out in reasonable time. A new inequality, cutting of an almost optimal iterate X_c close to the central path, causes the central path to "jump" to a new location, because the optimal face changes its place. This renders X_c a poor starting point; it is both infeasible and far away from the new central path. As we will explain below, we were, to some extent, able to construct a reasonable point for restarting in max-cut, but did not succeed for the quadratic 0-1 knapsack. In the latter case we had to recompute the relaxation from scratch. In view of the fact that this required typically only about 30 iterations of the interior point code, this could be acceptable. Unfortunately, the main drawback of primal-dual interior point methods in semidefinite programming is the long computation time per iteration. In each iteration, a system $M\Delta y = \ldots$ has to be solved for $\Delta y \in \mathbb{R}^m$ where m is the number of constraints of the semidefinite program and M is a dense positive definite matrix of order m (see Section 4.4). Factorizing M needs $m^3/3$ arithmetic operations and storage requirements are $O(m^2)$. Since m grows quickly in the case of cutting plane algorithms, storing and factorizing M is the bottleneck in our algorithm. On our machine, memory requirements limit the number of cutting planes to roughly 5000, but computation time of the factorization is prohibitive even for $m = 3000$, because we would like to modify and recompute the relaxations repeatedly. In comparison to linear programming relaxations with millions of constraints, such problems would be considered to be of small size.

6.1.1 Max-Cut

The max-cut problem (MC) (page 27) and its semidefinite relaxation (SMC) (page 28) have been described in detail in Section 3.1, but we repeat the most important definitions and facts for the convenience of the reader. For an undirected graph $G = (V, E)$ with node set $V = \{1, \ldots, n\}$, edge set $E = \{ij : i < j, i, j \in V\}$, and weighted adjacency matrix $A = [a_{ij}]$, denote by $\delta(S)$ for $S \subseteq V$ the set of edges that have one endpoint in S and the other endpoint in $V \setminus S$. Then the max-cut problem is defined by

$$\text{(MC)} \qquad \text{mc}(G) = \max_{S \subseteq V} \sum_{ij \in \delta(S)} a_{ij}.$$

By (3.3) it may equivalently be formulated as a quadratic $\{-1, 1\}$ programming problem with cost matrix $C = \frac{1}{4}(\text{Diag}(Ae) - A)$ (e denotes the vector of all ones, $\text{Diag}(\cdot)$ transforms a vector in $\mathbb{R}^n$ to a diagonal matrix in S_n),

$$\max_{x \in \{-1,1\}^n} x^T C x.$$

The Initial Semidefinite Relaxation

Since $x^T C x = \langle Cx, x\rangle = \langle C, xx^T\rangle$ and the matrix xx^T is positive semidefinite with diagonal entries equal to 1, relaxing xx^T to $X \succeq 0$ gives rise to the semidefinite relaxation ($\text{diag}(\cdot)$ extracts the diagonal of a symmetric matrix)

$$\text{(SMC)} \qquad \begin{array}{rl} \max & \langle C, X\rangle \\ \text{s.t.} & \text{diag}(X) = e \\ & X \succeq 0 \end{array} \qquad\qquad \begin{array}{rl} \min & e^T u \\ \text{s.t.} & C + Z - \text{Diag}(u) = 0 \\ & Z \succeq 0. \end{array}$$

Primal and dual feasible sets contain strictly feasible solutions, so optimizers X_* and (u_*, Z_*) exist and satisfy $\langle C, X_* \rangle = \langle b, u_* \rangle$ (Theorem 2.2.5). Its primal feasible set is a relaxation of the cut polytope (3.12)

$$P_C = \operatorname{conv} \left\{ xx^T : \; x \in \{-1,1\}^n \right\}.$$

The primal feasible set and P_C have the same vertices and enjoy invariance under the "flipping" operation, *i.e.*, let $y \in \{-1,1\}^n$, then for $xx^T \in P_C$ the flipped element $xx^T \circ yy^T = (x \circ y)(x \circ y)^T$ is also in P_C and for $X \in \{X \succeq 0 : \operatorname{diag}(X) = e\}$ the flipped matrix $X \circ yy^T$ is again primal feasible because of the Schur product theorem 1.2.11 (Laurent and Poljak [1995]). Therefore we can expect that the primal set wraps P_C in a symmetric way.

Symmetry considerations prove that the identity $X = I$ is the analytic center and the center of gravity of the primal feasible set of (SMC) and of P_C. The identity $X = I$ is, thus, the ideal primal starting point, because it is on the central path for all cost matrices C. For the dual starting point it suffices to choose $u = \lambda e$ with λ large enough so that $Z = \operatorname{Diag}(u) - C$ is diagonally dominant (Theorem 1.1.12) (the dual analytic center is approached for $\lambda \to \infty$).

Remark 6.1.1 *In linear programming the natural relaxation of max-cut for a not necessarily complete graph $G = (V, E)$ introduces variables x_{ij} only for $ij \in E$. For example, for toroidal grid graphs that arise in modeling Ising spin glasses, C is extremely sparse, $|E| = 2n$. On these special graphs, instances with several thousand nodes have been solved using linear cutting plane methods (see Jünger and Rinaldi [1998] and references therein). In semidefinite primal-dual algorithms we cannot make use of this structure because X is dense; our primal-dual approach is definitely not competitive for sparse problems. The dual method of Benson, Ye, and Zhang [2000] is able to exploit the sparsity of the dual variable Z, but the system matrix for computing the step direction is again in general a dense positive definite matrix of order m. Exploiting sparsity to full extent should be possible by using the spectral bundle method, see Section 6.2.*

Cutting Planes and Separation Algorithms

We employ separation heuristics for the following classes of valid inequalities. For $b \in \mathbb{Z}^n$ with $\min \left\{ (b^T x)^2 : \; x \in \{-1,1\}^n \right\} = 1$ the inequalities $b^T X b \geq 1$ are called *hypermetric* (see (3.13) and thereafter). They specialize to the facet-defining *clique inequalities* for $b \in \{-1,0,1\}^n$ with an odd number of nonzero entries (Barahona, Grötschel, and Mahjoub [1985]). Clique inequalities with three nonzero elements are called *triangle inequalities* (3.14).

For a given primal feasible X_c (the current iterate in the interior point method) we employ three different strategies for separating inequalities which are violated by X_c. First, we enumerate all triangle inequalities. This requires $O(n^3)$ arithmetic operations and is comparable to or significantly cheaper than an interior point iteration, depending on the number of constraints in the relaxation.

Second, we try to extend clique inequalities which have already proven to be important (the corresponding dual cost is "sufficiently large") by adding two more nodes. The two nodes and their signs are determined by complete enumeration over the zero components of the old inequality ($O(n^2)$ for each old inequality). The best two nodes with respect to a geometric criterion (see below) yield the candidate.

Finally, we employ a very simple heuristic to construct general hypermetric candidates. We are looking for integer vectors b with the following three properties. The sum of the elements of b is odd, $b^T X_c b$ is close to zero, and $| \langle b, c \rangle | = 1$ where c is the currently best cut vector obtained by applying a simple rounding and exchange heuristic to the rows of feasible primal matrices. The third property is motivated by the idea that the new inequality should be tight for the optimal solution. We start with a vector that satisfies the first and third property (we use e_i). Then we try to decrease $b^T X b$ while maintaining these two properties. Iteratively, we run through indices j from 1 to n. For each index j, we construct for each $k \neq j$ the vector

$$b^k = b + \begin{cases} e_j + e_k & \text{if } 1 \cdot c_j + 1 \cdot c_k = 0 \\ e_j - e_k & \text{if } 1 \cdot c_j + (-1) \cdot c_k = 0. \end{cases}$$

The case distinction guarantees that the third property is maintained. Let $\bar{k}$ be the index so that $(b^k)^T X_c b^k$ is minimal. If $(b^{\bar{k}})^T X_c b^{\bar{k}} < b^T X_c b$ then we replace b by $b^{\bar{k}}$ otherwise b remains unchanged for this j. Continue with the next j. The iteration over all j is repeated till all indices were tried without further progress. If in the end $b^T X_c b < 1$, then this is a violated inequality.

Remark 6.1.2 *The simple nature of the separation routines is due to the fact that we are working on the complete graph. On a sparse graph the inequalities corresponding to the projection of the triangle inequalities are the so-called odd cycle constraints (Barahona [1993]), these would be significantly more laborious to separate (even though the order of complexity remains $O(n^3)$). In linear programming, separating outside the support does not make much sense, in semidefinite programming it does. Including all triangle inequalities yields a slightly better relaxation than separating odd cycles on the support (see Gruber and Rendl [1999]). The semidefiniteness constraint provides a global link between all matrix elements.*

Selecting A Subset of Violated Inequalities

In our numerical experiments the separation routines generated a lot more violated inequalities for a given feasible X_c than we were willing to include in our relaxation, because adding k inequalities implied that each Newton step required $O((m+k)^3)$. To keep the code reasonably efficient it was important to select just a small number of promising inequalities from the vast set of violated inequalities.

The first criterion that comes to mind is the amount of violation. This was a satisfactory criterion as long as triangle inequalities were used exclusively. For general inequalities, we got better results by the following *geometric criterion*. For a violated inequality compute the intersection of the straight line segment between X_c and I, the barycenter of the primal feasible set. We prefer inequalities with small distance of this intersection to I.

During the separation process we maintain a heap of the best inequalities with respect to the geometric criterion. The heap offers room for a certain number of inequalities. If the heap is full, a new inequality is added to the heap only if it is better than the worst inequality in the heap. In this case the worst is replaced by the new inequality.

In the following we assume that $m-n$ cutting planes of hypermetric type have been added to the initial relaxation. We code these in the linear constraints $\mathcal{A}X - s = e$, where $s \in \mathbb{R}^{m-n}_+$ is a vector of slack variables. Then the semidefinite relaxation reads

$$\text{(SMC')} \quad \begin{array}{ll} \min & \langle C, X\rangle \\ \text{s.t.} & \operatorname{diag}(X) = e \\ & \mathcal{A}X - s = e \\ & X \succeq 0,\ s \geq 0 \end{array} \qquad \text{(DMC')} \quad \begin{array}{ll} \max & \langle e, u_d\rangle + \langle e, u_{\mathcal{A}}\rangle \\ \text{s.t.} & C + Z = \operatorname{Diag}(u_d) + \mathcal{A}^T u_{\mathcal{A}} \\ & t = -u_{\mathcal{A}} \\ & Z \succeq 0,\ t \geq 0. \end{array}$$

The variables $u_d \in \mathbb{R}^n$ and $u_{\mathcal{A}} \in \mathbb{R}^{m-n}$ are the Lagrange multipliers corresponding to the diagonal and the cutting plane constraints, respectively. The slack variables $s \in \mathbb{R}^{m-n}_+$ and $t \in \mathbb{R}^{m-n}_+$ will turn out to be convenient for our warm-start heuristic.

Warm-Start

In cutting plane algorithms it is of fundamental importance that the basic operation, adding a few cutting planes and reoptimizing, can be carried out in reasonable time. Since the new inequalities are separated with respect to the current iterate X_c, this point is infeasible after addition of the cutting planes. Therefore we have to construct a new feasible primal point X^+ for restarting the method. The interior point algorithm will work better if this point X^+ is in the neighborhood of the central path. We have considered the following two heuristic approaches for constructing X^+ (see Mitchell and Borchers [1996] for similar approaches in linear interior point cutting plane methods).

- *Backtracking along iterates.* Store all previous iterates and restart from the last iterate that is feasible with respect to the new inequalities. This point is, hopefully, reasonably close to the new central path.

- *Backtracking to the analytic center.* Backtrack towards I along the straight line between last iterate X_c and I. We hope that this straight line is reasonably close to the central path.

Both alternatives exhibited similar performance, so we decided in favor of the second approach because it required less memory. The new primal slack variables are determined by feasibility, see (SMC').

For the dual variables we did not succeed in constructing a good direction into the interior. Therefore we simply set the new u_i-values corresponding to the new inequalities to zero, and the corresponding t_j-values (the counterparts to the slack variables s_j) to one. This results in an infeasible dual point because feasibility requires $t_i = -u_i$. None the less the dual objective function value remains a valid upper bound as long as the u_i do not get positive. Due to the line search in the choice of the step size of the interior point algorithm, the t_i cannot get negative and since the Newton direction tries to satisfy $t_i = -u_i$, a positive y_i should be a rare event. As a safeguard, we iterate till a full dual Newton step is performed, this removed infeasibilities after roughly three iterations.

The restarting heuristic worked successfully if both, primal and dual solutions were still well in the interior of the semidefinite sets at the time of separation. It failed for the optimal solutions of the relaxations, *i.e.*, there was no advantage as compared to restarting from scratch. A heuristic explanation for this fact is that the location of the optimal solution may change considerably with the addition of cutting planes, so that the current primal-dual pair is far away from the neighborhood of the central path. The matrix X_c has to be modified significantly to obtain feasibility, thereby making it a bad match for the almost singular dual variable, which has not been changed. Our heuristic attempts to modify the dual variables appropriately failed, as well.

In connection with this warm-start procedure it turned out to be advantageous to check the violation of triangle inequalities in every iteration. When a triangle inequality is violated we continue the interior-point algorithm for two more steps in order to ensure that a significant amount of violated inequalities is available and then invoke all separation routines on this iterate X_c. We call this approach *early separation*.

Empirically, this resulted in significantly faster improvements of the bound in comparison to separating with respect to the optimal solution of the relaxation. We do not have a theoretically sound explanation for this phenomenon but can offer some heuristic arguments. Usually, the central path starts off from the center into the direction of the cost function and later bends towards the optimal solution of the relaxation. The early separation approach stops the algorithm as soon as it leaves the metric polytope (the set of points satisfying all triangle inequalities). One hopes that X_c is still close to P_C and that all inequalities violated at X_c are squeezed between P_C and X_c so that they cover the face of P_C that is traversed by the central path on its way out of P_C. Then the barrier terms of the slack variables of the new inequalities will push the new central path into a quite different direction, to the effect that the separation process seems to cover different regions of P_C in the direction of the cost function within a short time.

Elimination of Inequalities

We have already pointed out the negative influence of a large number of inequalities on the running time of a single interior point iteration. In order to keep the number of constraints as small as possible we remove cutting planes that are not tight in the optimal solution of the relaxation. In the simplex method of linear programming superfluous cutting planes can be recognized by their positive slack variables. In interior point methods the primal slack variables may converge to zero but they never reach zero. Therefore we prefer to eliminate inequalities whose dual variables are close to zero. Sensitivity analysis asserts that this cannot worsen the relaxation by much and we do not mind removing slightly more inequalities than necessary in order to speed up the Newton steps.

Repeated early separation and restarting prevents the dual and primal iterates from converging. This makes it impossible to recognize superfluous inequalities. At the same time, early separation increases the number of constraints so that the single iterations become computationally expensive

Table 6.1: The interior-point cutting plane algorithm for a max-cut problem of type $G_{.5}$ with $n = 70$. The maximum cut value is 708. *Iter* refers to the number of interior point iterations, *hh:mm:ss* gives the time in hours:minutes:seconds, *ubd* lists the optimal value of the current relaxation, m displays the number of constraints. The first row gives the value of (SMC), each further row is the result of separating n inequalities with respect to the solution of the previous relaxation plus separating ten times $n/3$ cutting planes upon violation of triangle inequalities by an iterate. The increase in computation time per iteration in dependence on the number of constraints is clearly visible.

Iterations	h:mm:ss	ubd	m
11	1	726.71	70
59	19	715.58	370
111	3:55	713.34	667
166	18:34	712.08	918
219	50:06	711.45	1124
277	1:58:47	711.06	1345

quickly. Therefore we compute the exact value of the relaxation from time to time in order to reduce the number of constraints and to check whether the integrality gap is closed sufficiently.

Numerical Results

After extensive numerical experiments we decided to use the following parameter settings. First we solve the initial semidefinite relaxation (SMC) to optimality without early separation. With respect to its optimal solution we separate and select n cutting planes. Then we restart from the center, perform 10 early separation and restarting steps (each separating and selecting $n/3$ inequalities), after this we compute an optimal solution of the current relaxation and eliminate inequalities with small dual costs. With respect to this solution we again separate and select n cutting planes, restart from scratch and continue with ten early separation steps, and so on, till the exact solution is found or a time limit is reached. After this time limit the code computes the solution of the current relaxation without calling the separation procedures and stops.

We illustrate the behavior of the algorithm on six classes of random graphs (the same instances were used in Helmberg and Rendl [1998]). $G_{.5}$ consists of unweighted graphs with edge probability $1/2$, $G_{-1/0/1}$ of weighted (complete) graphs with edge weights chosen uniformly from $\{-1, 0, 1\}$. G_p is a class of almost planar graphs with edge weights 1, in $G_{\pm p}$ the edge weights of the same graphs are chosen randomly from $\{-1, 1\}$. Q_{100} and $Q_{100,.2}$ were used by Williams [1985]; Barahona, Jünger, and Reinelt [1989]. Formulating Q_{100} with respect to (QP)(see Section 3.2), the lower triangle of B is set to zero, the upper triangle (including the diagonal) is chosen uniformly from $\{-100, \ldots, 100\}$. The diagonal takes the role of the linear term. $Q_{100,.2}$ represents instances with a density of 20%.

We start by looking at a particular instance of $G_{.5}$ in Table 6.1. The columns are explained in the caption. The first row corresponds to (SMC), each further row is the result of a full phase of adding inequalities ($n + 10 \cdot n/3$). Each phase takes about 50 interior point iterations, but computation time is increasing dramatically with the number of constraints (from 20 seconds for the first phase to one hour for the last phase listed). We point out that the results of Table 6.1 exhibit the typical tailing off effect often observed for cutting plane methods; the most significant improvement of the bound is achieved in the first phase.

Table 6.2 lists average results of the cutting plane algorithm when applied to the six random classes of cost matrices described above. All the problems listed can be solved exactly by employing a branch and bound approach, see Helmberg and Rendl [1998]. Partial results are also listed in Table 6.13 of Section 6.3. Before interior point methods for semidefinite programming became available, problems on complete graphs with 40 nodes were considered unsolvable, see De Simone and Rinaldi [1994] (they give results on at most 25 nodes). Here, the combination of interior point

Table 6.2: Average results by using the interior cutting plane algorithm for randomly generated max-cut examples with a time limit of two hours. n is the dimension, nr gives the number of instances, $h:mm:ss$ lists the average computation time in hours:minutes:seconds, *solved* displays the number of max-cut instances solved to optimality, *gap* gives the average relative error over the unsolved instances in percent.

n	nr	h:mm:ss	solved	gap(%)	h:mm:ss	solved	gap(%)
		$G_{.5}$			$G_{-1/0/1}$		
40	10	14	10	0	8	10	0
50	10	1:38	10	0	1:55	10	0
60	5	54:20	4	0.4	30:12	4	1.2
70	3	1:37:23	1	0.3	52:52	2	1.5
80	2	1:58:44	1	0.4	2:15:30	0	1.7
90	1	2:19:51	0	0.5	2:18:23	0	5.5
100	1	2:18:30	0	0.8	2:23:56	0	5.4
		G_p			$G_{\pm p}$		
40	10	4	10	0	1	10	0
50	10	29	10	0	6	10	0
60	5	2:40	5	0	45	5	0
70	3	4:10	3	0	2:50	3	0
80	2	46:42	2	0	2:29	2	0
90	1	7:02	1	0	17:49	1	0
100	1	2:20:45	0	0.9	48:23	1	0
		Q_{100}			$Q_{100,.2}$		
41	10	27	10	0	2	10	0
51	10	1:33	10	0	13	10	0
61	5	3:13	5	0	1:03	5	0
71	3	8:39	3	0	7:37	3	0
81	2	18:51	2	0	3:54	2	0
91	1	6:28	1	0	5:31	1	0
101	1	1:54:49	1	0	9:51	1	0

methods and cutting planes certainly helped to push the boundary. Unfortunately, most max-cut instances of practical relevance are much larger and sparse.

6.1.2 Quadratic 0-1 Knapsack

The quadratic knapsack problem (QK) and various semidefinite relaxations, (SQK1) to (SQK4), are introduced in Section 3.3.2. For the convenience of the reader we restate them here. The quadratic 0-1 knapsack problem reads

$$\text{(QK)} \qquad \begin{array}{rl} \max & y^T B y \\ \text{s.t.} & a^T y \leq b \\ & y \in \{0,1\}^n . \end{array}$$

The elements of a are called *weights*, the right hand side b is called *capacity*. Without loss of generality we assume that $0 \leq a_i \leq b$ for $i = 1, \ldots, n$. In addition, we assume that $a_i < b$, since $a_i = b$ allows to decompose the problem.

In applications, the quadratic knapsack problem usually appears as the subproblem of modeling each single inequality of a larger system of linear constraints. Therefore, our goal is not so much to solve the quadratic knapsack problem efficiently (specialized branch and bound routines are much faster, see Caprara, Pisinger, and Toth [1999]), but to develop separation techniques that are of practical value also in the case of several constraints. The discussion is based on the cutting plane algorithm for the quadratic knapsack problem that was used by Helmberg, Rendl, and Weismantel [2000]. Here, we follow the more detailed description given by Helmberg and Weismantel [1998].

The Initial Semidefinite Relaxation

Like in the relaxation of max-cut, observe that $y^T By = \langle By, y\rangle = \langle B, yy^T\rangle$. For $\bar{y} = \begin{bmatrix} 1 \\ y \end{bmatrix}$ the matrix $\bar{y}\bar{y}^T$ is positive semidefinite and the first column equals the diagonal. We relax $\bar{y}\bar{y}^T$ to

$$\bar{Y} = \begin{bmatrix} 1 & \operatorname{diag}(Y)^T \\ \operatorname{diag}(Y) & Y \end{bmatrix} \succeq 0,$$

which is equivalent to $Y - \operatorname{diag}(Y)\operatorname{diag}(Y)^T \succeq 0$ by the Schur complement theorem 1.1.9. It is convenient to define the index of the first row and column of $\bar{Y}$ to be zero. With this we may set up the following four basic relaxations (Section 3.3.2).

$$\begin{array}{lll} \text{(SQK1)} & \max & \langle B, Y\rangle \\ & \text{s.t.} & \langle \operatorname{Diag}(a), Y\rangle \le b \\ & & Y - \operatorname{diag}(Y)\operatorname{diag}(Y)^T \succeq 0 \end{array} \qquad \begin{array}{lll} \text{(SQK2)} & \max & \langle B, Y\rangle \\ & \text{s.t.} & \langle aa^T, Y\rangle \le b^2 \\ & & Y - \operatorname{diag}(Y)\operatorname{diag}(Y)^T \succeq 0 \end{array}$$

$$\begin{array}{lll} \text{(SQK3)} & \max & \langle B, Y\rangle \\ & \text{s.t.} & \left\langle \begin{bmatrix} b \\ -a \end{bmatrix} \begin{bmatrix} 0 \\ a \end{bmatrix}^T, \bar{Y} \right\rangle \ge 0 \\ & & Y - \operatorname{diag}(Y)\operatorname{diag}(Y)^T \succeq 0 \end{array} \qquad \begin{array}{lll} \text{(SQK4)} & \max & \langle B, Y\rangle \\ & \text{s.t.} & \sum_{j=1}^{n} a_j y_{ij} \le b y_{ii} \quad i = 1 \dots n \\ & & Y - \operatorname{diag}(Y)\operatorname{diag}(Y)^T \succeq 0. \end{array}$$

In Lemma 3.3.3 we proved that the feasible sets of Y-matrices $\mathcal{Y}_i$ of (SQKi) for $i = 1, \dots 4$ can be ordered with respect to set inclusion,

$$\mathcal{Y}_1 \supseteq \mathcal{Y}_2 \supseteq \mathcal{Y}_3 \supseteq \mathcal{Y}_4. \tag{6.1}$$

For all four problems, $\mathcal{Y}_i$ is a relaxation of the polytope

$$P_{QK} = \operatorname{conv}\{Y = yy^T : \ y \in \{0,1\}^N, a^T y \le b\}, \tag{6.2}$$

which we call the *quadratic knapsack polytope*. If $a_i + a_j > b$ for some $i \neq j$ then $Y_{ij} = 0$ for all $Y \in P_{QK}$; therefore we add the constraints $Y_{ij} = 0$ to the semidefinite relaxations for all such i and j. The matrices Y generated by the zero-element, the one-element, and feasible two-elements solutions (here, a k-element solution refers to a vector $y \in \{0, 1, \}^n$ with k ones and zeros otherwise) form an affinely independent set in P_{QK} and therefore their arithmetic mean Y_A is in the relative interior of P_{QK}.

In our cutting plane method for max-cut we have used the identity I as the reference point in a geometric criterion for selecting violated inequalities. Generalizing heuristically, we would expect this criterion to work reasonably well for several cost matrices if the reference point is well centered inside the underlying combinatorial polyhedron. The identity is certainly well centered in the max-cut polytope P_C. The point Y_A, however, is not likely to be well centered in P_{QK}, because it is inside the convex hull of the solutions with small support. We hope to improve the quality of the reference point and the starting point by including solutions with large support as well. Without attaching too much importance to this heuristic, we construct solutions with large support by starting from any solution of one or two elements and by iteratively adding elements with cyclically increasing indices that fit into the remaining capacity. We hope that this generates solutions with reasonable spread of the support. The arithmetic mean of these solutions and the previous solutions of small support gives rise to the point Y_S that is again in the relative interior of P_{QK}.

It is not difficult to check that the matrix $Y_S - \operatorname{diag}(Y_S)\operatorname{diag}(Y_S)$ is positive definite. We use Y_S as the primal starting point for the interior point method, because $Y_S \in \operatorname{relint} P_{QK}$ ensures that it remains strictly feasible throughout the cutting plane algorithm. By an appropriate linear combination of the coefficient matrices corresponding to the constraints $\bar{Y}_{00} = 1$ and $\bar{Y}_{0i} = \bar{Y}_{ii}$ we can construct a diagonally dominant dual feasible Z for the respective dual problem; this yields a strictly feasible dual starting point. Is this point well centered with respect to the central path?

Unfortunately, the analytic centers of the relaxations (SDP1) to (SDP4) cannot be expected to be contained in P_{QK} in general (in our numerical experiments it was outside P_{QK} for several examples). In fact, if the analytic center is not contained in P_{QK} then each hyperplane separating the analytic center from P_{QK} gives rise to a cost matrix B so that the associated central path cannot touch P_{QK}. Thus, in general we cannot hope to find a starting point in the relative interior of P_{QK} that is close to the central path.

Cutting Planes and Separation Algorithms

We now consider the separation problem for a given primal feasible Y_c. We present the inequalities that we use in three groups. The first group are *generic inequalities* that can be used without the need to understand the polyhedral structure of the knapsack problem and without restrictions on the coefficients of the linear constraint. The second group, *linear cutting planes*, makes use of cutting planes for the linear knapsack problem which are then lifted into quadratic space. Finally, the third group, *quadratic inequalities*, consists of cutting planes developed for the quadratic knapsack polytope P_{QK}.

Generic Inequalities. A generic way to improve the initial relaxations is to add valid inequalities for unconstrained quadratic 0-1 programming. These are certainly valid for constrained cases as well. In our implementation we include the constraints corresponding to the triangle inequalities in the max-cut setting (see Example 3.2.4),

$$\begin{aligned} y_{ij} &\geq 0 \\ y_{ij} &\leq y_{ii} \\ y_{ii} + y_{jj} &\leq 1 + y_{ij} \\ y_{ik} + y_{jk} &\leq y_{kk} + y_{ij} \\ y_{ij} + y_{ik} + y_{jk} + 1 &\geq y_{ii} + y_{jj} + y_{kk}. \end{aligned}$$

We separate them by enumeration.

In addition, we check for violated y_i- and $1 - y_i$-representations of $a^T y \leq b$ (see (3.25)). They correspond to multiplying the linear inequality by y_i or $1 - y_i$ for some index $i \in \{1, \ldots, n\}$ and then linearizing the terms $y_i y_j$ to Y_{ij}. In order to be able to exploit their low rank structure, we add them in the form

$$\left\langle \begin{bmatrix} b \\ -a \end{bmatrix} \begin{bmatrix} 0 \\ e_i \end{bmatrix}^T, \bar{Y} \right\rangle \geq 0, \tag{6.3}$$

$$\left\langle \begin{bmatrix} b \\ -a \end{bmatrix} \begin{bmatrix} 1 \\ -e_i \end{bmatrix}^T, \bar{Y} \right\rangle \geq 0. \tag{6.4}$$

These are again separated by enumeration. Observe, that (SQK4) already contains all inequalities of type (6.3).

Linear Cutting planes. For $a \in \mathbb{N}^n$, $b \in \mathbb{N}$, let

$$P_{LK} := \operatorname{conv}\left\{ y \in \{0,1\}^n : \sum_{i \in \{1,\ldots,n\}} a_i y_i \leq b \right\}$$

be the linear knapsack polyhedron. A first class of inequalities, that is valid for P_{LK}, is constructed by choosing a subset T of all elements that fit into the knapsack. These retain their weights as coefficients. Each remaining element is assigned, as a new weight, the amount by which the weight of the item and the subset exceeds the knapsack capacity (this value coincides with the coefficient that one obtains from lifting the inequality $\sum_{i \in T} a_i y_i \leq \sum_{i \in T} a_i$ according to any ordering of the items not contained in T).

Lemma 6.1.3 *(Weismantel [1997])***(weight inequalities)**
Let $T \subseteq \{1, \dots n\}$ *with* $r := b - \sum_{i \in T} a_i \geq 0$ *be given. The* weight inequality with respect to T

$$\sum_{i \in T} a_i y_i + \sum_{i \in \{1,\dots,n\} \setminus T} \max\{0, (a_i - r)\} y_i \leq \sum_{i \in T} a_i$$

is valid for P_{LK}.

The idea of weight inequalities can be extended to more general cases. Instead of working with the original weights of the items, we introduce "relative" weights and derive an analogon of weight inequalities for these relative weights.

Lemma 6.1.4 *(Weismantel [1997])***(extended weight inequalities)**
Let I *and* T *be disjoint subsets of* $\{1, \dots n\}$ *satisfying* $a_t \leq a_i \leq \sum_{j \in T} a_j$ *for all* $t \in T, i \in I$ *and* $r := b - \sum_{t \in T \cup I} a_i \geq 0$. *Define relative weights*

$$\begin{array}{ll} w_t = 1 & \forall t \in T, \\ w_i = \min\left\{ |S| : \ S \subseteq T, \sum_{t \in S} a_t \geq a_i \right\} & \forall i \in I. \end{array}$$

For $z \in \{1, \dots, n\} \setminus (T \cup I)$ *with lifted coefficient*

$$w_z = \min \left\{ \sum_{s \in S} w_s : \ S \subseteq (T \cup I), \sum_{s \in S} a_s \geq a_s - r \right\}$$

the extended weight inequality with respect to $T \cup I \cup \{z\}$

$$\sum_{i \in (T \cup I \cup \{z\})} w_i y_i \leq \sum_{i \in (T \cup I)} w_i$$

is valid for P_{LK}.

This inequality can be extended by standard lifting techniques. For a given ordering of the items in $\{1, \dots, n\} \setminus (T \cup I \cup \{z\})$ one can compute, step by step and in polynomial time, the maximum coefficient so that the lifted inequality remains valid.

The most difficult part in the separation process is the construction of promising sets T (and I). Once these are fixed the coefficients can be computed by a lifting procedure. Each of these inequalities can be represented in the quadratic space by the techniques discussed in Section 3.3.2. On account of Lemma 3.3.3 it suffices to consider the representations (6.3) and (6.4) arising from multiplication with y_i or $1 - y_i$.

It is quite unlikely that quadratic representations of weight or extended weight inequalities are violated by Y_c if all the variables in the support of the inequality attain small values. This motivates the search for inequalities whose support is contained in the set of variables with large Y_{ij}-values. In a first step we interpret the variables Y_{ii} as variables $y_1, \dots, y_n$ of a linear problem, see lemmas 6.1.3 and 6.1.4. We sort them by value in nonincreasing order. For later reference we denote this order by the symbol $\succ_d$. We first determine the maximal number k_0 such that the first k_0 elements (with respect to $\succ_0$) have weight not greater than b. These k_0 indices form the set T of a weight inequality.

In order to construct extended weight inequalities we choose for every $k > k_0$ the k first elements with respect to $\succ_d$. This forms the set J_k. We sort the elements of J_k with respect to nondecreasing weights and determine the maximal number h such that the first h elements have weight not greater than b. This defines the set $J_k^{\subset}$. If $k > k_0 + 1$ and $J_k^{\subset}$ is contained in $J_{k-1}^{\subset}$, we proceed to index $k + 1$. Otherwise every partition of $J_k^{\subset}$ into feasible sets T and I according to Lemma 6.1.4 gives rise to an extended weight inequality: We lift the elements not in $J_k^{\subset}$ following the order $\succ_d$; the item z is the first element of this order.

To generate promising starting sets for y_j-representations of linear knapsack inequalities it seems reasonable to work with variables Y_{ij} rather than Y_{ii}. The procedures outlined before can be easily adapted to this situation. One replaces $\succ_d$ by an appropriate order.

Quadratic cutting planes. The linear knapsack polytope P_{LK} is a convenient starting point for deriving cutting planes because it is well studied in the literature. However, the ultimate goal of a cutting plane algorithm is to arrive at the convex hull of all feasible integral solutions, the quadratic knapsack polytope P_{QK} of (6.2). The following lemma introduces a family of polytopes which are relaxations of P_{QK}.

Lemma 6.1.5 *(Helmberg, Rendl, and Weismantel [2000])*
Let $N_1, \ldots, N_k$ be a partition of $\{1, \ldots, n\}$. For every $v \in \{1, \ldots, k\}$ we choose a spanning tree (N_v, T_v) in the complete graph $K(N_v)$ on the node set N_v. By deg_i^v we denote the degree of node i in the tree (N_v, T_v). The polyhedron

$$\operatorname{conv}\left\{ Y \in S_n : \; y_{ij} \in \{0,1\}, \sum_{v=1}^{k} (\sum_{i \in N_v} a_i)[\sum_{ij \in T_v} y_{ij} + \sum_{i \in N_v} (1 - deg_i^v) y_{ii}] \leq b \right\}$$

contains all the feasible points of P_{QK}.

These polytopes may be interpreted as linear 0-1 knapsack polytopes defined over the upper triangle of Y. Hence we can use the same machinery as in the linear case to derive valid inequalities for any of these polytopes. Since these polytopes are relaxations of P_{QK}, their valid inequalities remain valid for P_{QK}.

In the special case when the sets N_i of Lemma 6.1.5 contain either one or two elements, we may associate every spanning tree with an isolated node or an edge. Collecting all isolated nodes in the set V and all edges in the set E, the inequality of Lemma 6.1.5 reads

$$\sum_{ij \in E} (a_i + a_j) y_{ij} + \sum_{i \in V} a_i y_{ii} \leq b. \tag{6.5}$$

We call an inequality of this form a *matching knapsack constraint.*

Our implementation includes only a separation routine for matching knapsack constraints. It uses the following ingredients: We compute a maximum weight matching[2] on a principal submatrix of Y. The support of the principal submatrix is determined by the indices in J_k, $k > k_0$, see the separation routines for linear cutting planes. For each J_k, $k > k_0$, our routine returns the maximum weight matching in form of sets E and V of (6.5). This defines the knapsack polyhedron

$$\operatorname{conv}\left\{ x \in \{0,1\}^{|E|+|V|} : \; \sum_{ij \in E} (a_i + a_j) x_{ij} + \sum_{i \in V} a_i x_i \leq b \right\}.$$

We apply the separation routines for weight and extended weight inequalities for the point with components Y_{ij}, $ij \in E$ and Y_{ii}, $i \in V$.

Selecting a Subset of Violated Inequalities

For a given iterate Y_c our separation routines delivered large quantities of the generic triangle inequalities and from time to time linear or quadratic cutting planes. For choosing a subset of these we use, with respect to Y_S, the same geometric criterion that we described for the max-cut cutting plane algorithm.

Warm-Start

We tried to employ the same early separation and warm-start techniques that were rather successful in the max-cut case. Unfortunately, these approaches did not work at all for our quadratic knapsack examples. In the following we report on our observations during these experiments.

[2] We solve the maximum weight matching problem using routines from the LEDA package of Mehlhorn, Näher, Seel, and Uhrig [1999].

Table 6.3: Comparison of relaxations (SQK1), (SQK2), (SQK3) and (SQK4). The instances are based on compiler design problems taken from Johnson, Mehrotra, and Nemhauser [1993]. *dim* is the number of 0-1 variables, each problem was examined for the three capacities of column *rhs*. *feas.* gives the best solution we know. The relative gap is computed via $100 \cdot (\frac{\text{upper bound}}{\text{feas.}} - 1)$.

			relative gap (%)			
dim	rhs	feas.	SQK1	SQK2	SQK3	SQK4
30	450	1580	41.0	17.3	14.5	13.5
	512	1802	38.9	19.6	17.1	16.7
	600	2326	23.9	12.0	10.6	10.6
45	450	2840	15.6	8.7	8.5	8.4
	512	3154	29.4	12.7	12.7	12.7
	600	3840	21.7	8.2	8.2	8.2
47	450	1732	6.7	5.9	5.9	5.8
	512	1932	29.7	11.9	11.6	11.6
	600	2186	30.8	17.7	17.5	17.4
61	450	26996	3.7	2.4	2.4	2.4
	512	29492	2.9	2.0	2.0	2.0
	600	32552	2.6	1.9	1.9	1.9

As we have mentioned in the discussion on the choice of the starting point on page 110, we have repeatedly observed in our numerical experiments, that the analytic center of the initial relaxation is not contained in the quadratic knapsack polytope P_{QK}. Indeed, we suspect that in several of the examples the central path does not even touch P_{QK} (we have not tried to verify this computationally). After starting the interior point method from Y_S, the first few iterations were needed to obtain a primal-dual pair of reasonable quality. During this time the barrier parameter was rather large and the primal iterates moved towards the corresponding point on the central path. The direction of movement had little to do with the actual objective function. If the separation algorithms were called for these iterates, then a large number of inequalities was delivered but almost all of them were not tight in the optimal solution of the relaxation. The restarting strategy had the effect that the primal iterate was thrown out of the neighborhood of the central path and the same phenomenon occurred again. Due to the sharp increase in computation time and memory consumption for a growing number of inequalities it was computationally infeasible to keep all these cutting planes with the sole purpose to control the location of the central path. Solving the relaxation to optimality and separating with respect to the optimal solution led to faster progress.

Elimination of Inequalities

Like in the algorithm for max-cut we eliminate inequalities with small dual variables whenever an optimal solution of a relaxation has been computed.

Numerical Results

We start with comparing, in Table 6.3, the initial relaxations (SQK1) to (SQK4) with respect to instances based on compiler design problems taken from Johnson, Mehrotra, and Nemhauser [1993]. In these examples, the effect of the hierarchical structure of the feasible sets (6.1) is clearly visible. From a computational point of view, the remarkable quality of (SQK3) is important. It is significantly better than (SQK1) and slightly better than (SQK2), while being comparable in computation time. In comparison to (SQK4), (SQK3) is almost as good in quality, but significantly faster to compute, because it is formed by just one constraint as opposed to n constraints for (SQK4). Computational experiments reveal that the addition of a few y_i-representations to

Table 6.4: Improvement of the bound by adding cutting planes for the examples of Table 6.3. The initial relaxation is (SQK3). For each solution we eliminate inequalities with small dual variables, add n violated inequalities, and iterate. Relative gaps refer to the bound after 30 minutes of computation time. If the relative gap was closed to $5 \cdot 10^{-6}$ within this time limit, we mark this by $\triangleright$ and give the computation time instead. In *gen. ineq.* the generic inequalities are used as cutting planes. In *lin. cuts* the knapsack specific linear cutting planes are separated as well. In *matching* we consider all inequalities together, including the matching constraints.

dim	rhs	feas. sol.	SQK3 %	gen. ineq. %/mm:ss	lin. cuts %/mm:ss	matching %/mm:ss
30	450	1580	14.5	0.23	$\triangleright$ 0:03	$\triangleright$ 0:03
	512	1802	17.1	5.60	0.20	0.20
	600	2326	10.6	2.64	$\triangleright$ 0:24	$\triangleright$ 0:13
45	450	2840	8.5	2.88	$\triangleright$ 0:12	$\triangleright$ 2:00
	512	3154	12.7	3.06	1.48	1.48
	600	3840	8.2	$\triangleright$ 3:32	$\triangleright$ 0:29	$\triangleright$ 0:29
47	450	1732	5.9	2.55	$\triangleright$ 4:03	$\triangleright$ 3:08
	512	1932	11.6	1.28	$\triangleright$ 1:22	$\triangleright$ 2:10
	600	2186	17.5	8.89	3.12	3.51
61	450	26996	2.4	0.36	$\triangleright$ 0:40	$\triangleright$ 0:36
	512	29492	2.0	1.28	$\triangleright$ 7:44	$\triangleright$ 5:14
	600	32552	1.9	1.04	0.23	0.28

Table 6.5: Cutting planes used. The numbers correspond to the solution given in column *matching* of Table 6.4. *knapsack* refers to y_i-representations of the original knapsack constraint. *triangle* gives the number of triangle inequalities. *weight* is the number of weight inequalities in y_i-representation. *ext. weight* refers to the y_i-representations of extended weight inequalities, and *matching* to the number of matching inequalities.

dim	rhs	knapsack	triangle	weight	ext. weight	matching
30	450	0	69	4	5	0
	512	4	557	7	13	0
	600	4	81	5	6	2
45	450	0	160	10	20	6
	512	9	589	11	36	1
	600	2	96	7	5	0
47	450	1	143	4	39	7
	512	5	158	9	23	1
	600	12	371	6	9	7
61	450	0	105	1	3	2
	512	4	274	6	13	0
	600	3	399	8	15	1

(SQK3) suffices to achieve the quality of (SQK4). Therefore, our cutting plane algorithm uses (SQK3) as initial relaxation.

Because of the lack of an acceptable warm-start heuristic, the structure of our cutting plane algorithm is simple. Start from Y_S with relaxation (SQK3). Solve the current relaxation to optimality, eliminate inequalities with small dual costs, separate n inequalities with respect to the optimal solution, and iterate by restarting from Y_S. We selected the rather small parameter n, because many initial cutting planes turned out to be superfluous at a later stage of the algorithm. A larger number than n only led to slower interior point iterations without comparable improvements in objective value.

In Table 6.4 we present the computational results of four different runs of the cutting plane algorithm. Starting from (SQK3) we include in each run a further class of cutting planes. First we

separate only generic inequalities, then we add the linear cutting planes, and finally the quadratic cutting planes.

The generic inequalities already led to considerable improvements with respect to the initial relaxation, see column $\triangle$-*ineq.* of Table 6.4, but they rarely sufficed to close the gap between feasible solution and upper bound. With the separation of the problem specific linear cutting planes the algorithm succeeded in proving optimality for most of our examples, see column *lin. cuts.* With respect to matching constraints the computational results are ambivalent, see column *matching.* Sometimes the code was faster with matching constraints, mostly it was slower due to the expensive computation of maximum weighted matchings. Table 6.5 lists the various cutting planes that were active at termination of the code.

Again, the examples we could attack with our semidefinite interior point cutting plane methods are tiny. None the less, interior point methods form an excellent test bed for the development of separation routines and help to provide some intuition on the importance of certain classes of cutting planes.

6.2 A Spectral Bundle Approach

The spectral bundle method of Chapter 5 is designed for eigenvalue optimization problems of the form $\min_y \lambda_{\max}(C - \mathcal{A}^T y) + b^T y$ and may be employed to solve the dual of most semidefinite relaxations described in Chapter 3. In particular, the duals of relaxations whose primal feasible matrices have constant trace may equivalently be written as eigenvalue optimization problems of this form (Proposition 5.1.1). For the semidefinite relaxation of max-cut the transformation is given explicitly in Lemma 3.1.11. The same transformation applies to the semidefinite relaxation of max k-cut which only differs in additional inequality constraints. For constrained quadratic 0-1 programming the scaling of Lemma 3.2.3 to max-cut shows that it also falls into this class.

The main advantage of the spectral bundle method is the possibility to exploit structural properties of the cost and coefficient matrices, such as sparsity or low rank representations. Its memory usage can be kept at the same order of magnitude as the input data. The order n of the feasible matrices and the number m of constraints is usually not a limiting factor.

Its main disadvantage is that it is a first order method. Initially, the objective value improves quickly, but convergence slows down dramatically as the optimum is approached.

In combinatorial applications the goal is to obtain a good bound quickly. Often there is no need to compute the precise value of a relaxation. On some occasions one is satisfied with a reasonable quality guarantee for an available feasible solution, in other cases faster progress can be achieved by improving the relaxation itself, *e.g.*, by means of cutting planes.

In the following we present some conceptual ideas, how the spectral bundle method could be combined with a cutting plane approach. In particular, we discuss possibilities for separating inequalities (Section 6.2.1) and restarting (Section 6.2.2). In order to illustrate some of these ideas we also present tentative implementations for max-cut (Section 6.2.3) and max k-cut (Section 6.2.4) and include preliminary computational results.

We first recall the main steps of the spectral bundle method.

In iteration k the spectral bundle method with bounds (Algorithm 5.4.4) solves a small quadratic semidefinite program of the form (see (QSP_η) of Lemma 5.4.1)

$$\max_{W \in \widehat{\mathcal{W}}^k} \langle C, W \rangle + \left\langle b - \hat{\eta} - \mathcal{A}W, \hat{y}^k \right\rangle - \frac{1}{2u} \left\| b - \hat{\eta} - \mathcal{A}W \right\|^2 . \tag{6.6}$$

The variable W corresponds to the primal matrix variable X of a primal maximization problem whose feasible matrices X all have the same trace; without loss of generality we assume $\operatorname{tr} X = 1$. The Lagrange multipliers $\hat{y}^k$ are the dual variables. $\hat{\eta}$ may be interpreted as a vector of primal slack variables. The set $\widehat{\mathcal{W}}^k$ is a small convex subset of $\{X \succeq 0 : \operatorname{tr}(X) = 1\}$ (see (5.12)),

$$\widehat{\mathcal{W}}^k = \left\{ P_k V P_k^T + \alpha \overline{W}_k : \operatorname{tr} V + \alpha = 1, V \in S^+_{r_k}, \alpha \geq 0 \right\},$$

with orthonormal $P_k \in M_{n,r_k}$ and $\overline{W}_k \succeq 0$ satisfying $\mathrm{tr}(\overline{W}_k) = 1$. The columns of the matrix P_k are called the *bundle*, the matrix $\overline{W}_k$ is referred to as the *aggregate*, and the number r_k of columns of P_k is the *bundle size*. In general, the set $\widehat{\mathcal{W}}^k$ does not contain feasible matrices of the original problem. The quadratic term $\|b - \hat{\eta} - \mathcal{A}W\|^2$ in (6.6) penalizes deviations from feasibility and the weight u controls the influence of this penalty. The optimizer

$$W^{k+1} = P_k V_* P_k^T + \alpha_* \overline{W}_k \in \widehat{\mathcal{W}}^k,$$

of (6.6) gives rise to a new test point y^{k+1} for the cost function $\lambda_{\max}(C - \mathcal{A}^T y) + b^T y$. If the test point yields "sufficient" improvement, the algorithm sets $\hat{y}^{k+1} = y^{k+1}$; this is called a *descent step*. Otherwise it continues with $\hat{y}^{k+1} = \hat{y}^k$, a *null step*, but ensures long term progress in the update to P_{k+1}, $\overline{W}_{k+1}$, and $\hat{\eta}$. In this update, at least one eigenvector of $\lambda_{\max}(C - \mathcal{A}^T y^{k+1})$ (or a sufficiently good approximation to it, see Section 5.6) is incorporated in the new bundle, while some old bundle information may be removed by aggregating it in $\overline{W}_{k+1}$ (Proposition 5.2.3). It is important to note, that the algorithm does not need $\overline{W}_k$ itself, but only the scalar $\langle C, \overline{W}_k \rangle$ and the vector $\mathcal{A}\overline{W}_k \in \mathbb{R}^m$, that can be updated without forming $\overline{W}_k$ explicitly (see Section 5.5).

6.2.1 Separation of Inequalities

The spectral bundle method is a pure dual approach for possibly bounded $y \in \mathbb{R}^m$. Thus, a feasible primal matrix is not available. However, under mild regularity assumptions, the spectral bundle method implicitly generates a sequence of matrices converging to a primal optimal solution. All accumulation points of the subsequence of the W^k that give rise to descent steps are optimal solutions of the primal maximization problem (see Theorem 5.3.8).

In the following we drop all iteration indices and denote the optimal solution of (6.6) by $W = PVP^T + \alpha\overline{W}$. If its deviation from feasibility $\|\mathcal{A}W + \eta - b\|$ is reasonably small, then W may be regarded as an acceptable approximation of the primal optimal solution. Although P, V, and α are easily stored explicitly, this may not always be possible for $\overline{W} \in S_n^+$ if n is large. Depending on what information is available with respect to $\overline{W}$, the matrix W offers different possibilities to employ separation procedures.

Separation if $\overline{W}$ is available

If all elements of the matrix $\overline{W}$ have been stored, then W is available in full, as well. The same separation routines may be used as in primal-dual interior point methods.

Separation with respect to PVP^T

If $\overline{W}$ is too big to be formed explicitly, and only $\mathcal{A}\overline{W}$ and $\langle C, \overline{W} \rangle$ have been stored, then it is not possible to reconstruct W completely. If, however, α is relatively small, then PVP^T holds sufficient information to be useful for separation. Sometimes the special structure of PVP^T may even come in handy in the design of efficient separation routines. Indeed, let $B = PV^{\frac{1}{2}} \in M_{n,r}$ then BB^T is a low rank Gram approximation of W. In many relaxations row i of B may be interpreted as a vector labeling of the i-th discrete variable (see Section 3.4.2). These vectors may be used to search for violated constraints over n low dimensional vectors geometrically rather than over $\binom{n+1}{2}$ real variables. An example of this approach will be given for max-cut below.

Separation with respect to the support

For combinatorial optimization problems defined on sparse graphs or by sparse cost matrices, our matrix notation is somewhat misleading in that it suggests that the problem is defined over $\binom{n+1}{2}$ variables. In fact, the problem definition only requires the variables corresponding to the support of the cost function. A natural object to study in the design of relaxations is therefore the convex hull of the feasible integral solutions in the support space. Since this yields typically the smallest integer programming formulations, this is the prevalent approach in polyhedral combinatorics and

there is a vast literature on polyhedral investigations for various support structures. We refer to Deza and Laurent [1997] for a survey on polyhedral results for max-cut and max k-cut polytopes.

We can follow a similar approach in the spectral bundle method by storing, in addition to $\mathcal{A}\overline{W}$ and $\langle C, \overline{W}\rangle$, the same elements of $\overline{W}$ that are in the support of C. In this case, W can be reconstructed on the support of C and all separation routines developed for the respective polytopes can also be applied to this solution. In addition, this approach has the advantage that the separation routines generate cutting planes whose support is again restricted to the support of C. So the sparsity pattern of the matrix $C - \mathcal{A}^T y$ will not change. This may help to exploit structure in the eigenvalue routine.

6.2.2 Warm-Start

Adding and deleting cutting planes in the primal problem corresponds to adding and deleting variables in the dual problem. For general bundle methods it is unfavorable to increase the problem dimension, because usually old subgradients cannot be updated in the new coordinates. The subgradient model of the function is lost.

The situation is much better in the case of the spectral bundle method. Here, a matrix $W \in \widehat{\mathcal{W}}$ and a slack vector η give rise to the linear minorant $\langle C, W\rangle + \langle b - \eta - \mathcal{A}W, \cdot\rangle$ of the objective function. The pointwise maximum over all these linear functions for $W \in \widehat{\mathcal{W}}$ forms the model function minorizing the objective function (see $f_{\widehat{\mathcal{W}},\eta}$ (5.42)). If $\overline{W}$ of $\widehat{\mathcal{W}}$ is stored explicitly, or at least on the support of the new inequalities, then the entire model information is independent of changes in the dimension of y. On the other hand, if instead of $\overline{W}$ only $\mathcal{A}\overline{W}$ and $\langle C, \overline{W}\rangle$ are stored as suggested in Remark 5.5.1, 2, then at least the part of the model formed by P is preserved.

The convergence analysis of the spectral bundle method demonstrates, that one has to be careful with changes in the model during consecutive null steps. But after a descent step, the method may start with the new point from scratch without endangering convergence (cf. Remark 5.3.7).

Therefore, descent steps are the appropriate place to call separation routines and to add new variables. By setting the new y variables to zero, the current objective value remains unchanged, *i.e.*, the algorithm continues exactly from the point where it stopped, except that some additional freedom has been added in the cost function.

Inactive inequalities, that are currently not tight with respect to the primal approximation, are characterized by the value of the dual variable being zero, say $y_i = 0$, and the corresponding Lagrange multiplier (or primal slack variable, see Remark 5.4.5) being greater than zero, $\eta_i > 0$ (see Lemma 5.4.2). They contribute neither to the current matrix $C - \mathcal{A}^T y$ nor to the objective value $\lambda_{\max}(C - \mathcal{A}^T y) + b^T y$ and may, therefore, be deleted without further consideration. This causes no changes in the cutting plane model, except that the corresponding coordinates of $\mathcal{A}\overline{W}$ have to be eliminated.

6.2.3 Max-Cut

Reformulating the dual of the basic semidefinite max-cut relaxation (SMC) (see page 28) as an eigenvalue optimization problem yields (compare this to (3.11) and Lemma 3.1.11)

$$\min_{y \in \mathbb{R}^n} n\lambda_{\max}(C - \mathrm{Diag}(y)) + e^T y.$$

It seems reasonable to choose, as a starting point, the diagonal entries of $C - \mathrm{Diag}(y)$ so that all diagonal elements have the same value (for max-cut problems on random graphs this is motivated by Theorem 3.1.13). Indeed, we may assume, without loss of generality, that C itself has this property, because changes in the diagonal entries of C that sum up to zero do not influence the value of the primal problem. Under this assumption we start with $y^0 = 0$.

Our numerical experiments are performed on sparse weighted graphs that were generated with `rudy`, a platform independent graph generator written by Giovanni Rinaldi; the arguments corresponding to the graphs are given in Appendix C.

Table 6.6: Comparison of the primal-dual interior point (PDIP) and the spectral bundle (SB) code for (SMC) applied to some sparse weighted graphs on 800 nodes. *sol* gives the computed bounds (dual objective value for interior points and objective value of spectral bundle) and *time* displays the computation time.

	PDIP-sol	PDIP-time	SB-sol	SB-time
G_1	12083.20	44:15	12083.287	1:25
G_2	12089.43	44:38	12089.518	1:14
G_3	12084.33	48:18	12084.407	2:45
G_4	12111.45	46:58	12111.500	1:03
G_5	12099.89	46:48	12099.981	1:06
G_6	2656.160	47:24	2656.1742	1:03
G_7	2489.263	49:12	2489.2829	55
G_8	2506.934	45:55	2506.9537	1:19
G_9	2528.733	48:29	2528.7462	1:03
G_{10}	2485.063	47:14	2485.0825	1:31
G_{11}	629.1648	49:44	629.18528	3:41
G_{12}	623.8744	52:41	623.88997	1:41
G_{13}	647.1365	54:50	647.14802	1:10
G_{14}	3191.567	1:08:49	3191.6093	1:07
G_{15}	3171.558	1:17:36	3171.5907	1:31
G_{16}	3175.017	1:12:56	3175.0431	1:19
G_{17}	3171.327	1:12:30	3171.3616	1:02
G_{18}	1166.010	1:36:34	1166.0343	1:11
G_{19}	1082.010	1:41:48	1082.0216	1:03
G_{20}	1111.393	1:44:17	1111.4008	1:18
G_{21}	1104.284	1:45:03	1104.2903	1:14

Table 6.6 compares the computation times of the primal-dual interior point code of Section 6.1.1 and the spectral bundle code for some weighted graphs on 800 nodes. Whereas the primal-dual interior point code needs roughly from three quarters up to two hours, the spectral bundle method requires one to four minutes. Recently, a pure dual interior point code has been proposed by Benson, Ye, and Zhang [1998] that exploits the sparsity of the dual slack matrix Z in forming the system matrix and in the line search. For the examples of Table 6.6 and Table 6.8 below they report comparable results. Unfortunately, the system matrix remains dense in their approach, in general; so the number of constraints is the limiting factor as in primal-dual interior point methods.

Table 6.7 gives more details on the numerical results of the spectral bundle code for the examples of Table 6.7, and Table 6.8 list some examples of graphs on 2000 nodes. In all these examples we set the termination criterion to $\varepsilon = 10^{-5}$ and work with a bundle size of at most 25 columns in P, where up to 20 columns are kept from the last iteration, and 5 columns are filled with new Lanczos vectors (see the description of the bundle update on page 79). In all examples mentioned so far, the bundle is large enough to keep the coefficient α of the aggregate matrix $\overline{W}$ at negligible small values. Indeed, the high iteration numbers for the grid graphs G_{11}-G_{13} and G_{32}-G_{34} are not a consequence of the bundle size (the full bundle size was never reached), but seem to be due to the flatness of the particular cost functions.

Since the computation of the cost coefficients is increasingly time consuming for larger n, m, and a larger bundle size (the latter also governs the solution time of the quadratic semidefinite subproblem), it may well be worth to reduce the bundle size for larger instances, if the eigenvalue computation is comparatively cheap. We illustrate this in Figure 6.1 for a graph on 10,000 nodes. Even though the plot only shows descent steps, it should be clear that the number of iterations increases significantly as the bundle size decreases. The plot nicely illustrates the strong tailing off effect which is even more articulate for small bundle sizes. Indeed, one hour of computation time seems to yield sufficiently accurate results for most problems up to this size. See Table 6.9 for examples computed with a bundle size of 12 with respect to graphs on 5000 to 10000 nodes

Table 6.7: The spectral bundle code applied to the max-cut relaxation (SMC) for sparse weighted graphs on 800 and 2000 nodes. *Instance* lists the name of the graph described in Appendix C, f^* shows what we regard as the true optimum, *rel_acc* is the relative accuracy of the value at termination with respect to f^* in percent ($\varepsilon = 10^{-5}$ is set as termination criterion), *time* gives the computation time in minutes:seconds, $\lambda\_\%$ is the percentage of time spent in the eigenvalue computation, k displays the number of iterations (including null steps), *descent* lists the number of descent steps, $\|\nabla \bar{f}_+\|$ gives the norm of the final subgradient of the model.

Instance	f^*	rel_acc	time	$\lambda\_\%$	k	descent	$\|\nabla \bar{f}_+\|$
G_1	12083.19	$8.0{\cdot}10^{-6}$	1:25	64	28	20	0.12
G_2	12089.42	$8.3{\cdot}10^{-6}$	1:14	63	27	18	0.16
G_3	12084.33	$6.4{\cdot}10^{-6}$	2:45	83	27	18	0.15
G_4	12111.44	$5.0{\cdot}10^{-6}$	1:03	60	23	16	0.14
G_5	12099.88	$8.1{\cdot}10^{-6}$	1:06	69	25	18	0.17
G_6	2656.157	$6.3{\cdot}10^{-6}$	1:03	54	28	19	0.06
G_7	2489.260	$9.0{\cdot}10^{-6}$	55	60	26	18	0.08
G_8	2506.932	$8.5{\cdot}10^{-6}$	1:19	64	31	22	0.07
G_9	2528.730	$6.4{\cdot}10^{-6}$	1:03	65	25	20	0.08
G_{10}	2485.061	$8.4{\cdot}10^{-6}$	1:31	63	36	25	0.08
G_{11}	629.1645	$3.3{\cdot}10^{-5}$	3:41	83	221	74	0.32
G_{12}	623.8742	$2.5{\cdot}10^{-5}$	1:41	75	102	50	0.28
G_{13}	647.1360	$1.9{\cdot}10^{-5}$	1:10	72	92	42	0.24
G_{14}	3191.562	$1.5{\cdot}10^{-5}$	1:07	54	42	27	0.26
G_{15}	3171.557	$1.0{\cdot}10^{-5}$	1:31	55	52	31	0.24
G_{16}	3175.017	$8.1{\cdot}10^{-6}$	1:19	47	49	27	0.15
G_{17}	3171.325	$1.2{\cdot}10^{-5}$	1:02	56	42	25	0.25
G_{18}	1166.009	$2.1{\cdot}10^{-5}$	1:11	65	52	32	0.15
G_{19}	1082.010	$1.1{\cdot}10^{-5}$	1:03	68	53	29	0.08
G_{20}	1111.392	$8.1{\cdot}10^{-6}$	1:18	75	63	31	0.07
G_{21}	1104.283	$6.4{\cdot}10^{-6}$	1:14	71	56	29	0.06

Table 6.8: The spectral bundle code applied to the max-cut relaxation (SMC) for sparse weighted graphs on 2000 nodes. The columns are described in Table 6.7.

Instance	f^*	rel_acc	time	$\lambda\_\%$	k	descent	$\|\nabla \bar{f}_+\|$
G_{22}	14135.94	$5.7{\cdot}10^{-6}$	4:43	40	38	23	0.20
G_{23}	14145.50	$9.0{\cdot}10^{-6}$	4:12	43	39	23	0.25
G_{24}	14140.85	$7.4{\cdot}10^{-6}$	4:25	41	38	28	0.25
G_{25}	14144.24	$7.0{\cdot}10^{-6}$	3:49	39	34	22	0.27
G_{26}	14132.86	$7.0{\cdot}10^{-6}$	4:09	41	35	24	0.27
G_{27}	4141.658	$6.8{\cdot}10^{-6}$	4:37	44	37	23	0.15
G_{28}	4100.788	$6.1{\cdot}10^{-6}$	7:12	41	43	28	0.12
G_{29}	4208.887	$7.9{\cdot}10^{-6}$	3:52	40	38	25	0.17
G_{30}	4215.380	$8.9{\cdot}10^{-6}$	4:05	39	39	22	0.14
G_{31}	4116.680	$9.0{\cdot}10^{-6}$	5:27	40	42	26	0.14
G_{32}	1567.638	$2.3{\cdot}10^{-5}$	15:10	76	285	81	0.42
G_{33}	1544.312	$7.7{\cdot}10^{-5}$	13:24	78	227	105	0.78
G_{34}	1546.687	$4.5{\cdot}10^{-5}$	8:13	75	166	78	0.59
G_{35}	8014.738	$1.6{\cdot}10^{-5}$	11:28	58	80	44	0.43
G_{36}	8005.958	$7.9{\cdot}10^{-6}$	14:37	53	90	38	0.21
G_{37}	8018.620	$1.9{\cdot}10^{-5}$	13:43	57	94	49	0.44
G_{38}	8014.967	$2.3{\cdot}10^{-5}$	12:28	62	80	46	0.44
G_{39}	2877.645	$1.1{\cdot}10^{-5}$	10:33	67	94	40	0.14
G_{40}	2864.787	$2.1{\cdot}10^{-5}$	9:34	69	99	42	0.16
G_{41}	2865.218	$2.1{\cdot}10^{-5}$	11:22	72	86	52	0.20
G_{42}	2946.250	$2.9{\cdot}10^{-5}$	9:34	65	79	47	0.22

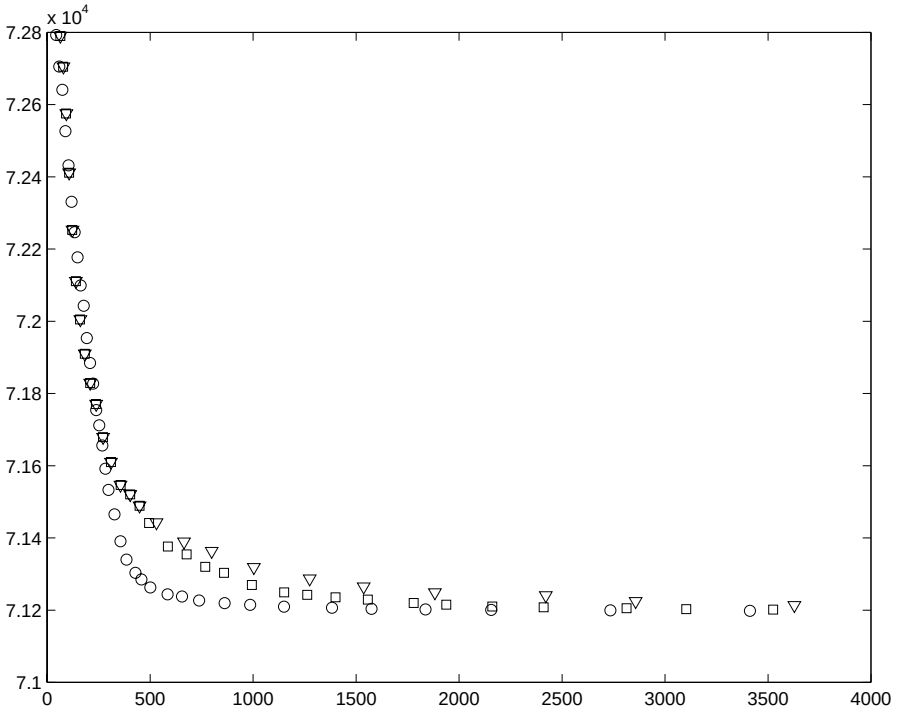

Figure 6.1 The values of the descent steps of the spectral bundle method are plotted versus the time in seconds for bundle sizes 45 (marked by $\triangledown$), 25 ($\square$), and 5 ($\circ$). In the example, the semidefinite relaxation (SMC) of max-cut is solved for a graph on 10,000 nodes. The relative precision reached after one hour is roughly 10^{-4}.

(these graphs were first used by S. Benson [personal communication] in a preliminary version of Benson, Ye, and Zhang [2000]).

Usually, there is no need to wait for the convergence of the algorithm. The bound can be improved faster by adding further freedom in the form of primal cutting planes such as the triangle inequalities (3.14). But already for graphs on 800 nodes the separation of triangle inequalities via complete enumeration in $O(n^3)$ is too expensive in comparison to the running time of the spectral bundle method. Instead, we employ the row vectors of $B = PV^{\frac{1}{2}}$ (see Section above) in order to devise an $O(n^2)$ separation heuristic.

By Theorem 5.3.8 the matrix W approximates a primal optimal X, and W itself is approximated by BB^T. Denoting column i of B^T by v_i we may therefore approximate x_{ij} by $v_i^T v_j$. The hypermetric inequalities (see (3.13) and thereafter) allow an intriguing representation in terms of these v_i. We illustrate this for the triangle constraint $x_{ij} + x_{ik} + x_{jk} \geq -1$ (3.14). This inequality is equivalent to $b^T X b \geq 1$ with $b_i = b_j = b_k = 1$ and $b_h = 0$ otherwise. Replacing X by BB^T we obtain $(b^T B)(B^T b) = \|v_i + v_j + v_k\|^2 \geq 1$. Thus, finding a violated triangle inequality corresponds to finding v_i, v_j, and v_k that sum up to a vector of small norm.

In order to find reasonable candidates for such vectors quickly, we employ a simple form of geometric searching (see Preparata and Shamos [1985] for more serious applications of geometric searching). We project the row vectors onto a smaller random subspace, say 5-dimensional (in the implementation the number depends on n), and partition the space symmetrically around zero into five sections per coordinate axis, forming 5^5 cells. Each cell is assigned the list of vectors whose projections fall into this cell. Now we test for any pair of vertices whether the cell matching the projection of the negative of their sum contains a vector, so that all three vectors sum up to almost zero. If the number of cells is selected so that the expected number of vectors assigned to each cell is a small constant in average, then this heuristic runs in roughly $O(n^2)$ time. The heuristic seems to work reasonably well as long as the contribution $\alpha_* \overline{W}$ to W is small, *i.e.*, B

Table 6.9: The spectral bundle code applied to the max-cut relaxation (SMC) for sparse weighted graphs on 5000 to 10000 nodes with a time limit of one hour. The columns are described in Table6.7. In each iteration the bundle consists of 4 aggregated columns and 8 new vectors.

Instance	f^*	rel_acc	time	λ_%	k	descent	$\|\nabla \bar{f}_+\|$
G_{55}	12869.94	$4.1{\cdot}10^{-6}$	30:22	79	279	41	2.00
G_{56}	4760.070	$1.2{\cdot}10^{-5}$	1:00:11	81	523	47	0.72
G_{57}	3884.376	$8.3{\cdot}10^{-5}$	1:00:01	77	714	46	0.45
G_{58}	20136.18	$1.4{\cdot}10^{-4}$	1:00:28	90	230	36	0.77
G_{59}	7312.325	$7.3{\cdot}10^{-4}$	1:00:09	88	286	41	0.61
G_{60}	15222.43	$1.3{\cdot}10^{-5}$	1:00:01	77	424	41	2.36
G_{61}	6828.456	$7.5{\cdot}10^{-6}$	1:00:05	78	404	39	1.70
G_{62}	5600.282	$1.6{\cdot}10^{-4}$	1:00:07	74	557	49	0.80
G_{63}	28244.39	$2.4{\cdot}10^{-4}$	1:01:00	88	188	36	1.50
G_{64}	10465.89	$1.0{\cdot}10^{-3}$	1:00:32	87	196	37	1.56
G_{65}	6214.986	$2.1{\cdot}10^{-4}$	1:00:02	74	485	48	0.77
G_{66}	6967.231	$2.5{\cdot}10^{-4}$	1:00:04	74	429	50	1.17
G_{67}	7744.428	$2.9{\cdot}10^{-4}$	1:00:04	74	376	50	1.34

Table 6.10: The max-cut relaxation (SMC) improved by triangle cutting planes for sparse weighted graphs on 800 nodes. Column *no cuts* gives the same objective value and computation time in minutes:seconds as in Table 6.7. Starting, at every second descent step, the separation routine when the stopping criterion indicates a relative precision of 0.5% yields after one minute the results of column *cuts, 1 min*, and after ten minutes column *cuts, 10 min*.

	no cuts		cuts, 1 min		cuts, 10 min	
Instance	f	time	f	time	f	time
G_1	12083.29	1:25	12074.15	1:05	12043.29	10:07
G_2	12089.52	1:14	12077.12	1:04	12049.14	10:12
G_3	12084.41	2:45	12069.81	1:06	12042.52	10:00
G_4	12111.50	1:03	12105.65	1:01	12067.36	10:12
G_5	12099.98	1:06	12082.89	1:12	12055.14	10:08
G_6	2656.174	1:03	2640.480	1:06	2611.760	10:12
G_7	2489.283	55	2472.192	1:05	2444.222	10:10
G_8	2506.954	1:19	2490.083	1:02	2464.143	10:06
G_9	2528.746	1:03	2512.633	1:04	2486.013	10:03
G_{10}	2485.082	1:31	2473.997	1:06	2447.028	10:06
G_{11}	629.1853	3:41	624.0085	1:03	604.1985	10:00
G_{12}	623.8900	1:41	618.6955	1:01	595.7278	10:05
G_{13}	647.1480	1:10	641.3670	1:02	618.2381	10:07
G_{14}	3191.609	1:07	3190.286	1:02	3177.571	10:10
G_{15}	3171.591	1:31	3169.741	1:07	3158.108	10:11
G_{16}	3175.043	1:19	3175.331	1:03	3163.338	10:07
G_{17}	3171.362	1:02	3173.060	1:02	3157.657	10:04
G_{18}	1166.034	1:11	1157.574	1:07	1138.837	10:11
G_{19}	1082.022	1:03	1076.057	1:03	1057.156	10:07
G_{20}	1111.401	1:18	1105.598	1:00	1086.020	10:06
G_{21}	1104.290	1:14	1101.117	1:05	1080.284	10:02

contains most of the information of W. See Tables 6.10 and 6.11 for some tentative results on the examples of Tables 6.7 and 6.8.

Unfortunately, the heuristic is only successful for graphs with up to, say, 5000 nodes, for two reasons. First, our separation routine is already too slow for larger sizes, and second, the necessary bundle size to represent W in sufficient accuracy by B alone grows too large to be computationally efficient. For graphs beyond 5000 nodes other techniques may be required, *e.g.*, heuristics or exact methods separating cutting planes with respect to the support of the cost matrix.

6.2.4 Max k-Cut

For a given undirected, possibly weighted graph $G = (V, E)$, max k-cut asks for a partition of the vertex set $V = \{1, \ldots, n\}$ into at most k subsets such that the sum of the edge weights of edges in the k-cut (the set of edges whose endpoints belong to distinct sets) is maximized. The semidefinite relaxation of max k-cut has been described in Section 3.4.2, its representation (3.28) differs from (SMC) by an additional constraint $x_{ij} \geq -\frac{1}{k-1}$ for each offdiagonal entry of X,

$$\begin{array}{lll} \max & \langle C, X \rangle & \\ \text{s.t.} & \operatorname{diag}(X) = e & \\ & x_{ij} \geq -\frac{1}{k-1} & \text{for } 1 \leq i < j \leq n \\ & X \succeq 0. & \end{array} \tag{6.7}$$

In the discrete setting, $x_{ij} = 1$ corresponds to i and j being in the same set, and if $x_{ij} = -\frac{1}{k-1}$ then i and j belong to distinct sets. Note, that the number of constraints grows quadratically in n.

An application of max k-cut is the computation of lower bounds on co-channel interference in cellular networks (see, *e.g.*, Borndörfer, Eisenblätter, Grötschel, and Martin [1998]). Given n antennas, k frequencies ($k \in \{50, \ldots, 100\}$), and the amount of interference between pairs of antennas if they are assigned the same frequency (co-channel interference), the task is to assign to each antenna a frequency so that the sum of the co-channel interferences is minimized. We assume that the interference is given by a nonnegative adjacency matrix A corresponding to an undirected weighted graph. For this graph we want to find a partition of the node set (the antennas) into k different sets (the frequencies) so that the sum of the edge weights of edges running between distinct sets is maximized. The appropriate cost matrix for (6.7) is $C = \frac{k-1}{2k}(\operatorname{Diag}(Ae) - A)$ (see Section 3.4.2). Lower bounds on the minimal interference are obtained by subtracting, for a feasible solution of the dual to (6.7), the dual objective value from the sum of all edge weights. Additional constraints arise if certain pairs of antennas must be assigned distinct frequencies because of quality restrictions, *i.e.*, the corresponding edge must be in the k-cut. The latter restrictions are easily incorporated into (3.28) by requiring $x_{ij} = -\frac{1}{k-1}$ for each such pair ij of antennas.

The basic relaxation (3.28) lists $\binom{n}{2}$ constraints, but there is no need to include all of them. Note, that the dual of (3.28) is a valid upper bound on the size of a maximal k-cut for any subset of the constraints $x_{ij} \geq -\frac{1}{k-1}$. In order to guarantee at least the trivial bound (all edges in the cut) we have to include the offdiagonal constraints corresponding to negative elements of the cost matrix. Computational experiments indicate that for our algorithm it is more efficient to start with the diagonal constraints only and to separate $x_{ij} \geq -\frac{1}{k-1}$ successively than to include all constraints at once, even if the final number of active inequality constraints is close to $\binom{n}{2}$.

In our numerical experiments, we compute lower bounds for co-channel interference incurred by frequency plans for GSM-networks. The data is supplied by A. Eisenblätter and originates in realistic planning problems provided by E-Plus Mobilfunk GmbH.

For these problems he was able to find presumably good feasible solutions. Efforts of proving a lower bound by a linear cutting plane approach did not succeed; the lower bounds remained close to zero (A. Eisenblätter [personal communication]).

The number of antennas n ranges between 100 and 4000 and the number of available frequencies k is either 50, 75, or 100. Typically, a single co-channel interference (a single edge) has a weight

between zero and one half, but sometimes it may be larger than one. Depending on the size of the instance (and the number of frequencies), the sum of all edges ranges between 1000 and 50000. The total interference in a good assignment sums up to a value between 0.1 and 10, *i.e.*, almost all edges end up in the k-cut. Therefore, any reasonable bound will have to be computed to high relative precision, which is an unfavorable situation for the spectral bundle method.

Since the matrix sizes range between 100 and 4000, it is possible to store the full aggregate $\overline{W}$. Updating $\overline{W}$ is an n^2 operation and slows down the algorithm for larger n, say $n > 1000$, but in return the separation procedure is efficient and independent of the bundle size. Furthermore, the algorithm can be restarted after the addition or deletion of constraints without loss of quality in the model.

In our current implementation (it is still changing a lot) we start the separation routine if the relative gap between objective and model value is below 1%, and then repeatedly after at least three further descent steps. Each time, we delete all inactive inequalities and select the $20n$ most violated inequalities with respect to the last solution W of the quadratic semidefinite subproblem. The termination criterion is set to a relative precision of 10^{-5} with respect to the objective value of the dual to (6.7).

Table 6.12 presents first results for some of the smaller examples of A. Eisenblätter. The quality of the lower bounds seems to be unmatched as of today. We do not expect that our reported bounds are very accurate estimates of the true optimal solution of relaxation (6.7); the accuracy of the lower bounds given in the table will be one or two digits at most due to the following three reasons. First, the termination criterion works with formulation (6.7), thus the relative precision is determined with respect to the sum of the edge weights. Second, in our experience the termination criterion usually stops somewhat short of the desired relative precision (in this case it may amount to a relative precision of only $5 \cdot 10^{-5}$ with respect to numbers greater than 1000). Finally, the cutting plane algorithm works with respect to an infeasible approximation to the optimal primal solution; typically, several constraints are still not satisfied by this matrix. Considering example FAP10, the last separation before termination detected inequalities violated by 0.02. It is therefore not surprising that the computed bound may stay below the trivial 0 as in FAP10. In view of this it is not possible to judge the true quality of (6.7) for examples FAP10 to FAP12 from our computations. However, the sum of the edge weights minus the lower bounds of Table 6.12 are certainly upper bounds on the dual optimal solution of (6.7).

We interpret the high computation times as a consequence of the interaction of two tailing off effects. The spectral bundle method and the cutting plane method both show a strong tailing off effect as the optimal solution of the respective problem is approached. In principle, it is possible to improve (6.7) with further cutting planes, see Grötschel and Wakabayashi [1989]; Grötschel and Wakabayashi [1990]; Deza, Grötschel, and Laurent [1991]; Chopra and Rao [1993]; Chopra and Rao [1995], but we have not yet experimented with these possibilities. Currently our approach is too slow to be used as a subroutine in a planning tool. Still, the availability of good lower bounds is of importance for judging the quality of faster heuristic approaches.

6.3 Fixing of Variables

The task of solving hard combinatorial problems to optimality leads naturally to *branch and bound* (or *branch and cut*). In the branch and bound approach a bound on the optimal solution is computed by solving a tractable relaxation of the combinatorial problem. If the bound is not tight enough to prove optimality of the best combinatorial solution found so far, then, in the so-called branching step, the problem is split into two or more subproblems by partitioning the space of combinatorial solutions, *e.g.*, by setting a $\{0,1\}$ variable to 0 in one subproblem and to 1 in the other. The same scheme is then applied recursively to the subproblems. The nature of this enumerative approach is best visualized by a, in graph theoretic terms, rooted tree (the *branch and bound tree*) with the root node corresponding to the original combinatorial problem, the arcs corresponding to the decisions, and the nodes corresponding to the subproblems. Subproblems that have not yet been solved are located in the end nodes (*leaves*). The decisions taken along a

Table 6.11: The max-cut relaxation (SMC) improved by triangle cutting planes for sparse weighted graphs on 2000 nodes. Column *no cuts* gives the same objective value and computation time in minutes:seconds as in Table 6.8. Starting, at every second descent step, the separation routine when the stopping criterion indicates a relative precision of 0.5% yields after ten minutes the results of column *cuts, 10 min*, and, after 30 minutes, column *cuts, 30 min*.

	no cuts		cuts, 10 min		cuts, 30 min	
Instance	f	time	f	time	f	time
G_{22}	14136.02	4:43	14121.32	10:05	14098.40	30:19
G_{23}	14145.63	4:12	14128.98	10:04	14105.58	30:12
G_{24}	14140.95	4:25	14129.07	10:34	14103.67	30:17
G_{25}	14144.34	3:49	14125.59	10:11	14102.45	30:17
G_{26}	14132.96	4:09	14114.31	10:07	14094.54	31:05
G_{27}	4141.685	4:37	4119.570	10:40	4103.103	30:01
G_{28}	4100.813	7:12	4085.120	10:11	4066.122	30:23
G_{29}	4208.921	3:52	4185.959	10:34	4166.341	31:05
G_{30}	4215.418	4:05	4192.520	10:34	4176.569	30:01
G_{31}	4116.717	5:27	4095.326	10:06	4079.510	30:18
G_{32}	1567.675	15:10	1558.093	10:38	1527.607	31:17
G_{33}	1544.430	13:24	1528.904	10:06	1502.231	30:01
G_{34}	1546.757	8:13	1524.249	10:12	1496.889	30:21
G_{35}	8014.868	11:28	8014.810	10:12	7995.954	30:05
G_{36}	8006.021	14:37	8005.810	10:05	7990.340	30:16
G_{37}	8018.772	13:43	8019.308	10:02	8001.319	30:18
G_{38}	8015.154	12:28	8013.450	10:08	7998.240	30:12
G_{39}	2877.678	10:33	2867.089	10:07	2841.002	30:18
G_{40}	2864.848	9:34	2852.139	10:04	2829.115	30:17
G_{41}	2865.278	11:22	2848.303	10:02	2821.780	30:05
G_{42}	2946.336	9:34	2929.661	10:04	2901.910	30:01

Table 6.12: Frequency assignment examples for minimizing co-channel interference. n is the number of antennas, k the number of frequencies, $\sum$ refers to the sum over all weights, *Interf.* shows the value of the best solution known (computed by A. Eisenblätter), *lbd* gives the lower bound on the minimal amount of interference obtained by the objective value of the relaxation, *time* is listed in hours:minutes:seconds, *# eq.* gives the number of (diagonal and offdiagonal) equality constraints, *# ineq.* refers to the number of inequality constraints $x_{ij \geq -1}$ in use at termination, *# it.* displays the number of function evaluations. We used a bundle of five new vectors per iteration (no aggregate columns), and a termination criterion of $\varepsilon = 10^{-5}$.

Instance	n	k	$\sum$	Interf.	lbd	time	# eq.	#ineq.	# it
FAP06	93	50	811.073	0.5264	0.4527	3:06:22.02	374	3758	73101
FAP07	98	50	1205.26	2.4857	2.0905	4:19:09.88	712	3599	93737
FAP08	120	50	1309.66	2.8132	2.4260	6:12:23.66	592	5938	81881
FAP09	174	50	2862.86	12.245	10.764	8:49:53.57	1200	11174	69358
FAP10	183	75	2001.45	0.0370	-0.0015	14:02:13.94	725	14794	90642
FAP11	252	75	3123.25	0.1357	0.0073	42:57:55.05	1017	26777	155855
FAP12	369	50	4481.76	0.8074	0.2210	131:05:31.78	2052	33805	326105

path (or *branch*) starting at the root node successively reduce the set of feasible solutions of the subproblems on this path. If the bound on a subproblem indicates that no better combinatorial solution can be found in the subproblems feasible set, then the node may be *fathomed*, *i.e.*, the recursion can be stopped for this branch of the tree. The problem is solved when all leaves of the branches have been fathomed. *Branch and cut* refers to branch and bound approaches where for each subproblem the bounding procedure is a cutting plane algorithm.

In this setting the efficiency of an expensive bound hinges on the tradeoff between the number of branch and bound nodes and the computation time needed for each node. If the bound is derived from the dual of a relaxation and its value is sufficiently close to an integral solution then it is often possible to reduce the number of branch and bound nodes by *fixing* variables, *i.e.*, by proving that better solutions can only be obtained for fixed values of certain variables.

In linear programming, fixing variables using reduced costs is a standard procedure. In order to illustrate this technique, consider the quadratic $\{-1,1\}$-programming problem (3.4),

$$\max_{x\in\{-1,1\}^n} x^T C x$$

with integral cost matrix C and assume that a 'good' integral solution $c^* = \bar{x}^T B \bar{x}$ with $\bar{x} \in \{-1,1\}^n$ is known. Linear relaxations of (3.4) are usually formulated in variables x_{ij}, that represent linearizations of the products $x_i \cdot x_j$. The relaxations typically include the box constraints $-1 \leq x_{ij} \leq 1$. Let the optimal solution of the relaxation have dual objective value $b^T\hat{u}$ where b is the vector of primal right hand sides and $\hat{u}$ the vector of Lagrange multipliers. If the optimal solution $\hat{x}$ of the linear relaxation yields $\hat{x}_{ij} = 1$, say, then we can quickly compute an upper bound for the problem with the additional constraint $x_{ij} = -1$ by exploiting the following observation. In the dual problem the change of the right hand side in the primal constraint $x_{ij} \leq 1 = b_k$ to $x_{ij} \leq -1 = \bar{b}_k$ affects only the cost coefficients, b changes to $\bar{b}$. Thus, the old Lagrange multipliers $\hat{u}$ remain feasible for the new cost coefficients $\bar{b}$ and the new dual objective value is, in this particular case, $\bar{b}^T\hat{u} = b^T\hat{u} - \hat{u}_k$. If this new bound satisfies $b^T\hat{u} - \hat{u}_k < c^* + 1$, then this proves: If there are better integral solutions than c^* in the feasible set then they satisfy $x_{ij} = 1$.

The semidefinite relaxation (SMC) for (3.4) already implies the box-constraints (Lemma 3.1.2). Consequently, they are not included in the relaxation and the corresponding Lagrange multipliers are unknown. Yet, if the optimal solution of the semidefinite relaxation displays some $x_{ij} = 1$, then there must be some corresponding active constraint buried in the semidefiniteness constraint. Indeed, we can associate with each active constraint $x_{ij} \geq -1$ or $x_{ij} \leq 1$ an active constraint $v^T X v \geq 0$ from the set of constraints ensuring the positive semidefiniteness of X as follows. Let $|x_{ij}| = 1$ for some $i \neq j$ in the optimal solution (X^*, u^*, Z^*) of the current relaxation. Then the vector $v \in \mathbb{R}^n$ with

$$v_k = \begin{cases} 1 & k = i \\ -\operatorname{sgn}(x_{ij}) & k = j \\ 0 & \text{otherwise,} \end{cases} \tag{6.8}$$

is in the null space of X^*. Although the Lagrange multiplier corresponding to the constraint $\langle vv^T, X\rangle \geq 0$ is not directly available it might be possible to construct one from information contained in the Lagrange multiplier to the semidefiniteness constraint on X, this multiplier being the dual slack matrix Z^*.

In this section we illustrate a practical method for extracting this information. We will present two algorithms developed in Helmberg [2000]. The first is based on a line search with respect to the maximum eigenvalue function. It is suitable for general inequalities and might also be useful if the dual matrix variable is a huge structured matrix as may be the case in the spectral bundle method or in the dual interior point method of Benson, Ye, and Zhang [2000]. The second algorithm is tailored to box constraints on offdiagonal variables and assumes the availability of the eigenvalue decomposition of the dual matrix variable. It is worth noting that the discussion is completely independent of the actual algorithm used to solve the semidefinite relaxation.

We start by investigating the general case of semidefinite programming.

6.3.1 The Theoretical Framework

Consider a standard primal-dual pair of semidefinite programs,

$$\begin{array}{cccc} & \min & \langle C, X\rangle & \\ (\mathrm{P}) & \text{s.t.} & \mathcal{A}X = b & \\ & & X \succeq 0 & \end{array} \qquad \begin{array}{cccc} & \max & \langle b, y\rangle & \\ (\mathrm{D}) & \text{s.t.} & \mathcal{A}^T y + Z = C & \\ & & Z \succeq 0. & \end{array}$$

We examine possibilities to extract duality information for equality or inequality constraints that are not explicitly given in the problem description. Assume that optimal solutions X^* of (P) and (y^*, Z^*) of (D) are given and denote the optimal objective value by $p^* = \langle C, X^*\rangle = \langle b, y^*\rangle$ (in the last equation we tacitly assume some constraint qualification to hold).

We are interested in the following question: How much does the optimal value of (P) increase if an additional constraint $\langle A_0, X\rangle = b_0$ is added to the problem? We would like to bound this quantity without actually computing the optimal solution of the new problem.

Let y_0 denote the new dual variable associated with the new constraint. The corresponding primal-dual pair reads

$$\begin{array}{cccc} & \min & \langle C, X\rangle & \\ & \text{s.t.} & \langle A_0, X\rangle = b_0 & \\ (\mathrm{P}_0) & & \mathcal{A}X = b & \\ & & X \succeq 0 & \end{array} \qquad \begin{array}{cccc} & \max & b_0 y_0 + \langle b, y\rangle & \\ (\mathrm{D}_0) & \text{s.t.} & y_0 A_0 + \mathcal{A}^T y + Z = C & \\ & & Z \succeq 0. & \end{array}$$

Computing the optimal solution is as hard as solving the original problem. However, we do already know a "good" dual feasible solution for (D_0), namely $(y_0 = 0, y^*, Z^*)$. To improve this solution with reasonable effort we restrict ourselves to a line search along an ascent direction $(\Delta y_0, \Delta y, \Delta Z)$ with

$$\begin{array}{rcll} b_0 \Delta y_0 + \langle b, \Delta y\rangle & > & 0 & \\ \Delta y_0 A_0 + \mathcal{A}^T \Delta y + \Delta Z & = & 0 & \\ Z^* + t\Delta Z & \succeq & 0 & \text{for some } t \geq 0. \end{array}$$

To determine the best search direction is again as difficult as the problem itself. The choice of a good direction will depend on our understanding of the problem at hand.

Having fixed an ascent direction $(\Delta y_0, \Delta y, \Delta Z)$ it remains to compute the maximal step size t such that $Z^* + t\Delta Z$ is still positive semidefinite, because the objective function is linear. With

$$\begin{array}{rcl} S & = & C - \mathcal{A}^T y^* = Z^* \succeq 0 \\ B & = & \Delta y_0 A_0 + \mathcal{A}^T \Delta y \end{array}$$

the problem reduces to

$$(LS) \quad \max\ t \ \text{ s.t. } S - tB \succeq 0.$$

There is no need to include the constraint $t \geq 0$ since $S \succeq 0$ implies that the optimal solution t^* is nonnegative. Problems of this form appear as matrix pencils in the literature (see, *e.g.*, Golub and van Loan [1989]). Indeed, the optimal t can be computed explicitly.

Let $P\Lambda_S P^T = S$ denote an eigenvalue decomposition of S with P an orthonormal matrix and Λ_S a diagonal matrix having the eigenvalues $\lambda_1(S) \geq \ldots \geq \lambda_n(S)$ on its diagonal in this order. Then $S - tB \succeq 0$ is equivalent to

$$\Lambda_S - tP^T B P \succeq 0.$$

If the rank of S is k, then $\lambda_i(S) = 0$ for $i = k+1, \ldots, n$. Scaling the equation above by $D = \mathrm{Diag}(\lambda_1(S)^{-\frac{1}{2}}, \ldots, \lambda_k(S)^{-\frac{1}{2}}, 1, \ldots, 1)$ we obtain

$$\begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix} - tDP^T BPD \succeq 0.$$

Assuming that $t > 0$ exists, we divide by t and impose the same block structure on DP^TBPD,

$$\begin{bmatrix} \frac{1}{t}I_k & 0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} B_{11} & B_{12} \\ B_{12}^T & B_{22} \end{bmatrix} \succeq 0,$$

with $B_{11} \in M_k$, $B_{22} \in M_{n-k}$, $B_{12} \in M_{k,n-k}$ such that

$$\begin{bmatrix} B_{11} & B_{12} \\ B_{12}^T & B_{22} \end{bmatrix} = DP^TBPD.$$

In the case where B_{12} and B_{22} are both zero, $\frac{1}{t} \geq \lambda_{\max}(B_{11})$ is the best choice (for $S \succ 0$ this specializes to $\frac{1}{t} = \lambda_{\max}(S^{-1}B)$. This is sometimes used in interior point methods to replace the line search for $X+\alpha\Delta X \succeq 0$, see Kojima, Shindoh, and Hara [1994]). Note, that for $\lambda_{\max}(B_{11}) \leq 0$ the problem is unbounded. If $-B_{22}$ is non-zero and not positive semidefinite, then $t = 0$ is the optimal solution. If $-B_{22}$ is positive semidefinite with rank h, we can apply a similar sequence of steps to obtain a condition

$$\begin{bmatrix} \frac{1}{t}I_k - B_{11} & \bar{B}_{12} & \bar{B}_{13} \\ \bar{B}_{12}^T & I_h & 0 \\ \bar{B}_{13}^T & 0 & 0 \end{bmatrix} \succeq 0.$$

If $\bar{B}_{13}$ is non-zero then again t must be zero. Otherwise we can apply the Schur complement Theorem 1.1.9 to obtain the condition

$$\frac{1}{t}I_k - B_{11} \succeq \bar{B}_{12}\bar{B}_{12}^T.$$

This yields $\frac{1}{t} \geq \lambda_{\max}(B_{11} + \bar{B}_{12}\bar{B}_{12}^T)$.

We specialize this general procedure to a case of particular importance in semidefinite programming. For the purpose of explanation assume that X^* and (y^*, Z^*) are a strictly complementary pair of optimal solutions, *i.e.*, $\operatorname{rank}(X^*) + \operatorname{rank}(Z^*) = n$ (these do not necessarily exist, see Alizadeh, Haeberly, and Overton [1997]). Furthermore let A_0 be a dyadic product vv^T for some $v \in \mathbb{R}^n$ with $\langle vv^T, X^* \rangle = 0$, *i.e.*, v is in the null space of X^*. The matrix vv^T may be interpreted as one of the active constraints ensuring the positive definiteness of X. The right hand side b_0 of the new constraint must be greater than zero, otherwise there is no feasible primal solution for the new problem. As ascent direction we choose $\Delta y_0 = 1$ and $\Delta y = 0$. This yields the following line search problem,

$$\max\ t \ \text{ s.t. } Z^* - tvv^T \succeq 0.$$

Because X^* and Z^* are strictly complementary solutions and v is in the null space of X^* we conclude that v lies in the span of the eigenvectors associated with non-zero eigenvalues of Z^*. Assume that $\operatorname{rank}(Z^*) = k$ and let $P\Lambda_{Z^*}P = Z^*$ denote the eigenvalue decomposition of Z^* with $P \in M_{n,k}$, $P^TP = I_k$, and the spectrum of non-zero eigenvalues $\Lambda_{Z^*} \in S_k$. Then the maximal t is given by

$$t^* = \frac{1}{v^TP\Lambda_{Z^*}^{-1}P^Tv}. \tag{6.9}$$

If in particular v happens to be an eigenvector of Z^* then t^* is the corresponding eigenvalue of Z^*.

This procedure can be extended to the case where X^* and Z^* are not strictly complementary. For any vector v in the null space of X^* but not in the span of the non-zero eigenvectors of Z^* the optimal t is zero.

6.3.2 A Practical Algorithm

With respect to the semidefinite relaxation (SMC), formula (6.9) suggests a convenient procedure for constructing Lagrange multipliers for constraints of the form (6.8). Denote by (X^*, Z^*, u^*) an optimal primal-dual solution to (SMC). For some $i \neq j$ with $|x_{ij}^*| = 1$ in this solution, let v be the corresponding vector (6.8). Assuming that the eigenvalue decomposition of Z^* into

$P\Lambda_{Z^*}P^T$ is available ($k = \operatorname{rank}(Z^*)$, $P \in M_{n,k}$, $P^TP = I_k$), it is easy to check whether v is in the span of the eigenvectors P. If it is not, then $t^* = 0$, otherwise $t^* = -1/(v^TP\Lambda_{Z^*}^{-1}P^Tv)$ is the best Lagrange multiplier for the constraint $v^TXv = 0$ obtainable from $Z^* = \operatorname{Diag}(u^*) - C$ (since (SMC) is a maximization problem, the optimal t is now nonpositive). Forcing the opposite assignment $x_{ij} = -\operatorname{sgn}(x_{ij}^*)$ can be modeled by changing the right hand side of the (currently active) constraint $v^TXv = 0$ to $v^TXv = 4$ in the current relaxation. The optimal value of the relaxation corresponding to this opposite assignment is therefore at most $e^Tu^* + 4t^*$.

In theory this yields a very efficient algorithm for checking several pairs $\{i, j\}$. The eigenvalue decomposition has to be computed only once for all pairs, the evaluation for a single pair requires roughly $O(nk)$ arithmetic operations.

In implementing this approach several difficulties are encountered. Indeed, we cannot expect any real world algorithm to deliver a true optimal solution (X^*, u^*, Z^*) of (PMC) for arbitrary cost matrices. For a computed solution $(\hat{X}, \hat{u}, \hat{Z})$ both, $\hat{X}$ and $\hat{Z}$, will be (rather ill conditioned) full rank matrices. Even in the case where the gap $\langle \hat{X}, \hat{Z}\rangle$ between the primal and dual solutions is almost zero, it is difficult to decide which of the eigenvalues of $\hat{X}$ and $\hat{Z}$ will eventually converge to zero. The space spanned by the eigenvectors corresponding to the "non-zero" eigenvalues of $\hat{X}$ and $\hat{Z}$ may still differ substantially from the true eigenspaces of X^* and Z^*. The vectors v of (6.8) will neither be contained in the null space of $\hat{X}$ nor in the space spanned by the "non-zero" eigenvectors of $\hat{Z}$ because no $|x_{ij}|$ will be exactly one. As a result the line search will allow only a very short step and the approach fails.

In the case of (SMC), however, there is an obvious way to get around these difficulties. It is worth noting that the framework can be applied when there are additional primal constraints, but as these have no influence on the considerations to follow, we ignore them here.

Within the branch and bound scenario let $(\hat{X}, \hat{u}, \hat{Z})$ be the solution computed for the relaxation of the current branch and bound node yielding the upper bound $e^T\hat{u}$. Let c^* denote the lower bound on the optimal solution of (MC). How much does the upper bound improve if we add a branching constraint $\langle A_0, \hat{X}\rangle = b_0$ (w.l.o.g. assume $b_0 > 0$) to the current relaxation? We denote the Lagrange multiplier for the new constraint by u_0. We would like to compute an upper bound, ideally smaller than c^*, for the problem

$$\begin{array}{ll} \min & b_0u_0 + e^Tu \\ \text{s.t.} & Z = u_0A_0 + \operatorname{Diag}(u) - C \\ & Z \succeq 0. \end{array}$$

Consider the situation of setting u_0 to some (negative) value required for achieving $b_0u_0 + e^T\hat{u} < c^*$. If $\hat{Z} + u_0A_0$ is still positive semidefinite then we are done . If not, we add $-\lambda_{\min}(\hat{Z} + u_0A_0)e$ to $\hat{u}$, giving

$$u = \hat{u} - \lambda_{\min}(\hat{Z} + u_0A_0)e.$$

This worsens the original bound of $e^T\hat{u}$ by $-n\lambda_{\min}$, but the new Z is feasible. Thus we seek a u_0 such that

$$4u_0 + e^T\hat{u} - n\lambda_{\min}(\hat{Z} + u_0vv^T) < c^*.$$

Summing up we specialize the semidefinite program above to

$$\min_{u_0 \in \mathbb{R}} b_0u_0 + e^T\hat{u} - n\lambda_{\min}(u_0A_0 + \hat{Z}). \tag{6.10}$$

We first outline a general method for arriving at good estimates for u_0 and present a specialized algorithm for constraints of the form $x_{ij} = -\operatorname{sgn}(\hat{x}_{ij})$ afterwards.

The minimal eigenvalue is a concave function, so (6.10) is a convex optimization problem. The objective function is differentiable if and only if the minimal eigenvalue has multiplicity one. In this case the gradient is determined by

$$\nabla_{u_0}(b_0u_0 - n\lambda_{\min}(u_0A_0 + \hat{Z})) = b_0 - n\,q(u_0)^TA_0q(u_0)$$

with $q(u_0)$ denoting the (normalized) eigenvector to the minimal eigenvalue of $u_0A_0 + \hat{Z}$. As explained above, it can be expected that $\hat{Z}$ has eigenvalue zero with high multiplicity. Therefore,

the function is not differentiable for $u_0 = 0$ and it is hard to guess the initial descent. It seems appropriate to choose the starting value u_0 with respect to the gap $c^* - e^T\hat{u}$, *e.g.*, $u_0 = (c^* - e^T\hat{u}) \cdot 1.2/b_0$. For reasonably large $|u_0|$ the minimal eigenvalue will be well separated and we can use the gradient to decide whether it is worth increasing $|u_0|$ even further or not. If the value of the gradient implies that it might be possible to beat c^* we do another step slightly overestimating the remaining gap. We repeat this procedure at most three times.

The computation of the gradient requires the computation of the eigenvector of the minimal eigenvalue of $u_0 A_0 + \hat{Z}$. Extremal eigenvalues and eigenvectors are best determined via iterative methods such as the Lanczos method, which can exploit problem structure (see Section 5.6 and references therein). In particular these methods are very fast if a good starting vector is known. For the first computation we suggest to choose an eigenvector associated with the maximum eigenvalue of A_0, for all further iterations the last eigenvector computed is the natural choice. We expect that this approach is efficiently applicable even in the case when $\hat{Z}$ is a huge structured matrix as may be the case in the spectral bundle method or in Benson, Ye, and Zhang [2000].

In the following we assume that the constraint is of the form $x_{ij} = 1$ (the argument for $x_{ij} = -1$ is analogous) and that the eigenvalue decomposition $\hat{Z} = P\Lambda P^T$ with $P^T P = I_n$ can be computed efficiently, *i.e.*, the dimension of $\hat{Z}$ is not above 1000, say. In practice, the representation $\langle E_{ij}, X\rangle = 2$ (the matrix E_{ij} has a 1 in positions ij and ji and is zero otherwise) of the constraint $x_{ij} = 1$ has proved superior to the representation $\langle vv^T, X\rangle = 4$ with v as in (6.8). This is probably due to the difference in the spectrum of the representation matrices.

The fact that E_{ij} is a rank two matrix can be exploited to avoid repeated eigenvalue computations in the line search[3]. Let $E_{ij} = vv^T - ww^T$ for appropriate $v, w \in \mathbb{R}^n$ and let $\hat{Z} = P\Lambda P^T$ with $P^T P = I_n$ be an eigenvalue decomposition of $\hat{Z}$. Then (6.10) specializes to

$$\min_{u_0 \le 0, \lambda \ge 0} 2u_0 + n\lambda \quad \text{s.t.} \quad u_0 P^T(vv^T - ww^T)P + \lambda I + \Lambda \succeq 0.$$

For a given $\lambda > 0$ we may rewrite the semidefinite constraint as

$$\left(\Lambda + \lambda I - u_0 \bar{w}\bar{w}^T\right) + u_0 \bar{v}\bar{v}^T \succeq 0,$$

where $\bar{v} = P^T v$ and $\bar{w} = P^T w$. Since, for $u_0 \le 0$, the matrix $-u_0\bar{w}\bar{w}^T$ is positive semidefinite and hence $\Lambda + \lambda I - u_0\bar{w}\bar{w}^T \succ 0$, we may employ (6.9) to obtain

$$u_0 \ge -\frac{1}{\bar{v}^T(\Lambda + \lambda I - u_0\bar{w}\bar{w}^T)^{-1}\bar{v}}.$$

Using the Sherman-Morrison-Woodbury formula for the inverse this leads to

$$\bar{v}(\Lambda + \lambda I)^{-1}\bar{v} + u_0 \frac{[\bar{v}(\Lambda + \lambda I)^{-1}\bar{w}]^2}{1 - u_0\bar{w}^T(\Lambda + \lambda I)^{-1}\bar{w}} \le -\frac{1}{u_0}$$

which is equivalent to

$$\left(\sum_{i=1}^n \frac{\bar{v}_i\bar{w}_i}{\lambda_i(\hat{Z}) + \lambda}\right)^2 \ge \left(\sum_{i=1}^n \frac{\bar{v}_i^2}{\lambda_i(\hat{Z}) + \lambda} + \frac{1}{u_0}\right)\left(\sum_{i=1}^n \frac{\bar{w}_i^2}{\lambda_i(\hat{Z}) + \lambda} - \frac{1}{u_0}\right).$$

From this quadratic relation the best u_0 can be computed explicitly for any given $\lambda > 0$ in $O(n)$ arithmetic operations. This speeds up the line search, which is now formulated in λ, considerably. The factorization of $\hat{Z}$ has to be computed only once for all x_{ij} that are considered for fixing.

In Table 6.13 we present some experimental results comparing the pure branch and cut code of Helmberg and Rendl [1998] (the underlying cutting plane algorithm is outlined in Section 6.1.1) to its versions with the fixing routines. The problems are the same as in Table 6.2. The fixing procedures are applied whenever a variable x_{ij} of the current optimal solution satisfies $|x_{ij}| > .98$. This leads to no additional cost for problems in which no variables satisfy this bound. Whenever variables of this size appeared then usually some of them could be fixed.

[3] This was observed by Kurt Anstreicher.

Table 6.13: Average branch and bound results. n gives the dimension of the problem, *nr* refers to the number of instances solved. We give the average computation time and the number of branch and cut nodes for the branch and cut code without fixing variables (column *no fixing*), with the specialized fixing routine exploiting the rank two property (column *with fixing*), and with the general fixing routine based on repeated eigenvalue computations (column *gen. fixing*).

	no fixing		with fixing		gen. fixing		no fixing		with fixing		gen. fixing	
n nr	h:mm:ss	nodes	h:mm:ss	nodes	h:mm:ss	nodes	h:mm:ss	nodes	h:mm:ss	nodes	h:mm:ss	nodes
	$G_{0.5}$						$G_{-1/0/1}$					
40 10	12	8	11	7	11	7	13	9	13	9	13	9
50 10	46	16	46	16	47	16	50	19	50	19	49	19
60 5	4:26	57	4:24	57	4:29	57	2:45	37	2:44	36	2:44	36
70 3	14:17	121	14:06	118	14:24	118	7:44	64	7:40	63	7:40	63
80 2	30:08	145	29:53	145	29:59	145	41:15	203	41:11	203	41:14	203
90 1	2:39:05	583	2:37:07	583	2:39:11	583	11:57:00	2363	11:52:28	2360	11:56:01	2360
	G_p						$G_{\pm p}$					
40 10	7	4	7	4	7	4	1	1	1	1	1	1
50 10	48	14	48	14	48	14	8	2	9	2	9	2
60 5	3:13	32	3:14	31	3:15	31	49	9	49	9	49	9
70 3	7:12	41	7:05	39	7:07	39	4:32	25	5:17	29	5:19	29
80 2	43:06	158	42:57	156	43:11	156	4:57	17	4:57	18	5:02	18
90 1	30:00	69	30:48	73	30:54	73	22:49	63	23:17	63	23:18	63
	Q_{100}						$Q_{100,.2}$					
41 10	23	23	12	10	16	10	1	1	1	1	1	1
51 10	1:12	39	34	14	42	14	6	1	5	1	7	1
61 5	2:47	51	1:11	16	1:26	16	56	15	40	9	47	9
71 3	5:57	71	3:42	35	4:38	35	7:20	71	3:41	32	4:19	32
81 2	11:23	86	9:24	45	10:50	45	7:32	33	3:22	11	5:28	11
91 1	18:35	79	1:21	3	6:04	3	18:17	73	2:56	7	4:33	7
101 1	31:39	137	28:47	104	32:34	104	31:31	113	4:29	7	8:30	7

It was observed in Helmberg and Rendl [1998] that in practice $G_{.5}$ and $G_{-1/0/1}$ are substantially more difficult to solve than G_p and $G_{\pm p}$ and these are again more difficult than Q_{100} and $Q_{100,.2}$. Indeed, for the classes $G_{.5}$, $G_{-1/0/1}$, G_p, and $G_{\pm p}$ the fixing routines were hardly ever called, because no variables satisfied $|x_{ij}| > .98$. Accordingly, the additional cost of the routine is negligible. However, for the "easy" classes of problems Q_{100} and $Q_{100,.2}$ the specialized fixing routine was very successful and lead to large savings in most cases. The general fixing routine was almost equally successful in fixing variables, but the cost of the repeated eigenvalue computations is clearly visible.

Note, that the algorithms trivially extend to other semidefinite relaxations, *e.g.*, for coloring or max k-cut, allowing the possibility to shift eigenvalues directly. It also extends to quadratic 0-1 programming via the scaling of Lemma 3.2.3 to max-cut. Furthermore, our approach is not restricted to the fixing of variables, but can be used to test whether a certain equality or inequality constraint must be satisfied by all optimal solutions. In this sense we may also call this approach *fixing of constraints*.

6.4 Remarks on the Literature

The first steps in combining cutting planes with interior point methods were taken in linear programming, see Mitchell and Borchers [1996] and references therein. In semidefinite programming, interior point cutting plane algorithms started with computational experiments on max-cut, Helmberg [1994]; Helmberg and Rendl [1998]. Building on this work, Karisch [1995]; Karisch and Rendl [1998]; Karisch, Rendl, and Clausen [1997] developed a cutting plane algorithm for graph equipartitioning. Wolkowicz and Zhao [1996] discuss graph partitioning with prespecified sizes. Semidefinite relaxations of the quadratic assignment problem were studied in Karisch [1995]; Zhao, Karisch, Rendl, and Wolkowicz [1998]; Lin and Saigal [1997]. In order to approach more general polyhedra, Helmberg, Rendl, and Weismantel [2000]; Helmberg and Weismantel [1998] investigated the quadratic knapsack problem.

The ideas presented in Section 6.2 for combining cutting planes and the spectral bundle method have not yet appeared in the literature. Even though they still lack proper computational justification we have decided to include them here, in the hope to encourage further work in this direction. Fruitful discussions with Franz Rendl are gratefully acknowledged. We also thank Andreas Eisenblätter for providing the real world test data sets of the co-channel interference minimization problems.

Sensitivity analysis of semidefinite programming is a well studied topic; we refer to Bonnans, Cominetti, and Shapiro [1998] and references therein. Unfortunately, the theory builds on exact solutions and full spectral information and is therefore of little help in implementations. The more practical approach presented here was developed by Helmberg [2000].

Appendix A

Linear Algebra

As a reference we recommend Horn and Johnson [1985]; Horn and Johnson [1991]. We assume familiarity with the basic concepts of linear algebra. In this appendix we give the notation and cite some results that are needed in the text. Unless stated explicitly otherwise, we consider the field $\mathbb{R}$ of real numbers. Vectors are column vectors, $M_{m,n}$ is the set of $m \times n$ matrices, M_n refers to the set of square matrices of order n, S_n denotes the space of symmetric matrices of order n.

The identity matrix in M_n is denoted by I_n or simply I if the dimension is clear. A matrix $P \in M_{m,n}$ is called *orthonormal* if $P^T P = I$.

Theorem A.0.1 ***(Spectral theorem for symmetric matrices)*** *If $A \in S_n$, then A is real orthogonal diagonalizable,* i.e.*, there is an orthonormal matrix $P \in M_n$ and a diagonal matrix $\Lambda \in S_n$ with $A = P\Lambda P^T$ ($P\Lambda P^T$ is called the* eigenvalue decomposition *of A).*

Proof. See Horn and Johnson [1985], (2.5.6). ■

It is a very special property of two matrices if they can be diagonalized by the same matrix.

Definition A.0.2 *$A, B \in M_n$ are* simultaneously diagonalizable *if there is a regular matrix $S \in M_n$ so that $S^{-1}AS$ and $S^{-1}BS$ are both diagonal.*

The columns of S may be interpreted as a common basis of A and B for which both transformations reduce to a scaling of the coordinates.

Likewise we may consider it a special property of two matrices if they are commutative with respect to matrix multiplication. We say that two matrices A and B *commute* if $AB = BA$. In case of symmetric matrices, A and B commute if and only if their product AB is a symmetric matrix. The following theorem shows that simultaneous diagonalization and commutativity are tightly linked together. To maintain its full generality we state the theorem for matrices over the complex numbers $\mathbb{C}$.

Theorem A.0.3 *$A, B \in M_n(\mathbb{C})$ are simultaneously diagonalizable if and only if A and B commute ($AB = BA$). In particular, for $A, B \in S_n$, $AB \in S_n$ if and only if there is an orthonormal matrix P which diagonalizes A and B.*

Proof. See Horn and Johnson [1985], (1.3.12). ■

We denote the eigenvalues of $A \in S_n$ by $\lambda_i(A)$, $i = 1, \ldots, n$. For our purposes it is convenient to sort the eigenvalues non-increasingly, $\lambda_{\max}(A) = \lambda_1(A) \geq \lambda_2(A) \geq \ldots \geq \lambda_n(A) = \lambda_{\min}(A)$.

Theorem A.0.4 ***(Rayleigh-Ritz)*** *Let $A \in S_n$. Then*

$$\lambda_{\min}(A)x^T x \leq x^T Ax \leq \lambda_{\max}(A)x^T x \qquad \text{for all } x \in \mathbb{R}^n$$
$$\lambda_{\max}(A) = \max_{\|x\|=1} x^T Ax$$
$$\lambda_{\min}(A) = \min_{\|x\|=1} x^T Ax.$$

Proof. See Horn and Johnson [1985], (4.2.2). ■

When a symmetric matrix is extended by adding an additional row (and column), its eigenvalues cannot change arbitrarily.

Theorem A.0.5 ***(Interlacing eigenvalues for bordered matrices)*** *Let $A \in S_n$, $y \in \mathbb{R}^n$, $a \in \mathbb{R}$, and let*

$$B = \begin{bmatrix} A & y \\ y^T & a \end{bmatrix}.$$

Then $\lambda_1(B) \geq \lambda_1(A) \geq \lambda_2(B) \geq \lambda_2(A) \geq \cdots \geq \lambda_n(B) \geq \lambda_n(A) \geq \lambda_{n+1}(B)$.

Proof. See Horn and Johnson [1985], (4.3.8). ■

The following theorem bounds the eigenvalues in terms of the matrix elements.

Theorem A.0.6 ***(Geršgorin disc theorem)*** *Let $A = [a_{ij}] \in M_n(\mathbb{C})$, and let $r_i = \sum_{j=1, j\neq i}^{n} |a_{ij}|$. Then all eigenvalues of A are located in the union of n discs*

$$\bigcup_{i=1}^{n} \{z \in \mathbb{C} : \ |z - a_{ii}| \leq r_i\}.$$

Furthermore, if a union of k of these n discs forms a connected region that is disjoint from all the remaining $n-k$ discs, then there are precisely k eigenvalues of A in this region.

Proof. See Horn and Johnson [1985], (6.1.1). ■

For the partitioned matrix

$$A = \begin{bmatrix} B & C \\ D & E \end{bmatrix} \in M_n$$

with B regular, the matrix

$$F = E - DB^{-1}C$$

is called the *Schur complement* of B in A. It corresponds to a block Gaussian elimination step and appears frequently in the text.

Appendix B

Convexity, Cones, and Polyhedra

References for convexity and convex minimization are Rockafellar [1970]; Hiriart-Urruty and Lemaréchal [1993a]; Hiriart-Urruty and Lemaréchal [1993b].

A set $C \subseteq \mathbb{R}^n$ is called *convex* if for $c_1, c_2 \in C$ all points on the straight line segment $\alpha c_1 + (1-\alpha)c_2$ for $\alpha \in [0,1]$ are contained in C as well. For example a *halfspace* $\{x \in \mathbb{R}^n : c^T x \geq b\}$ for some $c \in \mathbb{R}^n$ and $b \in \mathbb{R}$ and the *closed ball* $B_\varepsilon(y) = \{x \in \mathbb{R}^n : \|x - y\| \leq \varepsilon\}$ centered at $y \in \mathbb{R}^n$ with $\varepsilon > 0$ are convex sets.

The intersection of (possibly infinitely many) convex sets is again a convex set. Any closed convex set is the intersection of (possibly infinitely many) halfspaces. An inequality $c^T x \geq b$ is called *valid* for a convex set C if the associated halfspace contains C. A hyperplane $H = \{x : c^T x = b\}$ is called a *supporting hyperplane* of C if C is completely contained in one of its halfspaces and $H \cap C \neq \emptyset$.

If C is a convex set so is $\alpha C = \{\alpha x : x \in C\}$ for $\alpha \in \mathbb{R}$. The *(Minkowski) sum* of two convex sets $C_1 + C_2 = \{x_1 + x_2 : x_1 \in C_1, x_2 \in C_2\}$ is a convex set. The image of a convex set C under a linear transformation A is a convex set, $AC = \{Ax : x \in C\}$. The *convex combination* $\sum_{i=1}^m \alpha_i C_i$ with $\sum_{i=1}^m \alpha_i = 1$, $\alpha_i \geq 0$ for $i = 1, \ldots, m$ of convex sets is a convex set.

The *convex hull* $\mathrm{conv}(S)$ of some set $S \subseteq \mathbb{R}^n$ is the smallest convex set (with respect to set inclusion) that contains S. In fact, $\mathrm{conv}(S)$ is the set of all convex combinations of elements in S.

A vector $x \in \mathbb{R}^n$ is an *affine combination* of vectors $s_1, \ldots, s_k \in \mathbb{R}^n$ if $x = \lambda^T s$ for some $\lambda \in \mathbb{R}^k$ with $\sum_{i=1}^m \lambda_i = 1$. x is *affinely independent of* $s_1, \ldots, s_k$ if such a λ does not exist. A set $S \in \mathbb{R}^n$ is called *affinely independent* if each element $s \in S$ is affinely independent of $S \setminus \{s\}$. The *dimension* $\dim(S)$ is the cardinality of a largest affinely independent subset of S minus one. The *relative interior* $\mathrm{relint}(S)$ is the set of points of S which is in the interior of S with respect to $\mathrm{aff}(S)$, $\mathrm{relint}(S) = \{x : \exists\, \varepsilon > 0 : B_\varepsilon(x) \cap \mathrm{aff}(S) \subseteq S\}$.

A set $C \subseteq \mathbb{R}^n$ is a *cone* if it is closed under nonnegative multiplication and addition ($x, y \in C \Longrightarrow \lambda(x + y) \in C \;\; \forall\, \lambda \geq 0$). This definition implies that a cone is convex. A cone C is *pointed* if $C \cap (-C) = \{0\}$. Linear subspaces and a *ray* $\{x : x = \lambda c, \lambda \geq 0\}$ for some $0 \neq c \in \mathbb{R}^n$ are examples of cones. The *recession cone* 0^+C of a convex set C consists of all directions y such that for $x \in C$, $x + \lambda y \in C \; \forall\, \lambda \geq 0$. The lineality space $\mathrm{lin}(C)$ of a convex set C is the linear subspace $0^+C \cap (-0^+C)$. The cone *generated* by a set S is $\mathrm{cone}(S) = \{x : \exists \lambda \geq 0 : x \in \lambda\, \mathrm{conv}(S)\}$.

A convex subset F of a convex set C is a *face* if every closed line segment in C whose relative interior intersects F is contained in F ($x, y \in C \wedge \exists \alpha \in (0,1) : \alpha x + (1-\alpha)y \in F \Longrightarrow x, y \in F$). If a point in the relative interior of a convex set $C' \subseteq C$ is contained in a face F of C, then $C' \subseteq F$. The collection of relative interiors of all nonempty faces of C forms a partition of C. A face is *proper* if it does not contain C and is called *trivial* if it is empty. An *exposed* face is a face that arises from the intersection of C with a supporting hyperplane.

The faces of dimension zero are called *extreme points* of C. A closed bounded convex set is the convex hull of its extreme points. A half-line face of C (rays with shifted origin) determines an *extreme direction* of C. The extreme directions of cones are called *extreme rays*. A closed convex set not containing a line is the convex hull of its extreme points and extreme directions. If a convex

set is the convex hull of finitely many points and extreme directions it is called *finitely generated*. Convex sets with lineality space not equal zero have neither extreme points nor extreme directions. However, the extremal structure can be obtained by considering the convex set $C \cap \text{lin}(C)^\perp$.

A convex set $P \subseteq \mathbb{R}^n$ is a *polyhedron* if it is finitely generated, *i.e.*, there are finite sets $V, E \in \mathbb{R}^n$ such that $P = \text{conv}(V) + \text{cone}(E)$. Equivalently, a polyhedron is the intersection of finitely many halfspaces, *i.e.*, there is a matrix $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ such that $P = \{x \in \mathbb{R}^n : Ax \leq b\}$. A bounded polyhedron it is called a *polytope*. A cone is *polyhedral* if it is a polyhedron. For a polyhedral cone $C \in \mathbb{R}^n$ there exists some matrix $A \in \mathbb{R}^{m \times n}$ such that $C = \{x \in \mathbb{R}^n : Ax \leq 0\}$. For polyhedra all faces are exposed. The extreme points of a polyhedron are called *vertices*, the maximal (with respect to dimension) proper faces are called *facets*.

Let $D \subseteq \mathbb{R}^n$ and $f : D \to \mathbb{R} \cup \{+\infty, -\infty\}$ be a function. The *epigraph* $\text{epi}(f)$ of f is the set $\{(x, y) : x \in D, y \in \mathbb{R}, y \geq f(x)\}$. If *epi*$(f)$ is convex in $\mathbb{R}^{n+1}$ then f is called *convex*. Any convex function can be extended from D to $\mathbb{R}^n$ by setting $f(x) = +\infty$ for $x \in \mathbb{R}^n \setminus D$. A convex function is called *proper* if it is finite somewhere and $f(x) > -\infty$ everywhere. The pointwise supremum of a (possibly infinite) family of convex functions is again a convex function, because its epigraph is the intersection of convex epigraphs and therefore convex. For convex functions every local minimum is a global minimum. A convex function is called *strictly convex* if $f(\alpha x + (1-\alpha)y) < \alpha f(x) + (1-\alpha)f(y)$. If for a strictly convex function the minimum is attained then it is attained in a unique point. If a convex function is twice continuously differentiable then the positive definiteness of the Hessian matrix (for all x) is equivalent to strict convexity. A *subgradient* of a convex function f at a point $x \in D$ is a vector $s \in \mathbb{R}^n$ satisfying the *subgradient inequality* $f(y) \geq f(x) + \langle s, y - x \rangle$ for all $y \in D$. The set of all subgradients of f at x is the *subdifferential* at x and is denoted by $\partial f(x)$. x is a minimizer of a convex function f if and only if $0 \in \partial f(x)$.

A convex function $\sigma : \mathbb{R}^n \to \mathbb{R}$ is called *sublinear* if it satisfies $\sigma(tx) = t\sigma(x)$ for all $x \in \mathbb{R}^n$ and $t \geq 0$. A function is sublinear if and only its epigraph is a cone in $\mathbb{R}^{n+1}$. For a set $S \in \mathbb{R}^n$ (not necessarily convex), the *support function* $\sigma_S(x) = \sup_{s \in S} \langle s, x \rangle$ is a closed sublinear function.

The projection of a point x onto a closed convex set C is the point $y \in C$ minimizing $\frac{1}{2}\|x - y\|^2$ (the point having minimal Euclidean distance to x). Since the function is strictly convex, y is unique. If $x \notin C$ there is a hyperplane separating x from C strictly, *i.e.*, $\exists\, c \in \mathbb{R}^n, b \in \mathbb{R} : c^T x < b \leq c^T y \quad \forall\, y \in C$. The hyperplane may be chosen so that it is supporting in the projection of x on C.

Appendix C

Graph Instances

The numerical results of Section 6.2.3 were obtained for graphs generated by `rudy`, a device independent graph generator written by Giovanni Rinaldi. It is currently available under

`http://www.zib.de/helmberg/rudy.tar.gz`

The arguments used to generate the graphs are listed below.

G_1 `-rnd_graph 800 6 8001`
G_2 `-rnd_graph 800 6 8002`
G_3 `-rnd_graph 800 6 8003`
G_4 `-rnd_graph 800 6 8004`
G_5 `-rnd_graph 800 6 8005`
G_6 `-rnd_graph 800 6 8001 -random 0 1 8001 -times 2 -plus -1`
G_7 `-rnd_graph 800 6 8002 -random 0 1 8002 -times 2 -plus -1`
G_8 `-rnd_graph 800 6 8003 -random 0 1 8003 -times 2 -plus -1`
G_9 `-rnd_graph 800 6 8004 -random 0 1 8004 -times 2 -plus -1`
G_{10} `-rnd_graph 800 6 8005 -random 0 1 8005 -times 2 -plus -1`
G_{11} `-toroidal_grid_2D 100 8 -random 0 1 8001 -times 2 -plus -1`
G_{12} `-toroidal_grid_2D 50 16 -random 0 1 8002 -times 2 -plus -1`
G_{13} `-toroidal_grid_2D 25 32 -random 0 1 8003 -times 2 -plus -1`
G_{14} `-planar 800 99 8001 -planar 800 99 8002 +`
G_{15} `-planar 800 99 8003 -planar 800 99 8004 +`
G_{16} `-planar 800 99 8005 -planar 800 99 8006 +`
G_{17} `-planar 800 99 8007 -planar 800 99 8008 +`
G_{18} `-planar 800 99 8001 -planar 800 99 8002 + -random 0 1 8001 -times 2 -plus -1`
G_{19} `-planar 800 99 8003 -planar 800 99 8004 + -random 0 1 8002 -times 2 -plus -1`
G_{20} `-planar 800 99 8005 -planar 800 99 8006 + -random 0 1 8003 -times 2 -plus -1`
G_{21} `-planar 800 99 8007 -planar 800 99 8008 + -random 0 1 8004 -times 2 -plus -1`
G_{22} `-rnd_graph 2000 1 20001`
G_{23} `-rnd_graph 2000 1 20002`
G_{24} `-rnd_graph 2000 1 20003`
G_{25} `-rnd_graph 2000 1 20004`
G_{26} `-rnd_graph 2000 1 20005`
G_{27} `-rnd_graph 2000 1 20001 -random 0 1 20001 -times 2 -plus -1`
G_{28} `-rnd_graph 2000 1 20002 -random 0 1 20002 -times 2 -plus -1`
G_{29} `-rnd_graph 2000 1 20003 -random 0 1 20003 -times 2 -plus -1`
G_{30} `-rnd_graph 2000 1 20004 -random 0 1 20004 -times 2 -plus -1`
G_{31} `-rnd_graph 2000 1 20005 -random 0 1 20005 -times 2 -plus -1`
G_{32} `-toroidal_grid_2D 100 20 -random 0 1 20003 -times 2 -plus -1`
G_{33} `-toroidal_grid_2D 80 25 -random 0 1 20002 -times 2 -plus -1`
G_{34} `-toroidal_grid_2D 50 40 -random 0 1 20001 -times 2 -plus -1`

G_{35} -planar 2000 99 20001 -planar 2000 99 20002 +
G_{36} -planar 2000 99 20003 -planar 2000 99 20004 +
G_{37} -planar 2000 99 20005 -planar 2000 99 20006 +
G_{38} -planar 2000 99 20007 -planar 2000 99 20008 +
G_{39} -planar 2000 99 20001 -planar 2000 99 20002 + -random 0 1 20001 -times 2 -plus -1
G_{40} -planar 2000 99 20003 -planar 2000 99 20004 + -random 0 1 20002 -times 2 -plus -1
G_{41} -planar 2000 99 20005 -planar 2000 99 20006 + -random 0 1 20003 -times 2 -plus -1
G_{42} -planar 2000 99 20007 -planar 2000 99 20008 + -random 0 1 20004 -times 2 -plus -1
G_{43} -rnd_graph 1000 2 10001
G_{44} -rnd_graph 1000 2 10002
G_{45} -rnd_graph 1000 2 10003
G_{46} -rnd_graph 1000 2 10004
G_{47} -rnd_graph 1000 2 10005
G_{48} -toroidal_grid_2D 50 60
G_{49} -toroidal_grid_2D 30 100
G_{50} -toroidal_grid_2D 25 120
G_{51} -planar 1000 100 10001 -planar 1000 100 10002 +
G_{52} -planar 1000 100 10003 -planar 1000 100 10004 +
G_{53} -planar 1000 100 10005 -planar 1000 100 10006 +
G_{54} -planar 1000 100 10007 -planar 1000 100 10008 +
G_{55} -rnd_graph 5000 0.12 50001
G_{56} -rnd_graph 5000 0.1 50001 -random 0 1 50001 -times 2 -plus -1
G_{57} -toroidal_grid_2D 50 100 -random 0 1 50001 -times 2 -plus -1
G_{58} -planar 5000 99 50001 -planar 5000 99 50002 +
G_{59} -planar 5000 99 50001 -planar 5000 99 50002 + -random 0 1 50001 -times 2 -plus -1
G_{60} -rnd_graph 7000 0.07 70001
G_{61} -rnd_graph 7000 0.07 70001 -random 0 1 70001 -times 2 -plus -1
G_{62} -toroidal_grid_2D 70 100 -random 0 1 70001 -times 2 -plus -1
G_{63} -planar 7000 99 70001 -planar 7000 99 70002 +
G_{64} -planar 7000 99 70001 -planar 7000 99 70002 + -random 0 1 70001 -times 2 -plus -1
G_{65} -toroidal_grid_2D 80 100 -random 0 1 100001 -times 2 -plus -1
G_{66} -toroidal_grid_2D 90 100 -random 0 1 100001 -times 2 -plus -1
G_{67} -toroidal_grid_2D 100 100 -random 0 1 100001 -times 2 -plus -1

Bibliography

Aardal, K. and Weismantel, R. (1997). Polyhedral combinatorics. In Dell'Amico, M., Maffioli, F., and Martello, S., editors, *Annotated Bibliographies in Combinatorial Optimization*, pages 31–44. John Wiley & Sons, Ltd., Chichester, England.

Alizadeh, F. (1995). Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optim.*, 5(1):13–51.

Alizadeh, F., Haeberly, J.-P. A., and Overton, M. L. (1994). Primal–dual interior–point methods for semidefinite programming. Draft presented at the XV symposium on mathematical programming, Computer Science Department, Courant Institute of Mathematical Sciences, New York University, New York.

Alizadeh, F., Haeberly, J.-P. A., and Overton, M. L. (1997). Complementarity and nondegeneracy in semidefinite programming. *Math. Programming*, 77(2):111–128.

Alizadeh, F., Haeberly, J.-P. A., and Overton, M. L. (1998). Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results. *SIAM J. Optim.*, 8(3):746–768.

Anstreicher, K., Chen, X., Wolkowicz, H., and Yuan, Y.-X. (1998). Strong duality for a trust-region type relaxation of the quadratic assignment problem. Research Report CORR 98-31, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada.

Anstreicher, K. M. (1999). Linear programming in $O(\frac{n^3}{\ln n}L)$ operations. *SIAM J. Optim.*, 9:803–812.

Anstreicher, K. M. (2000). Eigenvalue bounds versus semidefinite relaxations for the quadratic assignment problem. *SIAM J. Optim.*, 11(1):254–265.

Anstreicher, K. M. and Brixius, N. W. (1999). A new bound for the quadratic assignment problem based on convex quadratic programming. Technical report, Department of Management Science, University of Iowa, Iowa City, IA 52242, USA.

Balas, E. (1975). Disjunctive programming: Cutting planes from logical conditions. In Mangasarian et al., O., editor, *Nonlinear Programming 2*, pages 279–312. Academic Press.

Balas, E., Ceria, S., Cornuejols, G., and Pataki, G. (1994). Updated semi-definite constraints. Technical report, Carnegie Mellon University, Pittsburgh.

Barahona, F. (1983). The max–cut problem in graphs not contractible to K_5. *Oper. Res. Letters*, 2:107–111.

Barahona, F. (1993). On cuts and matchings in planar graphs. *Math. Programming*, 60:53–68.

Barahona, F., Grötschel, M., and Mahjoub, R. (1985). Facets of the bipartite subgraph polytope. *Math. Oper. Res.*, 10:340–358.

Barahona, F., Jünger, M., and Reinelt, G. (1989). Experiments in quadratic 0–1 programming. *Math. Programming*, 44:127–137.

Barker, G. P. and Carlson, D. (1975). Cones of diagonally dominant matrices. *Pacific Journal of Mathematics*, 57(1):15–32.

Ben-Israel, A., Charnes, A., and Kortanek, K. O. (1969). Duality and asymptotic solvability over cones. *Bulletin Amer. Math. Soc.*, 75:318–324.

Ben-Israel, A., Charnes, A., and Kortanek, K. O. (1971). Asymptotic duality over closed convex sets. *Journal of Mathematical Analysis and Apllications*, 35:677–691.

Ben-Tal, A. and Nemirovski, A. (1998). Robust convex optimization. *Math. Oper. Res.*, 23(4):769–805.

Benson, S., Ye, Y., and Zhang, X. (1998). Mixed linear and semidefinite programming for combinatorial and quadratic optimization. Technical report, Department of Management Science, University of Iowa, IA, 52242, USA. to appear in Optimization Methods and Software.

Benson, S., Ye, Y., and Zhang, X. (2000). Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM J. Optim.*, 10(2):443–461.

Berge, C. (1961). Färbung von Graphen, deren sämtliche bzw. deren ungerade Kreise starr sind (Zusammenfassung). *Wissenschaftliche Zeitschrift, Mathematisch-Naturwissenschaftliche Reihe*, pages 114–115.

Bertsekas, D. P. (1995). *Nonlinear Programming*. Athena Scientific, Belmont, Massachusetts.

Bertsimas, D. and Ye, Y. (1998). Semidefinite relaxations, multivariate normal distributions, and order statistics. In Du, D.-Z. and Pardalos, P., editors, *Handbook of Combinatorial Optimization*, pages 1–19. Kluwer Academic Publishers.

Bonnans, F., Cominetti, R., and Shapiro, A. (1998). Sensitivity analysis of optimization problems under second order regular constraints. *Math. Oper. Res.*, 23(4):806–831.

Borndörfer, R., Eisenblätter, A., Grötschel, M., and Martin, A. (1998). Frequency assignment in cellular phone networks. *Annals of Operations Research*, 76:73–93.

Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V. (1994). *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *SIAM Studies in Applied Mathematics*. SIAM, Philadelphia.

Burer, S. and Monteiro, R. D. (1998). An efficient algorithm for solving the MAXCUT SDP relaxation. Manuscript, School of ISyE, Georgia Tech, Atlanta, GA 30332, USA.

Caprara, A. and Fischetti, M. (1997). Branch-and-cut algorithms. In Dell'Amico, M., Maffioli, F., and Martello, S., editors, *Annotated Bibliographies in Combinatorial Optimization*, pages 45–63. John Wiley & Sons, Ltd., Chichester, England.

Caprara, A., Pisinger, D., and Toth, P. (1999). Exact solution of the quadratic knapsack problem. *INFORMS J. Comput.*, 11(2):125–137.

Chen, H. and Frieze, A. (1996). Coloring bipartite hypergraphs. In Cunningham, W. H., McCormick, S. T., and Queyranne, M., editors, *Integer Programming and Combinatorial Optimization*, volume 1084 of *Lecture Notes in Computer Science*, pages 345–358. Springer.

Chopra, S. and Rao, M. R. (1993). The partition problem. *Math. Programming*, 59:87–115.

Chopra, S. and Rao, M. R. (1995). Facets of the k-partition polytope. *Discrete Applied Mathematics*, 61:27–48.

Chor, B. and Sudan, M. (1995). A geometric approach to betweenness. In *ESA '95 Proceedings*, volume 979 of *Lecture Notes in Computer Science*, pages 227–237. Springer.

Cullum, J., Donath, W. E., and Wolfe, P. (1975). The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices. *Math. Programming Study*, 3:35–55.

de Klerk, E., Roos, C., and Terlaky, T. (1997). Initialization in semidefinite programming via a self-dual skew-symmetric embedding. *Operations Research Letters*, 20:213–221.

de Klerk, E., van Maaren, H., and Warners, J. (1999). Relaxations of the satisfiability problem using semidefinite programming. CWI Report SEN-R9903, CWI, Amsterdam, The Netherlands. To appear in the SAT2000 issue of the J. of Automated Reasoning.

De Simone, C. (1989). The cut polytope and the boolean quadric polytope. *Discrete Mathematics*, 79:71–75.

De Simone, C. and Rinaldi, G. (1994). A cutting plane algorithm for the max–cut problem. *Optimization Methods and Software*, 3:195–214.

del Corso, G. M. (1997). Esitimating an eigenvector by the power method with a random start. *SIAM J. Matrix Anal. Appl.*, 18(4):913–937.

Delorme, C. and Poljak, S. (1991). Combinatorial properties and the complexity of a max-cut approximation. Technical Report 91687-OR, LRI University Paris-Sud, F-91405 ORSAY Cedex, France.

Delorme, C. and Poljak, S. (1993). Laplacian eigenvalues and the maximum cut problem. *Math. Programming*, 62:557–574.

Deza, M., Grötschel, M., and Laurent, M. (1991). Complete descriptions of small multicut polytopes. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 4:221–252.

Deza, M. and Laurent, M. (1997). *Geometry of Cuts and Metrics*, volume 15 of *Algorithms and Combinatorics*. Springer.

Deza, M., Laurent, M., and Poljak, S. (1992). The cut cone III: On the triangle facets. *Graphs Combin.*, 8:125–142.

Donath, W. E. and Hoffman, A. J. (1972). Algorithms for partitioning of graphs and computer logic based on eigenvectors of connection matrices. *IBM Technical Disclosure Bulletin*, 15(3):938–944.

Donath, W. E. and Hoffman, A. J. (1973). Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17:420–425.

Duffin, R. J. (1956). Infinite programs. In Kuhn, H. W. and Tucker, A. W., editors, *Linear Inequalitites and Related Systems*, volume 38 of *Annals of Math. Studies*, pages 157–170. Princeton Univ. Press, Princeton, N.J.

Feige, U. and Goemans, M. X. (1995). Approximating the value of two prover proof systems, with applications to MAX 2SAT and MAX DICUT. In *Proceedings of the Third Israel Symposium on Theory of Computing and Systems*, pages 182–189, Tel Aviv, Israel.

Fiacco, A. V. and McCormick, G. P. (1968). *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Wiley, New York.

Fletcher, R. (1985). Semi-definite matrix constraints in optimization. *SIAM J. Control and Optimization*, 23(4):493–513.

Frieze, A. and Jerrum, M. (1995). Improved approximation algorithms for MAX k-CUT and MAX BISECTION. In Balas, E. and Clausen, J., editors, *Integer Programming and Combinatorial Optimization*, volume 920 of *Lecture Notes in Computer Science*, pages 1–13. Springer.

Frisch, K. R. (1955). The logarithmic potential method for convex programming. Memorandum, University Institute of Economics, Oslo, Norway.

Gill, P. E., Murray, W., Saunders, M. A., Tomlin, J. A., and Wright, M. H. (1986). On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method. *Math. Programming*, 36:183–209.

Goemans, M. X. (1997). Semidefinite programming in combinatorial optimization. *Math. Programming*, 79:143–161.

Goemans, M. X. (1998). Semidefinite programming and combinatorial optimization. *Documenta Mathematica*, Extra Volume ICM 1998, III:657–666.

Goemans, M. X. and Williamson, D. P. (1995). Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42:1115–1145.

Goldfarb, D. and Scheinberg, K. (1996). Interior point trajectories in semidefinite programming. Technical report, Dept. of IEOR, Columbia University, New York, NY.

Golub, G. H. and van Loan, C. F. (1989). *Matrix Computations*. The Johns Hopkins University Press, 2^{nd} edition.

Graham, A. (1981). *Kronecker Products and Matrix Calculus With Applications*. Mathematics and its Applications. Ellis Horwood Limited, Chichester.

Grötschel, M., Lovász, L., and Schrijver, A. (1981). The ellipsoid method and its consequences in combinatorial optimziation. *Combinatorica*, 1:169–197.

Grötschel, M., Lovász, L., and Schrijver, A. (1984). Polynomial algorithms for perfect graphs. *Annals of Discrete Mathematics*, 21:325–356.

Grötschel, M., Lovász, L., and Schrijver, A. (1988). *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 2^{nd} edition.

Grötschel, M. and Wakabayashi, Y. (1989). A cutting plane algorithm for a clustering problem. *Math. Programming*, 45:59–96.

Grötschel, M. and Wakabayashi, Y. (1990). Facets of the clique partitioning polytope. *Math. Programming*, 47:367–387.

Gruber, G. and Rendl, F. (1999). Approximating stable sets using the ϑ-function and cutting planes. In Inderfurth, K., Schwödiauer, G., Domschke, W., Juhnke, F., Kleinschmidt, P., and Wäscher, G., editors, *Operations Research Proceedings 1999*, pages 69–74, Berlin. Gesellschaft für Operations Research e.V.(GOR), Springer.

Halperin, E. and Zwick, U. (1999). Approximation algorithms for MAX 4-SAT and rounding procedures for semidefinite programming. In Cornuéjols, G., Burkard, R., and Woeginger, G. J., editors, *Integer Programming and Combinatorial Optimization*, volume 1610 of *Lecture Notes in Computer Science*, pages 202–217. Springer.

Hammer, P. L. (1965). Some network flow problems solved with pseudo-boolean programming. *Operations Research*, 13:388–399.

Håstad, J. (1997). Some optimal inapproximability results. *Electronic Colloquium on Computational Complexity*, TR97-037. Revision 01 of Sept. 24, 1999.

Helmberg, C. (1994). *An interior point method for semidefinite programming and max–cut bounds*. PhD thesis, Graz University of Technology, Rechbauerstraße 12, A-8010 Graz, Austria.

Helmberg, C. (2000). Fixing variables in semidefinite relaxations. *SIAM J. Matrix Anal. Appl.*, 21(3):952–969.

Helmberg, C. and Kiwiel, K. C. (1999). A spectral bundle method with bounds. ZIB Preprint SC-99-37, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Takustraße 7, 14195 Berlin, Germany.

Helmberg, C. and Oustry, F. (2000). Bundle methods to minimize the maximum eigenvalue function. In Wolkowicz, H., Saigal, R., and Vandenberghe, L., editors, *Handbook of Semidefinite Programming*, pages 307–337. Kluwer Academic Publishers, Boston/Dordrecht/London.

Helmberg, C., Poljak, S., Rendl, F., and Wolkowicz, H. (1995). Combining semidefinite and polyhedral relaxations for integer programs. In Balas, E. and Clausen, J., editors, *Integer Programming and Combinatorial Optimization*, volume 920 of *Lecture Notes in Computer Science*, pages 124–134. Springer.

Helmberg, C. and Rendl, F. (1998). Solving quadratic (0,1)-problems by semidefinite programs and cutting planes. *Math. Programming*, 82(3):291–315.

Helmberg, C. and Rendl, F. (2000). A spectral bundle method for semidefinite programming. *SIAM J. Optim.*, 10(3):673–696.

Helmberg, C., Rendl, F., Vanderbei, R. J., and Wolkowicz, H. (1996). An interior–point method for semidefinite programming. *SIAM J. Optim.*, 6(2):342–361.

Helmberg, C., Rendl, F., and Weismantel, R. (2000). A semidefinite programming approach to the quadratic knapsack problem. *J. of Combinatorial Optimization*, 4(2):197–215.

Helmberg, C. and Weismantel, R. (1998). Cutting plane algorithms for semidefinite relaxations. In Pardalos, P. M. and Wolkowicz, H., editors, *Topics in Semidefinite and Interior-Point Methods*, volume 18 of *Fields Institute Communications*, pages 197–213. American Mathematical Society.

Hill, R. D. and Waters, S. R. (1987). On the cone of positive semidefinite matrices. *Linear Algebra and its Applications*, 90:81–88.

Hiriart-Urruty, J.-B. and Lemaréchal, C. (1993a). *Convex Analysis and Minimization Algorithms I*, volume 305 of *Grundlehren der mathematischen Wissenschaften*. Springer, Berlin, Heidelberg.

Hiriart-Urruty, J.-B. and Lemaréchal, C. (1993b). *Convex Analysis and Minimization Algorithms II*, volume 306 of *Grundlehren der mathematischen Wissenschaften*. Springer, Berlin, Heidelberg.

Homer, S. and Peinado, M. (1995). Design and performance of parallel and distributed approxiamtion algorithms for max-cut. Technical report, Department of Computer Science, Boston University, 111 Cummington Street, Boston, MA 02215.

Horn, R. A. and Johnson, C. R. (1985). *Matrix Analysis*. Cambridge University Press.

Horn, R. A. and Johnson, C. R. (1991). *Topics in Matrix Analysis*. Cambridge University Press.

Jarre, F. (1993). An interior–point method for minimizing the maximum eigenvalue of a linear combination of matrices. *Siam J. Control and Optimization*, 31(5):1360–1377.

Johnson, E., Mehrotra, A., and Nemhauser, G. (1993). Min-cut clustering. *Math. Programming*, 62:133–152.

Juhász, F. (1981). On the spectrum of a random graph. In Lovász, L. and Sós, V. T., editors, *Algebraic Methods in Graph Theory*, volume 25 of *Colloquia Mathematica Societatis János Bolyai*, pages 313–316. North-Holland Publishing Company, Amsterdam-Oxford-New York.

Jünger, M. and Rinaldi, G. (1998). Relaxations of the max cut problem and computation of spin glass ground states. In Kischka, P., editor, *Proc. SOR '97*, pages 74–83.

Karger, D., Motwani, R., and Sudan, M. (1994). Approximate graph coloring by semidefinite programming. In *FOCS 94*, pages 2–13.

Karisch, S. E. (1995). *Nonlinear Approaches for Quadratic Assignment and Graph Partition Problems*. PhD thesis, Department of Mathematics, Graz University of Technology, Steyrergasse 30, A-8010 Graz, Austria.

Karisch, S. E. and Rendl, F. (1998). Semidefinite programming and graph equipartition. In Pardalos, P. M. and Wolkowicz, H., editors, *Topics in Semidefinite and Interior-Point Methods*, volume 18 of *Fields Institute Communications*, pages 77–95. American Mathematical Society.

Karisch, S. E., Rendl, F., and Clausen, J. (1997). Solving graph bisection problems with semidefinite programming. Technical Report DIKU-TR-97/9, Department of Computer Science, University of Copenhagen, Universitetsparken 1, DK-2100 Copenhagen, Denmark. To appear in INFORMS J. Comput.

Karloff, H. (1996). How good ist the Goemans-Williamson MAX-CUT algorithm? In *STOC'96*, pages 427–434.

Karloff, H. and Zwick, U. (1997). A 7/8-approximation algorithm for MAX 3SAT? In *Proc. of 38th FOCS*, pages 406–415.

Karmarkar, N. (1984). An new polynomial–time algorithm for linear programming. *Combinatorica*, 4:373–395.

Khachiyan, L. and Porkolab, L. (1997). Computing integral points in convex semi-algebraic sets. In *FOCS 97*, pages 162–171.

Kiwiel, K. C. (1983). An aggregate subgradient method for nonsmooth convex minimization. *Math. Programming*, 27:320–341.

Kiwiel, K. C. (1986). A linearization algorithm for optimizing control systems subject to singular value inequalities. *IEEE Transactions on Automatic Control*, AC-31(7):595–602.

Kiwiel, K. C. (1990). Proximity control in bundle methods for convex nondifferentiable minimization. *Math. Programming*, 46:105–122.

Kiwiel, K. C. (1995). Approximation in proximal bundle methods and decomposition of convex programs. *Journal of Optimization Theory and Applications*, 84(3):529–548.

Klein, P. and Lu, H.-I. (1996). Efficient approximation algorithms for semidefinite programs arising from MAXCUT and COLORING. In *STOC'96*, pages 338–347.

Knuth, D. E. (1994). The sandwich theorem. *The Electronic Journal of Combinatorics*, 1:#A1.

Kojima, M., Mizuno, S., and Yoshise, A. (1989a). A polynomial–time algorithm for a class of linear compelementarity problems. *Math. Programming*, 44:1–26.

Kojima, M., Mizuno, S., and Yoshise, A. (1989b). A primal–dual interior point algorithm for linear programming. In Megiddo, N., editor, *Progress in Mathematical Programming*, chapter 2, pages 29–47. Springer, New York.

Kojima, M., Shida, M., and Shindoh, S. (1997). Reduction of monotone linear complementarity problems over cones to linear programs over cones. *Acta Mathematica Vietnamica*, 22:147–157.

Kojima, M., Shida, M., and Shindoh, S. (1998). Local convergence of predictor-corrector infeasible-interior-point algorithms for SDPs and SDLCPs. *Math. Programming*, 80(2):129–160.

Kojima, M., Shida, M., and Shindoh, S. (1999). A predictor-corrector interior-point algorithm for the semidefinite linear complementarity problem using the Alizadeh-Haeberly-Overton search direction. *SIAM J. Optim.*, 9(2):444–465.

Kojima, M., Shindoh, S., and Hara, S. (1994). Interior–point methods for the monotone linear complementarity problem in symmetric matrices. Research Report B–282, Department of Information Sciences, Tokyo Institute of Technology. Revised April 1995.

Kojima, M., Shindoh, S., and Hara, S. (1997). Interior–point methods for the monotone linear complementarity problem in symmetric matrices. *SIAM J. Optim.*, 7(1):86–125.

Laurent, M. and Poljak, S. (1995). On a positive semidefinite relaxation of the cut polytope. *Linear Algebra and its Applications*, 223/224:439–461.

Laurent, M. and Poljak, S. (1996a). Gap inequalities for the cut polytope. *Europ. J. Combinatorics*, 17:233–254.

Laurent, M. and Poljak, S. (1996b). On the facial structure of the set of correlation matrices. *SIAM Journal on Matrix Analysis and Applications*, 17:530–547.

Laurent, M., Poljak, S., and Rendl, F. (1997). Connections between semidefinite relaxations of the max-cut and stable set problems. *Math. Programming*, 77(2):225–246.

Lemaréchal, C., Oustry, F., and Sagastizábal, C. (2000). The $\mathcal{U}$-Lagrangian of a convex function. *Trans. Amer. Math. Soc.*, 352(2):711–729.

Lewis, A. S. and Overton, M. L. (1996). Eigenvalue optimization. *Acta Numerica*, pages 149–190.

Lin, C.-J. and Saigal, R. (1995a). An infeasible start predictor corrector method for semi-definite linear programming. Technical report, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, Michigan 48109-2117.

Lin, C.-J. and Saigal, R. (1995b). A predictor-corrector method for semidefinite linear programming. Technical report, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, Michigan 48109-2117. Revised December 1995.

Lin, C.-J. and Saigal, R. (1997). On solving large-scale semidefinite programming problems – a case study of quadratic assignment problem. Technical report, Department of Industrial and Operations Engineering, University of Michigan.

Lovász, L. (1979). On the Shannon capacity of a graph. *IEEE Transactions on Information Theory*, IT-25(1):1–7.

Lovász, L. and Schrijver, A. (1991). Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optim.*, 1(2):166–190.

Luo, Z.-Q., Sturm, J. F., and Zhang, S. (1996). Duality and self-duality for conic convex programming. Report 9620/A, Econometric Institute, Erasmus University of Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands.

Luo, Z.-Q., Sturm, J. F., and Zhang, S. (1997). Duality results for conic convex programming. Report 9719/A, Econometric Institute, Erasmus University of Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands.

Mahajan, S. and Ramesh, H. (1995). Derandomizing semidefinite programming based approximation algorithms. In *FOCS 95*, pages 162–169.

Megiddo, N. (1989). Pathways to the optimal set in linear programming. In Megiddo, N., editor, *Progress in Mathematical Programming*, chapter 8, pages 131–158. Springer, New York.

Mehlhorn, K., Näher, S., Seel, M., and Uhrig, C. (1999). *The LEDA User Manual, Version 3.8.* Max-Planck-Institut für Informatik, 66123 Saarbrücken, Germany. `http://www.mpi-sb.mpg.de/LEDA/leda.html`.

Mitchell, J. E. and Borchers, B. (1996). Solving real-world linear ordering problems using a primal-dual interior point cutting plane method. *Annals of Operations Research*, 62:253–276.

Mohar, B. and Poljak, S. (1990). Eigenvalues and the max-cut problem. *Czechoslovak Mathematical Journal*, 40(115):343–352.

Monteiro, R. D. C. (1997). Primal–dual path-following algorithms for semidefinite programming. *SIAM J. Optim.*, 7(3):663–678.

Monteiro, R. D. C. (1998). Polynomial convergence of primal-dual algorithms for semidefinite programming based on Monteiro and Zhang family of directions. *SIAM J. Optim.*, 8(3):797–812.

Monteiro, R. D. C. and Adler, I. (1989a). Interior path following primal–dual algorithms. Part I: Linear programming. *Math. Programming*, 44:27–41.

Monteiro, R. D. C. and Adler, I. (1989b). Interior path following primal–dual algorithms. Part II: Convex quadratic programming. *Math. Programming*, 44:43–66.

Monteiro, R. D. C. and Zanjacomo, P. (1997). A note on the existence of the Alizadeh-Haeberly-Overton direction for semidefinite programming. *Math. Programming*, 78(3):393–396.

Monteiro, R. D. C. and Zhang, Y. (1998). A unified analysis for a class of path-following primal-dual interior-point algorithms for semidefinite programming. *Math. Programming*, 81(3):281–299.

Nesterov, Y. (1998). Semidefinite relaxation and nonconvex quadratic optimization. *Optimization Methods and Software*, 9:141–160.

Nesterov, Y. and Nemirovskii, A. (1994). *Interior–Point Polynomial Algorithms in Convex Programming.* SIAM Studies in Applied Mathematics, Philadelphia.

Nesterov, Y. and Todd, M. J. (1997). Self-scaled barriers and interior-point methods for convex programming. *Math. Oper. Res.*, 22:1–42.

Nesterov, Y. and Todd, M. J. (1998). Primal-dual interior-point methods for self-scaled cones. *SIAM J. Optim.*, 8:324–364.

Nesterov, Y., Todd, M. J., and Ye, Y. (1999). Infeasible-start primal-dual methods and infeasibility detectors for nonlinear programming problems. *Math. Programming*, 84(2):227–267.

Oustry, F. (1999a). The $\mathcal{U}$-Lagrangian of the maximum eigenvalue function. *SIAM J. Optim.*, 9(2):526–549.

Oustry, F. (1999b). A second-order bundle method to minimize the maximum eigenvalue function. Technical Report RR-3738, INRIA Rhone-Alpes, ZIRST - 655 avenue de l'Europe, F-38330 Montbonnot Saint-Martin. To appear in Math. Programming.

Overton, M. L. (1992). Large–scale optimization of eigenvalues. *SIAM J. Optim.*, 2(1):88–120.

Overton, M. L. and Womersly, R. S. (1992). On the sum of the largest eigenvalues of a symmetric matrix. *SIAM J. Matrix Anal. Appl.*, 13(1):41–45.

Overton, M. L. and Womersly, R. S. (1993). Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices. *Math. Programming*, 62(2):321–357.

Overton, M. L. and Womersly, R. S. (1995). Second derivatives for optimizing eigenvalues of symmetric matrices. *SIAM J. Matrix Anal. Appl.*, 16(3):697–718.

Padberg, M. (1989). The boolean quadric polytope. *Math. Programming*, 45:132–172.

Parlett, B. N. (1998). *The Symmetric Eigenvalue Problem*. SIAM.

Pataki, G. (1998). On the rank of extreme matrices in semidefinite programming and the multiplicity of optimal eigenvalues. *Math. Oper. Res.*, 23(2):339–358.

Polak, E. and Wardi, Y. (1982). Nondifferentiable optimization algorithm for designing control systems having singular value inequalities. *Automatica*, 18:267–283.

Poljak, S. and Rendl, F. (1995a). Nonpolyhedral relaxations of graph-bisection problems. *SIAM J. Optim.*, 5(3):467–487.

Poljak, S. and Rendl, F. (1995b). Solving the max-cut problem using eigenvalues. *Discrete Applied Mathematics*, 62:249–278.

Poljak, S., Rendl, F., and Wolkowicz, H. (1995). A recipe for semidefinite relaxation for $(0,1)$–quadratic programming. *J. Global Optim.*, 7(1):51–73.

Poljak, S. and Tuza, Z. (1995). Maximum cuts and large bipartite subgraphs. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 20:181–244.

Porkolab, L. and Khachiyan, L. (1997). On the complexity of semidefinite programs. *Journal of Global Optimization*, 10(4):351–365.

Potra, F. A. and Sheng, R. (1996). Superlinear convergence of a predictor-corrector method for semidefinite programming without shrinking central path neighborhood. Reports on Computational Mathematics 91, Department of Mathematics, University of Iowa, Iowa City, IA 52242.

Potra, F. A. and Sheng, R. (1998a). On homogenous interior-point algorithms for semidefinite programming. *Optimization Methods and Software*, 9.

Potra, F. A. and Sheng, R. (1998b). A superlinearly convergent primal-dual infeasible-interior-point algorithm for semidefinite programming. *SIAM J. Optim.*, 8(4):1007–1028.

Preparata, F. P. and Shamos, M. I. (1985). *Computational Geometry*. Texts and Monographs in Computer Science. Springer-Verlag. Third corrected printing 1990.

Ramana, M. (1997). An exact duality theory for semidefinite programming and its complexity implications. *Math. Programming*, 77(2):129–162.

Ramana, M., Tunçel, L., and Wolkowicz, H. (1997). Strong duality for semidefinite porgramming. *SIAM J. Optim.*, 7(3):641–662.

Rendl, F. and Wolkowicz, H. (1995). A projection technique for partitioning the nodes of a graph. *Annals of Operations Research*, 58:155–180.

Renegar (1988). A polynomial–time algorithm, based on Newton's Method, for linear programming. *Math. Programming*, 40:50–94.

Rockafellar, R. T. (1970). *Convex Analysis*. Princeton University Press, Princeton, New Jersey.

Saad, Y. (1992). *Numerical Methods for Large Eigenvalue Problems*. Halsted Press, New York.

Schramm, H. and Zowe, J. (1992). A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results. *SIAM J. Optim.*, 2:121–152.

Seymour, P. D. (1981). Matroids and multicommodity flows,. *Europ. J. Combin.*, 2:257–290.

Shapiro, A. and Fan, M. K. H. (1995). On eigenvalue optimization. *SIAM J. Optim.*, 5(3):552–569.

Sherali, H. and Adams, W. P. (1990). A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Disc. Math.*, 3(3):411–430.

Shor, N. Z. (1987). Quadratic optimization problems. *Soviet Journal of Computer and Systems Sciences*, 25:1–11. Originally published in Tekhnicheskaya Kibernetika, No. 1, 1987, pp. 128-139.

Skutella, M. (1998). Semidefinite relaxations for parallel machine scheduling. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS'98)*, pages 472–481.

Sonnevend, G. (1985). An analytic center for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming. In et al., A. P., editor, *Proceedings of the 12th IFIP Conference*, Lecture Notes in Control and Inform. Sci. , Vol. 84, pages 866–876. Springer-Verlag.

Sorensen, D. C. (1992). Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM J. Matrix Anal. Appl.*, 13(1):357–385.

Sturm, J. F. (1997). *Primal-Dual Interior Point Approach to Semidefinite Programming*. PhD thesis, Erasmus University Rotterdam, Rotterdam Netherlands. Tinbergen Institute Research Series vol. 156, Thesis Publishers, Amsterdam, The Netherlands, 1997.

Sturm, J. F. and Zhang, S. (1999). Symmetric primal-dual path following algorithms for semidefinite programming. *Applied Numerical Mathematics*, 29:301–315.

Todd, M. J. (1999). A study of search directions in primal-dual interior-point methods for semidefinite programming. *Optimization Methods and Software*, 11:1–46.

Todd, M. J., Toh, K. C., and Tütüncü, R. H. (1998). On the Nesterov-Todd direction in semidefinite programming. *SIAM J. Optim.*, 8(3):769–796.

Toh, K. C., Todd, M. J., and Tütüncü, R. H. (1998). SDPT3 — a MATLAB software package for semidefinite programming. Technical report, Department of Mathematics, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260.

Tunçel, L. (1998). Primal-dual symmetry and scale invariance of interior-point algorithms for convex optimization. *Math. Oper. Res.*, 23(3):708–718.

Vandenberghe, L. and Boyd, S. (1995). A primal–dual potential reduction method for problems involving matrix inequalities. *Mathematical Programming, Series B*, 69(1):205–236.

Vandenberghe, L. and Boyd, S. (1996). Semidefinite programming. *SIAM Review*, 38(1):49–95.

Weismantel, R. (1997). On the 0/1 knapsack polytope. *Math. Programming*, 77:49–68.

Williams, A. (1985). Quadratic 0-1 programming using the roof dual with computational results. RUTCOR Research Report 8-85, Rutgers Unversity.

Wolkowicz, H. (1981). Some applications of optimization in matrix theory. *Linear Algebra and its Applications*, 40:101–118.

Wolkowicz, H., Saigal, R., and Vandenberghe, L., editors (2000). *Handbook of Semidefinite Programming*, volume 27 of *International Series in Operations Research and Management Science*. Kluwer Academic Publishers, Boston/Dordrecht/London.

Wolkowicz, H. and Zhao, Q. (1996). Semidefinite programming relaxations for the graph partitioning problem. CORR Report, University of Waterloo, Ontario, Canada.

Ye, Y. (1997a). Approximating quadratic optimization with linear and boolean constraints. Working paper, Department of Management Sciences, The University of Iowa, Iowa City, IA 52242, USA.

Ye, Y. (1997b). Approximating quadratic programming with bound constraints. Working paper, Department of Management Science, The University of Iowa, Iowa City, IA 52242, USA.

Ye, Y. (1997c). *Interior Point Algorithms, Theory and Analysis*. Interscience Series in Discrete Mathematics and Optimization. Wiley, New York.

Ye, Y. (1999a). A .699-approximation algorithm for max-bisection. Working note, Department of Management Sciences, Henry B. Tippie College of Business, The University of Iowa, Iowa City, IA 52242, USA.

Ye, Y. (1999b). Approximation quadratic programming with bound and quadratic constraints. *Math. Programming*, 84(2):219–226.

Zhang, Y. (1998). On extending some primal–dual interior-point algorithms from linear programming to semidefinite programming. *SIAM J. Optim.*, 8(2):365–386.

Zhao, Q., Karisch, S. E., Rendl, F., and Wolkowicz, H. (1998). Semidefinite programming relaxations for the quadratic assignment problem. *Journal of Combinatorial Optimization*, 2(1):71–109.

Zwick, U. (1998a). Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint. In *Proc. of 9th SODA*, pages 201–210.

Zwick, U. (1998b). Finding almost satisfying assignments. In *Proc. of 30th STOC*, pages 551–560.

Zwick, U. (1999). Outward rotations: a tool for rounding solutions of semidefinite programming relaxations, with applications to MAX CUT and other problems. In *Proc. of 31st STOC*.

Notation:

Sets and Matrices

$\mathbb{C}$	complex numbers
$\mathbb{N}$	nonnegative integers (including zero)
$\mathbb{Q}$	rational numbers
$\mathbb{R}$	real numbers
$\mathbb{Z}$	set of integers
$\mathbb{R}^n$	real column vector of dimension n
$\mathbb{R}^n_+$	nonnegative real column vector
$M_{m,n}$	$m \times n$ real matrices
M_n	$n \times n$ real matrices
$M_n(\mathbb{C})$	$n \times n$ complex matrices
S_n	$n \times n$ symmetric real matrices
S_n^{++}, $A \succ 0$	$n \times n$ symmetric positive definite matrices
S_n^{+}, $A \succeq 0$	$n \times n$ symmetric positive semidefinite matrices
I, I_n	identity of appropriate size or of size n
e	vector of all ones of appropriate dimension
e_i	i-th column of I
$\lambda_i(A)$	i-th eigenvalue of $A \in M_n$, usually $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$
$\lambda_{\min}(A)$, $\lambda_{\max}(A)$	minimal and maximal eigenvalue of A
Λ_A	diagonal matrix with $(\Lambda_A)_{ii} = \lambda_i(A)$
$\det(A)$	determinant of A
$\mathcal{N}(A)$	nullspace of A
$\mathcal{R}(A)$	range space of A
$\operatorname{rank}(A)$	rank of A
A^T	transpose of A
$\operatorname{tr}(A)$	trace of $A \in M_n$, $\operatorname{tr}(A) = \sum_{i=1}^n a_{ii} = \sum_{i=1}^n \lambda_i(A)$
$\langle A, B \rangle$	inner product in $M_{m,n}$, $\langle A, B \rangle = \operatorname{tr}(B^T A)$
$\|A\|_{\mathrm{F}}$	Frobenius norm of A, $\|A\|_{\mathrm{F}} = \sqrt{\langle A, A \rangle}$
$A \circ B$	Hadamard or Schur product of $A, B \in M_{m,n}$
$A \otimes B$	Kronecker product of $A \in M_{m,n}, B \in M_{k,l}$
$\operatorname{vec}(A)$	vector obtained by stacking the columns of A
$A \otimes_s B$	symmetric Kronecker product of $A, B \in M_n$
$\operatorname{svec}(A)$	vector obtained by stacking the columns of the lower triangle of $A \in S_n$ with offdiagonals multiplied by $\sqrt{2}$
$A_{i,\cdot}$ ($A_{\cdot,i}$)	i-th row (column) of A
$\operatorname{Diag}(v)$	diagonal matrix with v on its main diagonal
$\operatorname{diag}(A)$	the diagonal of $A \in M_n$ as a column vector
$\operatorname{sgn}(x)$	signum of x
$\operatorname{aff}(S)$	affine hull of a set S
$\operatorname{cone}(S)$	conic hull of a set S
$\operatorname{conv}(S)$	convex hull of a set S
$\operatorname{relint}(F)$	relative interior of a convex set F
∇_X	first derivative with respect to X
min / max	minimum/maximum value attained in an ordered set or minimize/maximize a program
inf / sup	infimum/supremum of an ordered set
argmin / argmax	minimizing/maximizing argument of a function
Argmin / Argmax	set of minimizing/maximizing arguments of a function

Graphs

$G = (V, E)$	graph with node set V and edge set E
V	set of nodes, usually $V = \{1, \ldots, n\}$
E	set of edges
ij	edge with end nodes i and j
$\delta(S)$	cut; the set of edges with exactly one endnode in $S \subseteq V$