LNMB PhD Course

Networks and Semidefinite Programming

2012/2013

Monique Laurent

CWI, Amsterdam, and Tilburg University

These notes are based on material developed by M. Laurent and F. Vallentin for the Mastermath course **Semidefinite Optimization**

Details at https://sites.google.com/site/mastermathsdp

February 4, 2013

CONTENTS

CHAPTER 1

POSITIVE SEMIDEFINITE MATRICES

In this chapter we collect basic facts about positive semidefinite matrices, which we will need in the next chapter to define semidefinite programs.

We use the following notation. Throughout $\|x\|$ denotes the Euclidean norm of $x \in \mathbb{R}^n$, defined by $\|x\| = \sqrt{x^{\mathsf{T}}x} = \sqrt{\sum_{i=1}^n x_i^2}$. An *orthonormal basis* of $\mathbb{R}^n$ is a set of unit vectors $\{u_1, \dots, u_n\}$ that are pairwise orthogonal: $\|u_i\| = 1$ for all i and $u_i^{\mathsf{T}}u_j = 0$ for all $i \neq j$. For instance, the standard unit vectors $e_1, \dots, e_n \in \mathbb{R}^n$ form an orthonormal basis. I_n denotes the $n \times n$ identity matrix and J_n denotes the all-ones matrix (we may sometimes omit the index n if the dimension is clear from the context). We let $\mathcal{S}^n$ denote the set of symmetric $n \times n$ matrices and $\mathcal{O}(n)$ denote the set of orthogonal matrices. A matrix $P \in \mathbb{R}^{n \times n}$ is *orthogonal* if $PP^{\mathsf{T}} = I_n$ or, equivalently, $P^{\mathsf{T}}P = I_n$, i.e. the rows (resp., the columns) of P form an orthonormal basis of $\mathbb{R}^n$. A diagonal matrix $D \in \mathcal{S}^n$ has entries zero at all off-diagonal positions: $D_{ij} = 0$ for all $i \neq j$.

1.1 Basic definitions

1.1.1 Characterizations of positive semidefinite matrices

We recall the notions of eigenvalues and eigenvectors. For a matrix $X \in \mathbb{R}^{n \times n}$, a nonzero vector $u \in \mathbb{R}^n$ is an *eigenvector* of X if there exists a scalar $\lambda \in \mathbb{R}$ such that $Xu = \lambda u$, then λ is the *eigenvalue* of X for the eigenvector u. A fundamental property of symmetric matrices is that they admit a set of eigenvectors $\{u_1, \dots, u_n\}$ forming an orthonormal basis of $\mathbb{R}^n$. This is the spectral decomposition theorem, one of the most important theorems about symmetric matrices.

Theorem 1.1.1. (Spectral decomposition theorem) *Any real symmetric matrix $X \in \mathcal{S}^n$ can be decomposed as*

$$X = \sum_{i=1}^{n} \lambda_i u_i u_i^{\mathsf{T}}, \tag{1.1}$$

where $\lambda_1, \ldots, \lambda_n \in \mathbb{R}$ are the eigenvalues of X and where $u_1, \ldots, u_n \in \mathbb{R}^n$ are the corresponding eigenvectors which form an orthonormal basis of $\mathbb{R}^n$. In matrix terms, $X = PDP^{\mathsf{T}}$, where D is the diagonal matrix with the λ_i's on the diagonal and P is the orthogonal matrix with the u_i's as its columns.

Next we define positive semidefinite matrices and give several equivalent characterizations.

Theorem 1.1.2. (Positive semidefinite matrices) *The following assertions are equivalent for a symmetric matrix $X \in \mathcal{S}^n$.*

(1) *X is* positive semidefinite*, written as $X \succeq 0$, which is defined by the property: $x^T X x \geq 0$ for all $x \in \mathbb{R}^n$.*

(2) *The smallest eigenvalue of X is nonnegative, i.e., the spectral decomposition of X is of the form $X = \sum_{i=1}^{n} \lambda_i u_i u_i^T$ with all $\lambda_i \geq 0$.*

(3) *$X = LL^{\mathsf{T}}$ for some matrix $L \in \mathbb{R}^{n \times k}$ (for some $k \geq 1$), called a* Cholesky decomposition *of X.*

(4) *There exist vectors $v_1, \ldots, v_n \in \mathbb{R}^k$ (for some $k \geq 1$) such that $X_{ij} = v_i^{\mathsf{T}} v_j$ for all $i, j \in [n]$; the vectors v_i's are called a* Gram representation *of X.*

(5) *All principal minors of X are non-negative.*

Proof. (i) $\Longrightarrow$ (ii): By assumption, $u_i^{\mathsf{T}} X u_i \geq 0$ for all $i \in [n]$. On the other hand, $Xu_i = \lambda_i u_i$ implies $u_i^{\mathsf{T}} X u_i = \lambda_i \|u_i\|^2 = \lambda_i$, and thus $\lambda_i \geq 0$ for all i.
(ii) $\Longrightarrow$ (iii): By assumption, X has a decomposition (1.1) where all scalars λ_i are nonnegative. Define the matrix $L \in \mathbb{R}^{n \times n}$ whose i-th column is the vector $\sqrt{\lambda_i} u_i$. Then $X = LL^{\mathsf{T}}$ holds.
(iii) $\Longrightarrow$ (iv): Assume $X = LL^{\mathsf{T}}$ where $L \in \mathbb{R}^{n \times k}$. Let $v_i \in \mathbb{R}^k$ denote the i-th row of L. The equality $X = LL^{\mathsf{T}}$ gives directly that $X_{ij} = v_i^{\mathsf{T}} v_j$ for all $i, j \in [n]$.
(iv) $\Longrightarrow$ (i): Assume $X_{ij} = v_i^{\mathsf{T}} v_j$ for all $i, j \in [n]$, where $v_1, \ldots, v_n \in \mathbb{R}^k$, and let $x \in \mathbb{R}^n$. Then, $x^{\mathsf{T}} X x = \sum_{i,j=1}^{n} x_i x_j X_{ij} = \sum_{i,j=1}^{n} x_i x_j v_i^{\mathsf{T}} v_j = \| \sum_{i=1}^{n} x_i v_i \|^2$ is thus nonnegative. This shows that $X \succeq 0$.
The equivalence (i) $\Longleftrightarrow$ (v) can be found in any standard Linear Algebra textbook (and will not be used here). □

Observe that for a diagonal matrix X, $X \succeq 0$ if and only if its diagonal entries are nonnegative: $X_{ii} \geq 0$ for all $i \in [n]$.

The above result extends to positive definite matrices. A matrix X is said to be *positive definite*, which is denoted as $X \succ 0$, if it satisfies any of the following equivalent properties: (1) $x^T X x > 0$ for all $x \in \mathbb{R}^n \setminus \{0\}$; (2) all eigenvalues of X are strictly positive; (3) in a Cholesky decomposition of X, the matrix L is nonsingular; (4) in any Gram representation of X as $(v_i^\mathsf{T} v_j)_{i,j=1}^n$, the system of vectors $\{v_1, \ldots, v_n\}$ has full rank n; and (5) all the principal minors of X are positive (in fact positivity of all the leading principal minors already implies positive definiteness, this is known as Sylvester's criterion).

1.1.2 The positive semidefinite cone $\mathcal{S}^n_{\succeq 0}$

We let $\mathcal{S}^n_{\succeq 0}$ denote the set of all positive semidefinite matrices in $\mathcal{S}^n$, called the *positive semidefinite cone*. Indeed, $\mathcal{S}^n_{\succeq 0}$ is a convex cone in $\mathcal{S}^n$, i.e., the following holds:

$$X, X' \succeq 0,\ \lambda, \lambda' \geq 0 \Longrightarrow \lambda X + \lambda' X' \succeq 0$$

(check it). Moreover, $\mathcal{S}^n_{\succeq 0}$ is a closed subset of $\mathcal{S}^n$. (Assume we have a sequence of matrices $X^{(i)} \succeq 0$ converging to a matrix X as $i \to \infty$ and let $x \in \mathbb{R}^n$. Then $x^\mathsf{T} X^{(i)} x \geq 0$ for all i and thus $x^\mathsf{T} X x \geq 0$ by taking the limit.) Moreover, as a direct application of (1.1), we find that the cone $\mathcal{S}^n_{\succeq 0}$ is generated by rank one matrices, i.e.,

$$\mathcal{S}^n_{\succeq 0} = \text{cone}\{xx^\mathsf{T} : x \in \mathbb{R}^n\}. \tag{1.2}$$

Furthermore, the cone $\mathcal{S}^n_{\succeq 0}$ is full-dimensional and the matrices lying in its interior are precisely the positive definite matrices.

1.1.3 The trace inner product

The *trace* of an $n \times n$ matrix A is defined as

$$\text{Tr}(A) = \sum_{i=1}^n A_{ii}.$$

Taking the trace is a linear operation:

$$\text{Tr}(\lambda A) = \lambda \text{Tr}(A),\ \text{Tr}(A + B) = \text{Tr}(A) + \text{Tr}(B).$$

Moreover, the trace satisfies the following properties:

$$\text{Tr}(A) = \text{Tr}(A^\mathsf{T}),\ \text{Tr}(AB) = \text{Tr}(BA),\ \text{Tr}(uu^\mathsf{T}) = u^\mathsf{T} u = \|u\|^2 \text{ for } u \in \mathbb{R}^n. \tag{1.3}$$

Using the fact that $\text{Tr}(uu^\mathsf{T}) = 1$ for any unit vector u, combined with (1.1), we deduce that the trace of a symmetric matrix is equal to the sum of its eigenvalues.

Lemma 1.1.3. *If $X \in \mathcal{S}^n$ has eigenvalues $\lambda_1, \ldots, \lambda_n$, then* $\text{Tr}(X) = \lambda_1 + \ldots + \lambda_n$.

One can define an inner product, denoted as $\langle \cdot, \cdot \rangle$, on $\mathbb{R}^{n\times n}$ by setting

$$\langle A, B\rangle = \mathrm{Tr}(A^{\mathsf{T}}B) = \sum_{i,j=1}^{n} A_{ij}B_{ij} \text{ for } A, B \in \mathbb{R}^{n\times n}. \tag{1.4}$$

This defines the *Frobenius norm* on $\mathbb{R}^{n\times n}$ by setting $\|A\| = \sqrt{\langle A, A\rangle} = \sqrt{\sum_{i,j=1}^{n} A_{ij}^2}$. In other words, this is the usual Euclidean norm, just viewing a matrix as a vector in $\mathbb{R}^{n^2}$. For a vector $x \in \mathbb{R}^n$ we have

$$\langle A, xx^{\mathsf{T}}\rangle = x^{\mathsf{T}}Ax.$$

The following property is useful to know:

Lemma 1.1.4. *Let $A, B \in \mathcal{S}^n$ and $P \in \mathcal{O}(n)$. Then, $\langle A, B\rangle = \langle PAP^{\mathsf{T}}, PBP^{\mathsf{T}}\rangle$.*

Proof. Indeed, $\langle PAP^{\mathsf{T}}, PBP^{\mathsf{T}}\rangle$ is equal to

$$\mathrm{Tr}(PAP^{\mathsf{T}}PBP^{\mathsf{T}}) = \mathrm{Tr}(PABP^{\mathsf{T}}) = \mathrm{Tr}(ABP^{\mathsf{T}}P) = \mathrm{Tr}(AB) = \langle A, B\rangle,$$

where we have used the fact that $P^{\mathsf{T}}P = PP^{\mathsf{T}} = I_n$ and the commutativity rule from (1.3). □

Positive semidefinite matrices satisfy the following fundamental property:

Lemma 1.1.5. *For a symmetric matrix $A \in \mathcal{S}^n$,*

$$A \succeq 0 \iff \langle A, B\rangle \geq 0 \text{ for all } B \in \mathcal{S}^n_{\succeq 0}.$$

Proof. The proof is based on the fact that $\mathcal{S}^n_{\succeq 0}$ is generated by rank 1 matrices (recall (1.2)). Indeed, if $A \succeq 0$ then $\langle A, xx^T\rangle = x^TAx \geq 0$ for all $x \in \mathbb{R}^n$, and thus $\langle A, B\rangle \geq 0$ for all $B \in \mathcal{S}^n_{\succeq 0}$. Conversely, if $\langle A, B\rangle \geq 0$ for all $B \in \mathcal{S}^n_{\succeq 0}$ then, for $B = xx^{\mathsf{T}}$, we obtain that $x^{\mathsf{T}}Ax \geq 0$, which shows $A \succeq 0$. □

In other words, the cone $\mathcal{S}^n_{\succeq 0}$ is *self dual*, i.e., it coincides with its dual cone[1].

1.2 Basic properties

1.2.1 Schur complements

We recall some basic operations about positive semidefinite matrices. The proofs of the following Lemmas 1.2.1, 1.2.2 and 1.2.3 are easy and left as an exercise.

Lemma 1.2.1. *If $X \succeq 0$ then every principal submatrix of X is positive semidefinite.*

[1] By definition, the dual of the cone $\mathcal{S}^n_{\succeq 0}$ is the set of all matrices $Y \in \mathcal{S}^n$ satisfying $\langle Y, X\rangle \geq 0$ for all $X \in \mathcal{S}^n_{\succeq 0}$.

Moreover, any matrix *congruent* to $X \succeq 0$ (i.e., of the form PXP^{T} where P is nonsingular) is positive semidefinite:

Lemma 1.2.2. *Let $P \in \mathbb{R}^{n\times n}$ be a nonsingular matrix. Then,*

$$X \succeq 0 \Longleftrightarrow PXP^{\mathsf{T}} \succeq 0.$$

Lemma 1.2.3. *Let $X \in \mathcal{S}^n$ be a matrix having the following block-diagonal form:*

$$X = \begin{pmatrix} A & 0 \\ 0 & C \end{pmatrix}.$$

Then,

$$X \succeq 0 \Longleftrightarrow A \succeq 0 \text{ and } B \succeq 0.$$

We now introduce the notion of *Schur complement*, which can be very useful for showing positive semidefiniteness.

Lemma 1.2.4. *Let $X \in \mathcal{S}^n$ be a matrix in block form*

$$X = \begin{pmatrix} A & B \\ B^{\mathsf{T}} & C \end{pmatrix}, \tag{1.5}$$

where $A \in \mathcal{S}^p$, $C \in \mathcal{S}^{n-p}$ and $B \in \mathbb{R}^{p\times(n-p)}$. If A is non-singular, then

$$X \succeq 0 \iff A \succeq 0 \text{ and } C - B^{\mathsf{T}}A^{-1}B \succeq 0.$$

The matrix $C - B^{\mathsf{T}}A^{-1}B$ is called the Schur complement *of A in X.*

Proof. One can verify that the following identity holds:

$$X = P^{\mathsf{T}} \begin{pmatrix} A & 0 \\ 0 & C - B^{\mathsf{T}}A^{-1}B \end{pmatrix} P, \quad \text{where } P = \begin{pmatrix} I & A^{-1}B \\ 0 & I \end{pmatrix}.$$

As P is nonsingular, we deduce that $X \succeq 0$ if and only if $(P^{-1})^{\mathsf{T}}XP^{-1} \succeq 0$ (use Lemma 1.2.2), which is thus equivalent to $A \succeq 0$ and $C - B^{\mathsf{T}}A^{-1}B \succeq 0$ (use Lemma 1.2.3). □

1.2.2 Kronecker and Hadamard products

Given two matrices $A = (A_{ij}) \in \mathbb{R}^{n\times m}$ and $B = (B_{hk}) \in \mathbb{R}^{p\times q}$, their *Kronecker product* is the matrix $A \otimes B \in \mathbb{R}^{np\times mq}$ with entries

$$A_{ih,jk} = A_{ij}B_{hk} \ \forall i \in [n], j \in [m], h \in [p], \ k \in [q].$$

The matrix $A \otimes B$ can be seen as the $n \times m$ block matrix whose ij-th block is the $p \times q$ matrix $A_{ij}B$ for all $i \in [n], j \in [m]$. Alternatively, it can be seen as the

$p \times q$ block matrix whose hk-block is the $n \times m$ matrix $B_{hk}A$ for $h \in [p], k \in [q]$. As an example, $I_2 \otimes J_3$ takes the form:

$$\begin{pmatrix} I_2 & I_2 & I_2 \\ I_2 & I_2 & I_2 \\ I_2 & I_2 & I_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix},$$

or, after permuting rows and columns, the form:

$$\begin{pmatrix} J_3 & 0 \\ 0 & J_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

This includes in particular defining the Kronecker product $u \otimes v \in \mathbb{R}^{np}$ of two vectors $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^p$, with entries $(u \otimes v)_{ih} = u_i v_h$ for $i \in [n], h \in [p]$.

Given two matrices $A, B \in \mathbb{R}^{n \times m}$, their *Hadamard product* is the matrix $A \circ B \in \mathbb{R}^{n \times m}$ with entries

$$(A \circ B)_{ij} = A_{ij} B_{ij} \ \forall i \in [n], j \in [m].$$

Note that $A \circ B$ coincides with the principal submatrix of $A \otimes B$ indexed by the subset of all 'diagonal' pairs of indices of the form (ii, jj) for $i \in [n], j \in [m]$.

Here are some (easy to verify) facts about these products, where the matrices and vectors have the appropriate sizes.

1. $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$.
2. In particular, $(A \otimes B)(u \otimes v) = (Au) \otimes (Bv)$.
3. Assume $A \in \mathcal{S}^n$ and $B \in \mathcal{S}^p$ have, respectively, eigenvalues $\alpha_1, \ldots, \alpha_n$ and $\beta_1, \ldots, \beta_p$. Then $A \otimes B \in \mathcal{S}^{np}$ has eigenvalues $\alpha_i \beta_h$ for $i \in [n], h \in [p]$. In particular,
$$A, B \succeq 0 \Longrightarrow A \otimes B \succeq 0 \ \text{ and } \ A \circ B \succeq 0,$$
$$A \succeq 0 \Longrightarrow A^{\circ k} = ((A_{ij})^k)_{i,j=1}^n \succeq 0 \ \forall k \in \mathbb{N}.$$

1.2.3 Properties of the kernel

Here is a first useful property of the kernel of positive semidefinite matrices.

Lemma 1.2.5. *Assume $X \in \mathcal{S}^n$ is positive semidefinite and let $x \in \mathbb{R}^n$. Then,*

$$Xx = 0 \Longleftrightarrow x^{\mathsf{T}} X x = 0.$$

Proof. The 'only if' part is clear. Conversely, decompose $x = \sum_{i=1}^n x_i u_i$ in the orthonormal base of eigenvectors of X. Then, $Xx = \sum_i \lambda_i x_i u_i$ and $x^\mathsf{T} Xx = \sum_i x_i^2 \lambda_i$. Hence, $0 = x^\mathsf{T} Xx$ gives $0 = \sum_i \lambda_i x_i^2$ and thus $x_i = 0$ for each i for which $\lambda_i > 0$. This shows that x is a linear combination of the eigenvectors u_i with eigenvalue $\lambda_i = 0$, and thus $Xx = 0$. □

Clearly, $X \succeq 0$ implies $X_{ii} \geq 0$ for all i (because $X_{ii} = e_i^\mathsf{T} X e_i \geq 0$). Moreover, if $X \succeq 0$ has a zero diagonal entry at position (i,i) then the whole i-th row/column is identically zero. This follows from the following property:

Lemma 1.2.6. *Let $X \in \mathcal{S}^n$ be a matrix in block form*

$$X = \begin{pmatrix} A & B \\ B^\mathsf{T} & C \end{pmatrix}, \tag{1.6}$$

where $A \in \mathcal{S}^p$, $C \in \mathcal{S}^{n-p}$ and $B \in \mathbb{R}^{p\times(n-p)}$. Assume $y \in \mathbb{R}^p$ belongs to the kernel of A, i.e., $Ay = 0$. Then the vector $x = (y, 0, \ldots, 0) \in \mathbb{R}^n$ (obtained from y by adding zero coordinates at the remaining $n - p$ positions) belongs to the kernel of X, i.e., $Xx = 0$.

Proof. We have: $x^\mathsf{T} Xx = u^\mathsf{T} Au = 0$ which, in view of Lemma 1.2.5, implies that $Xx = 0$. □

We conclude with the following property: The inner product of two positive semidefinite matrices is zero if and only if their matrix product is equal to 0.

Lemma 1.2.7. *Let $A, B \succeq 0$. Then,*

$$\langle A, B\rangle = 0 \Longleftrightarrow AB = 0.$$

Proof. The 'only if' part is clear since $\langle A, B\rangle = \text{Tr}(AB)$. Assume now $\langle A, B\rangle = 0$. Say, $B = \sum_{i=1}^n \lambda_i u_i u_i^\mathsf{T}$, where $\lambda_i \geq 0$ and the u_i form an orthonormal base. Then, $0 = \langle A, B\rangle = \sum_i \lambda_i \langle A, u_i u_i^\mathsf{T}\rangle$. This implies that each term $\lambda_i \langle A, u_i u_i^\mathsf{T}\rangle = \lambda_i u_i^\mathsf{T} A u_i$ is equal to 0, since $\lambda_i \geq 0$ and $u_i^\mathsf{T} A u_i \geq 0$ (as $A \succeq 0$). Hence, $\lambda_i > 0$ implies $u_i^\mathsf{T} A u_i = 0$ and thus $Au_i = 0$ (by Lemma 1.2.5). Therefore, each term $\lambda_i A u_i$ is equal to 0 and thus $AB = A(\sum_i \lambda_i u_i u_i^\mathsf{T}) = \sum_i \lambda_i A u_i u_i^\mathsf{T} = 0$. □

1.3 Exercises

1.1 Given $x_1, \ldots, x_n \in \mathbb{R}$, consider the following matrix

$$X = \begin{pmatrix} 1 & x_1 & \ldots & x_n \\ x_1 & x_1 & 0 & 0 \\ \vdots & 0 & \ddots & 0 \\ x_n & 0 & 0 & x_n \end{pmatrix}.$$

That is, $X \in \mathcal{S}^{n+1}$ is the matrix indexed by $\{0, 1, \ldots, n\}$, with entries $X_{00} = 1$, $X_{0i} = X_{i0} = X_{ii} = x_i$ for $i \in [n]$, and all other entries are equal to 0.

Show: $X \succeq 0$ if and only if $x_i \geq 0$ for all $i \in [n]$ and $\sum_{i=1}^n x_i \leq 1$.

Hint: Use Schur complements.

1.2. Define the matrix $F_{ij} = (e_i - e_j)(e_i - e_j)^\mathsf{T} \in \mathcal{S}^n$ for $1 \leq i < j \leq n$. That is, F_{ij} has entries 1 at positions (i, i) and (j, j), entries -1 at (i, j) and (j, i), and entries 0 at all other positions.

(a) Show: $F_{ij} \succeq 0$.

(b) Assume that $X \in \mathcal{S}^n$ satisfies the condition:

$$X_{ii} \geq \sum_{j \in [n] : j \neq i} |X_{ij}| \quad \text{for all } i \in [n].$$

(Then X is said to be *diagonally dominant.*)
Show: $X \succeq 0$.

1.3 Let $X \in \{\pm 1\}^{n \times n}$ be a symmetric matrix whose entries are 1 or -1.

Show: $X \succeq 0$ if and only if $X = xx^\mathsf{T}$ for some $x \in \{\pm 1\}^n$.

CHAPTER 2

SEMIDEFINITE PROGRAMS

Semidefinite programming is the analogue of linear programming but now, instead of having variables that are vectors assumed to lie in the nonnegative orthant $\mathbb{R}^n_{\geq 0}$, we have variables that are matrices assumed to lie in the cone $\mathcal{S}^n_{\succeq 0}$ of positive semidefinite matrices. Thus semidefinite optimization can be seen as linear optimization over the convex cone of positive semidefinite matrices.

In this chapter we introduce semidefinite programs and give some basic properties, in particular, about duality and complexity.

For convenience we recap some notation, mostly already introduced in the previous chapter. $\mathcal{S}^n$ denotes the set of symmetric $n \times n$ matrices. For a matrix $X \in \mathcal{S}^n$, $X \succeq 0$ means that X is positive semidefinite and $\mathcal{S}^n_{\succeq 0}$ is the cone of positive semidefinite matrices; $X \succ 0$ means that X is positive definite.

Throughout I_n (or simply I when the dimension is clear from the context) denotes the $n \times n$ identity matrix, e denotes the all-ones vector, i.e., $e = (1, \ldots, 1)^{\mathsf{T}} \in \mathbb{R}^n$, and $J_n = ee^{\mathsf{T}}$ (or simply J) denotes the all-ones matrix. The vectors $e_1, \ldots, e_n$ are the standard unit vectors in $\mathbb{R}^n$, and the matrices $E_{ij} = (e_i e_j^{\mathsf{T}} + e_j e_i^{\mathsf{T}})/2$ form the standard basis of $\mathcal{S}^n$. $\mathcal{O}(n)$ denotes the set of orthogonal matrices, where A is orthogonal if $AA^{\mathsf{T}} = I_n$ or, equivalently, $A^{\mathsf{T}}A = I_n$.

We consider the *trace inner product*: $\langle A, B\rangle = \mathrm{Tr}(A^{\mathsf{T}}B) = \sum_{i,j=1}^n A_{ij}B_{ij}$ for two matrices $A, B \in \mathbb{R}^{n\times n}$. Here $\mathrm{Tr}(A) = \langle I_n, A\rangle = \sum_{i=1}^n A_{ii}$ denotes the trace of A. Recall that $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$; in particular, $\langle QAQ^{\mathsf{T}}, QBQ^{\mathsf{T}}\rangle = \langle A, B\rangle$ if Q is an orthogonal matrix. A well known property of the positive semidefinite cone $\mathcal{S}^n_{\succeq 0}$ is that it is self-dual: for a matrix $X \in \mathcal{S}^n$, $X \succeq 0$ if and only if $\langle X, Y\rangle \geq 0$ for all $Y \in \mathcal{S}^n_{\succeq 0}$. For a matrix $A \in \mathcal{S}^n$, diag(A) denotes the vector in $\mathbb{R}^n$ with entries are the diagonal entries of A and, for a vector $a \in \mathbb{R}^n$, Diag$(a) \in \mathcal{S}^n$ is the diagonal matrix with diagonal entries the entries of a.

2.1 Semidefinite programs

2.1.1 Recap on linear programs

We begin with recalling the standard form of a linear program, in primal form:

$$p^* = \max_{x \in \mathbb{R}^n} \{c^\mathsf{T} x : a_j^\mathsf{T} x = b_j \ (j \in [m]), \ x \geq 0\}, \tag{2.1}$$

where $c, a_1, \ldots, a_m \in \mathbb{R}^n$ and $b = (b_j)_{j=1}^m \in \mathbb{R}^m$ are the given data of the LP. Then the dual LP reads:

$$d^* = \min_{y \in \mathbb{R}^m} \left\{ \sum_{j=1}^m b_j y_j : \sum_{j=1}^m y_j a_j - c \geq 0 \right\}. \tag{2.2}$$

We recall the following well known facts about LP duality:

Theorem 2.1.1. *The following holds for the programs (2.1) and (2.2).*

1. *(***weak duality***) If x is primal feasible and y is dual feasible then $c^\mathsf{T} x \leq b^\mathsf{T} y$. Thus, $p^* \leq d^*$.*

2. *(***strong duality***) $p^* = d^*$ unless both programs (2.1) and (2.2) are infeasible (in which case $p^* = -\infty$ and $d^* = +\infty$).*

   If p^* is finite (i.e., (2.1) is feasible and bounded) or if d^* is finite (i.e., (2.2) is feasible and bounded), then $p^* = d^*$ and both (2.1) and (2.2) have optimum solutions.

3. *(***optimality condition***) If (x, y) is a pair of primal/dual feasible solutions, then they are primal/dual optimal solutions if and only if $c^\mathsf{T} x = b^\mathsf{T} y$ or, equivalently, the* complementary slackness condition *holds:*

$$x_i \left(\sum_{j=1}^m y_j a_j - c \right)_i = 0 \ \forall i \in [n].$$

2.1.2 Semidefinite program in primal form

The standard form of a semidefinite program (abbreviated as SDP) is a maximization problem of the form

$$p^* = \sup_X \{\langle C, X \rangle : \langle A_j, X \rangle = b_j \ (j \in [m]), \ X \succeq 0\}. \tag{2.3}$$

Here $A_1, \ldots, A_m \in \mathcal{S}^n$ are given $n \times n$ symmetric matrices and $b \in \mathbb{R}^m$ is a given vector, they are the *data* of the semidefinite program (2.3). The matrix X is the *variable*, which is constrained to be positive semidefinite and to lie in the affine subspace

$$\mathcal{W} = \{X \in \mathcal{S}^n \mid \langle A_j, X \rangle = b_j \ (j \in [m])\}$$

of $\mathcal{S}^n$. The goal is to maximize the linear objective function $\langle C, X\rangle$ over the *feasible region*

$$\mathcal{F} = \mathcal{S}^n_{\succeq 0} \cap \mathcal{W},$$

obtained by intersecting the positive semidefinite cone $\mathcal{S}^n_{\succeq 0}$ with the affine subspace $\mathcal{W}$.

Of course, one can also handle minimization problems, of the form

$$\inf_X \{\langle C, X\rangle : \langle A_j, X\rangle = b_j \ (j \in [m]), \ X \succeq 0\}$$

since they can be brought into the above standard maximization form using the fact that $\inf\langle C, X\rangle = -\sup\langle -C, X\rangle$.

In the special case when the matrices A_j, C are diagonal matrices, with diagonals $a_j, c \in \mathbb{R}^n$, then the program (2.3) reduces to the linear program (2.1). Indeed, let x denote the vector consisting of the diagonal entries of the matrix X, so that $x \geq 0$ if $X \succeq 0$, and $\langle C, X\rangle = c^{\mathsf{T}}x$, $\langle A_j, X\rangle = a_j^{\mathsf{T}}x$. Hence semidefinite programming contains linear programming as a special instance.

A feasible solution $X \in \mathcal{F}$ is said to be *strictly feasible* if X is positive definite. The program (2.3) is said to be *strictly feasible* if it admits at least one strictly feasible solution.

Note that we write a *supremum* in (2.3) rather than a *maximum*. This is because the optimum value p^* might not be attained in (2.3). In general, $p^* \in \mathbb{R} \cup \{\pm\infty\}$, with $p^* = -\infty$ if the problem (2.3) is infeasible (i.e., $\mathcal{F} = \emptyset$) and $p^* = +\infty$ might occur in which case we say that the problem is unbounded.

We give a small example as an illustration.

Example 2.1.2. *Consider the problem of minimizing/maximizing X_{11} over the feasible region*

$$\mathcal{F}_a = \left\{X \in \mathcal{S}^2 : X = \begin{pmatrix} X_{11} & a \\ a & 0 \end{pmatrix} \succeq 0\right\} \text{ where } a \in \mathbb{R} \text{ is a given parameter.}$$

Note that $\det(X) = -a^2$ for any $X \in \mathcal{F}_a$. Hence, if $a \neq 0$ then $\mathcal{F}_a = \emptyset$ (the problem is infeasible). Moreover, if $a = 0$ then the problem is feasible but not strictly feasible. The minimum value of X_{11} over $\mathcal{F}_0$ is equal to 0, attained at $X = 0$, while the maximum value of X_{11} over $\mathcal{F}_0$ is equal to ∞ (the problem is unbounded).

Example 2.1.3. *As another example, consider the problem*

$$p^* = \inf_{X \in \mathcal{S}^2} \left\{X_{11} : \begin{pmatrix} X_{11} & 1 \\ 1 & X_{22} \end{pmatrix} \succeq 0\right\}. \tag{2.4}$$

Then the infimum is $p^* = 0$ which is reached at the limit when $X_{11} = 1/X_{22}$ and letting X_{22} tend to ∞. So the infimum is not attained.

2.1.3 Semidefinite program in dual form

The program (2.3) is often referred to as the *primal SDP* in standard form. One can define its *dual SDP*, which takes the form:

$$d^* = \inf_y \left\{ \sum_{j=1}^m b_j y_j = b^\mathsf{T} y : \sum_{j=1}^m y_j A_j - C \succeq 0 \right\}. \tag{2.5}$$

Thus the dual program has variables y_j, one for each linear constraint of the primal program. The positive semidefinite constraint arising in (2.5) is also named a *linear matrix inequality (LMI)*. The SDP (2.5) is said to be *strictly feasible* if it has a feasible solution y for which $\sum_j y_j A_j - C \succ 0$.

Example 2.1.4. *Let us work out the dual SDP of the SDP in Example 2.1.3. First we write (2.4) in standard primal form as*

$$-p^* = \max_{X \in \mathcal{S}^2} \{ \langle -E_{11}, X \rangle : \langle E_{12}, X \rangle = 2 \}. \tag{2.6}$$

As there is one linear equation, there is one dual variable y and the dual SDP reads:

$$-d^* = \inf_{y \in \mathbb{R}} \{ 2y : yE_{12} + E_{11} = \begin{pmatrix} 1 & y \\ y & 0 \end{pmatrix} \succeq 0 \}. \tag{2.7}$$

Hence $y = 0$ is the only dual feasible solution. Hence, the dual optimum value is $d^* = 0$, attained at $y = 0$.

2.1.4 Duality

The following facts relate the primal and dual SDP's. They are simple, but very important.

Lemma 2.1.5. *Let X be a feasible solution of (2.3) and let y be a feasible solution of (2.5). Then the following holds.*

1. **(weak duality)** *We have: $\langle C, X \rangle \leq b^\mathsf{T} y$ and thus $p^* \leq d^*$.*
2. **(optimality condition)** *Assume that $p^* = d^*$ holds. Then X is an optimal solution of (2.3) and y is an optimal solution of (2.5) if and only if equality: $\langle C, X \rangle = b^\mathsf{T} y$ holds or, equivalently, $\langle X, \sum_{j=1}^m y_j A_j - C \rangle = 0$ which, in turn, is equivalent to the following complementarity condition:*

$$X \left(\sum_{j=1}^m y_j A_j - C \right) = 0.$$

Proof. Let (X, y) is a primal/dual pair of feasible solutions.
1. We have:

$$\langle X, \sum_j y_j A_j - C \rangle = \sum_j \langle X, A_j \rangle y_j - \langle X, C \rangle = \sum_j b_j y_j - \langle X, C \rangle = b^\mathsf{T} y - \langle C, X \rangle, \tag{2.8}$$

where we used the fact that $\langle A_j, X\rangle = b_j$ to get the second equality. As both X and $\sum_j y_j A_j - C$ are positive semidefinite, we get: $\langle X, \sum_j y_j A_j - C\rangle \geq 0$, which implies $\langle C, X\rangle \leq b^{\mathsf{T}} y$ and thus $p^* \leq d^*$.
2. By assumption, we have: $\langle C, X\rangle \leq p^* = d^* \leq b^{\mathsf{T}} y$. Hence, (X, y) form a pair of primal/dual optimal solutions if and only if $\langle C, X\rangle = b^{\mathsf{T}} y$ or, equivalently (in view of relation (2.8)), $\langle X, \sum_j y_j A_j - C\rangle = 0$. Finally, as both X and $Z = \sum_j y_j A_j - C$ are positive semidefinite, we deduce that $\langle X, Z\rangle = 0$ if and only if $XZ = 0$. (Recall Lemma 1.2.7.) □

The quantity $d^* - p^*$ is called the *duality gap*. While there is no duality gap in LP, there might be a positive duality gap between the primal and dual SDP's. When there is no duality gap, i.e., when $p^* = d^*$, one says that *strong duality* holds. Having strong duality is a very desirable situation, which happens when at least one of the primal and dual semidefinite programs is strictly feasible. We only quote the following result on strong duality. For its proof we refer e.g. to the textbook [1] or to [3].

Theorem 2.1.6. (Strong duality: no duality gap) *Consider the pair of primal and dual programs (2.3) and (2.5).*

1. *Assume that the dual program (2.5) is bounded from below ($d^* > -\infty$) and that it is strictly feasible. Then the primal program (2.3) attains its supremum (i.e., $p^* = \langle C, X\rangle$ for some primal feasible X) and there is no duality gap: $p^* = d^*$.*

2. *Assume that the primal program (2.3) is bounded from above ($p^* < \infty$) and that it is strictly feasible. Then the dual program (2.5) attains its infimum (i.e., $d^* = b^{\mathsf{T}} y$ for some dual feasible y) and there is no duality gap: $p^* = d^*$.*

Consider again the primal and dual SDP's of Example 2.1.4. Then, the primal (2.6) is strictly feasible, the dual (2.7) attains its optimum value and there is no duality gap, while the dual is not strictly feasible and the primal does not attain its optimum value.

We conclude with an example having a positive duality gap.

Example 2.1.7. *Consider the primal semidefinite program with data matrices*

$$C = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, A_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix},$$

and $b_1 = 0$, $b_2 = 1$. It reads

$$p^* = \sup\{-X_{11} - X_{22} : X_{11} = 0,\ 2X_{13} + X_{22} = 1,\ X \succeq 0\}$$

and its dual reads

$$d^* = \inf\left\{ y_2 : y_1 A_1 + y_2 A_2 - C = \begin{pmatrix} y_1 + 1 & 0 & y_2 \\ 0 & y_2 + 1 & 0 \\ y_2 & 0 & 0 \end{pmatrix} \succeq 0 \right\}.$$

Then any primal feasible solution satisfies $X_{13} = 0$, $X_{22} = 1$, so that the primal optimum value is equal to $p^* = -1$, attained at the matrix $X = E_{22}$. Any dual feasible solution satisfies $y_2 = 0$, so that the dual optimum value is equal to $d^* = 0$, attained at $y = 0$. Hence there is a positive duality gap: $d^* - p^* = 1$.

Note that in this example both the primal and dual programs are not strictly feasible.

2.2 Application to eigenvalue optimization

Given a matrix $C \in \mathcal{S}^n$, let $\lambda_{\min}(C)$ (resp., $\lambda_{\max}(C)$) denote its smallest (resp., largest) eigenvalue. One can express them (please check it) as follows:

$$\lambda_{\max}(C) = \max_{x \in \mathbb{R}^n \setminus \{0\}} \frac{x^{\mathsf{T}} C x}{\|x\|^2} = \max_{x \in \mathbb{S}^{n-1}} x^{\mathsf{T}} C x, \tag{2.9}$$

where $\mathbb{S}^{n-1} = \{x \in \mathbb{R}^n \mid \|x\|x = 1\}$ denotes the unit sphere in $\mathbb{R}^n$, and

$$\lambda_{\min}(C) = \min_{x \in \mathbb{R}^n \setminus \{0\}} \frac{x^{\mathsf{T}} C x}{\|x\|^2} = \min_{x \in \mathbb{S}^{n-1}} x^{\mathsf{T}} C x. \tag{2.10}$$

(This is known as the Rayleigh principle.) As we now see the largest and smallest eigenvalues can be computed via a semidefinite program. Namely, consider the semidefinite program

$$p^* = \sup \{\langle C, X\rangle : \mathrm{Tr}(X) = \langle I, X\rangle = 1, X \succeq 0\} \tag{2.11}$$

and its dual program

$$d^* = \inf_{y \in \mathbb{R}} \{y : yI - C \succeq 0\}. \tag{2.12}$$

In view of (2.9), we have that $d^* = \lambda_{\max}(C)$. The feasible region of (2.11) is bounded (all entries of any feasible X lie in $[0, 1]$) and contains a positive definite matrix (e.g., the matrix I_n/n), hence the infimum is attained in (2.12). Analogously, the program (2.12) is bounded from below (as $y \geq \lambda_{\max}(C)$ for any feasible y) and strictly feasible (pick y large enough), hence the infimum is attained in (2.12). Moreover there is no duality gap: $p^* = d^*$. Here we have applied Theorem 2.1.6. Thus we have shown:

Lemma 2.2.1. *The largest and smallest eigenvalues of a symmetric matrix $C \in \mathcal{S}^n$ can be expressed with the following semidefinite programs:*

$$\begin{array}{llllll} \lambda_{\max}(C) = & \max & \langle C, X\rangle & = & \min & y \\ & \textit{s.t.} & \mathrm{Tr}(X) = 1, X \succeq 0 & & \textit{s.t.} & yI_n - C \succeq 0 \end{array}$$

$$\begin{array}{llllll} \lambda_{\min}(C) = & \min & \langle C, X\rangle & = & \max & y \\ & \textit{s.t.} & \mathrm{Tr}(X) = 1, X \succeq 0 & & \textit{s.t.} & C - yI_n \succeq 0 \end{array}$$

More generally, also the sum of the k largest eigenvalues of a symmetric matrix can be computed via a semidefinite program. For details see [4].

Theorem 2.2.2. (Fan's theorem) *Let $C \in \mathcal{S}^n$ be a symmetric matrix with eigenvalues $\lambda_1 \geq \ldots \geq \lambda_n$. Then the sum of its k largest eigenvalues is given by any of the following two programs:*

$$\lambda_1 + \cdots + \lambda_k = \max_{X \in \mathcal{S}^n} \left\{ \langle C, X \rangle : \mathrm{Tr}(X) = k,\ I_n \succeq X \succeq 0 \right\}, \tag{2.13}$$

$$\lambda_1 + \cdots + \lambda_k = \max_{Y \in \mathbb{R}^{n \times k}} \left\{ \langle C, YY^\mathsf{T} \rangle : Y^\mathsf{T} Y = I_k \right\}. \tag{2.14}$$

2.3 Some facts about complexity

2.3.1 More differences between LP and SDP

We have already seen above several differences between linear programming and semidefinite programming: there might be a duality gap between the primal and dual programs and the supremum/infimum might not be attained even though they are finite. We point out some more differences regarding rationality and bit size of optimal solutions.

In the classical bit (Turing machine) model of computation an integer number p is encoded in binary notation, so that its bit size is $\log p + 1$ (logarithm in base 2). Rational numbers are encoded as two integer numbers and the bit size of a vector or a matrix is the sum of the bit sizes of its entries.

Consider a linear program

$$\max\{c^\mathsf{T} x : Ax = b, x \geq 0\} \tag{2.15}$$

where the data A, b, c is *rational* valued. From the point of view of computability this is a natural assumption and it would be desirable to have an optimal solution which is also rational-valued. A fundamental result in linear programming asserts that this is indeed the case: If program (4.4) has an optimal solution, then it has a *rational* optimal solution $x \in \mathbb{Q}^n$, whose bit size is polynomially bounded in terms of the bit sizes of A, b, c (see e.g. [7]).

On the other hand it is easy to construct instances of semidefinite programming where the data are rational valued, yet there is no rational optimal solution. For instance, the following program

$$\max \left\{ x : \begin{pmatrix} 1 & x \\ x & 2 \end{pmatrix} \succeq 0 \right\} \tag{2.16}$$

attains its maximum at $x = \pm\sqrt{2}$.

Consider now the semidefinite program, with variables $x_1, \ldots, x_n$,

$$\inf \left\{ x_n : \begin{pmatrix} 1 & 2 \\ 2 & x_1 \end{pmatrix} \succeq 0, \begin{pmatrix} 1 & x_{i-1} \\ x_{i-1} & x_i \end{pmatrix} \succeq 0 \text{ for } i = 2, \ldots, n \right\}. \tag{2.17}$$

Then any feasible solution satisfies $x_n \geq 2^{2^n}$. Hence the bit-size of an optimal solution is exponential in n, thus exponential in terms of the bit-size of the data.

2.3.2 Algorithms

It is well known that linear programs (with rational data $c, a_1, \ldots, a_m, b$) can be solved in polynomial time. Although the simplex method invented by Dantzig in 1948 performs very well in practice, it is still an open problem whether it gives a polynomial time algorithm for solving general LP's. The first polynomial-time algorithm for solving LP's was given by Khachiyan in 1979, based on the ellipsoid method. The value of this algorithm is however mainly theoretical as it is very slow in practice. Later the algorithm of Karmarkar in 1984 opened the way to polynomial time algorithms for LP based on interior-point algorithms, which also perform well in practice.

What about algorithms for solving semidefinite programs?

First of all, one cannot hope for a polynomial time algorithm permitting to solve any semidefinite program *exactly*. Indeed, even if the data of the SDP are assumed to be rational valued, the output might be an irrational number, thus not representable in the bit model of computation. Such an instance was mentioned above in (2.16). Therefore, one can hope at best for an algorithm permitting to compute in polynomial time an ϵ-approximate optimal solution.

However, even if we set up to this less ambitious goal of just computing ϵ-approximate optimal solutions, we should make some assumptions on the semidefinite program, roughly speaking, in order to avoid having too large or too small optimal solutions. An instance of SDP whose output is exponentially large in the bit size of the data was mentioned above in (2.17).

On the positive side, it is well known that one can test whether a given rational matrix is positive semidefinite in polynomial time — using Gaussian elimination. Hence one can test in polynomial time membership in the positive semidefinite cone and, moreover, if $X \notin \mathcal{S}^n_{\succeq 0}$, then one can compute in polynomial time a hyperplane strictly separating X from $\mathcal{S}^n_{\succeq 0}$ (again as a byproduct of Gaussian elimination). See Section **??** below for details.

This observation is at the base of the polynomial time algorithm for solving approximately semidefinite programs, based on the ellipsoid method. Roughly speaking, one can solve a semidefinite program in polynomial time up to any given precision. More precisely, we quote the following result describing the complexity of solving semidefinite programming with the ellipsoid method:

Consider the semidefinite program

$$p^* = \sup\{\langle C, X\rangle : \langle A_j, X\rangle = b_j \ (j \in [m]),\ X \succeq 0\},$$

where A_j, C, b_j are integer valued. Denote by $\mathcal{F}$ its feasibility region. Suppose that an integer R is known a priori such that either $\mathcal{F} = \emptyset$ or there exists $X \in \mathcal{F}$ with $\|X\| \leq R$. Let $\epsilon > 0$ be given.Then, either one can find a matrix X^* at distance at most ϵ from $\mathcal{F}$ and such that $|\langle C, X^*\rangle - p^*| \leq \epsilon$, or one can find a certificate that $\mathcal{F}$ does not contain a ball of radius ϵ. The complexity of this algorithm is polynomial in n, m, $\log R$, $\log(1/\epsilon)$, and the bit size of the input data.

Again, although polynomial time in theory, algorithms based on the ellipsoid method are not practical. Instead, interior-point algorithms are used to solve semidefinite programs in practice. We refer e.g. to [1], [2], [7], [6] for more information about algorithms for linear and semidefinite programming.

2.3.3 Gaussian elimination

Let $A = (a_{ij}) \in \mathcal{S}^n$ be a rational matrix. Gaussian elimination permits to do the following tasks in polynomial time:

(i) Either: find a rational matrix $U \in \mathbb{Q}^{n\times n}$ and a rational diagonal matrix $D \in \mathbb{Q}^{n\times n}$ such that $A = UDU^{\mathsf{T}}$, thus showing that $A \succeq 0$.

(ii) Or: find a rational vector $x \in \mathbb{Q}^n$ such that $x^{\mathsf{T}}Ax < 0$, thus showing that A is not positive semidefinite and giving a hyperplane separating A from the cone $\mathcal{S}^n_{\succeq 0}$.

Here is a sketch. We distinguish three cases.

Case 1: $a_{11} < 0$. Then, (ii) applies, since $e_1^{\mathsf{T}}Ae_1 < 0$.

Case 2: $a_{11} = 0$, but some entry a_{1j} is not zero, say $a_{12} \neq 0$. Then choose $\lambda \in \mathbb{Q}$ such that $2\lambda a_{12} + a_{22} < 0$, so that $x^{\mathsf{T}}Ax < 0$ for the vector $x = (\lambda, 1, 0, \dots, 0)$ and thus (ii) applies again.

Case 3: $a_{11} > 0$. Then we apply Gaussian elimination to the rows R_j and columns C_j of A for $j = 2, \dots, n$. Namely, for each $j = 2, \dots, n$, we replace C_j by $C_j - \frac{a_{1j}}{a_{11}}C_1$, and analogously we replace R_j by $R_j - \frac{a_{12}}{a_{11}}R_j$, which amounts to making all entries of A equal to zero at the positions $(1, j)$ and $(j, 1)$ for $j \neq 1$.

For this, define the matrices $P_j = I_n - \frac{a_{1j}}{a_{11}}E_{1j}$ and $P = P_2 \cdots P_n$. Then, P is rational and nonsingular, and $P^{\mathsf{T}}AP$ has the block form:

$$P^{\mathsf{T}}AP = \begin{pmatrix} 1 & 0 \\ 0 & A' \end{pmatrix},$$

where $A' \in \mathcal{S}^{n-1}$. Thus, $A \succeq 0 \Longleftrightarrow P^{\mathsf{T}}AP \succeq 0 \Longleftrightarrow A' \succeq 0$.

Then, we proceed inductively with the matrix $A' \in \mathcal{S}^{n-1}$:

- Either, we find $W \in \mathbb{Q}^{(n-1)\times(n-1)}$ and a diagonal matrix $D' \in \mathbb{Q}^{(n-1)\times(n-1)}$ such that $A' = W^{\mathsf{T}}D'W$. Then, we obtain that $A = U^{\mathsf{T}}DU$, setting

$$U = \begin{pmatrix} 1 & 0 \\ 0 & W \end{pmatrix} P^{-1},\ D = \begin{pmatrix} 1 & 0 \\ 0 & D' \end{pmatrix}.$$

- Or, we find $y \in \mathbb{Q}^{n-1}$ such that $y^{\mathsf{T}}A'y < 0$. Then, we obtain that $x^{\mathsf{T}}Ax < 0$, after defining $z = (0, y)$ and $x = Pz \in \mathbb{Q}^n$.

2.4 Exercises

2.1. Let $G = (V = [n], E)$ be a graph and let $d = (d_{ij})_{\{i,j\} \in E} \in \mathbb{R}^E_{\geq 0}$ be given nonnegative weights on the edges. Consider the following problem (P):

Find vectors $v_1, \ldots, v_n \in \mathbb{R}^k$ (for some integer $k \geq 1$) such that

$$\sum_{i=1}^n \|v_i\|^2 = 1,\ \|v_i - v_j\|^2 = d_{ij} \text{ for all } \{i,j\} \in E$$

and for which the sum $\sum_{i,j=1}^n v_i^{\mathsf{T}} v_j$ is minimum.

(a) Formulate problem (P) as an instance of semidefinite program.

(b) If in problem (P) we add the additional constraint that the vectors $v_1, \ldots, v_n$ should belong to $\mathbb{R}^k$ for some *fixed dimension k*, how would you translate this condition on the semidefinite program?

BIBLIOGRAPHY

[1] A. Ben-Tal, A. Nemirovski. *Lectures on Modern Convex Optimization*, SIAM, 2001.

[2] M. Grötschel, L. Lovász and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, 1988.

[3] M. Laurent and A. Vallentin. *Semidefinite Optimization*. Lecture Notes.

`https://sites.google.com/site/mastermathsdp/lectures`

[4] M. Overton and R.S. Womersley. On the sum of the k largest eigenvalues of a symmetric matrix. *SIAM Journal on Matrix Analysis and its Applications* **13(1)**:41–45, 1992.

[5] A. Schrijver, *Theory of linear and integer programming*, John Wiley & Sons, 1986.

[6] L. Vandenberghe and S. Boyd. Semidefinite programming. SIAM Review **38**:49–95, 1996.

`http://stanford.edu/~boyd/papers/sdp.html`

CHAPTER 3

GRAPH COLORING AND INDEPENDENT SETS

In this chapter we discuss how semidefinite programming can be used for constructing tractable bounds for two hard combinatorial problems: for finding maximum independent sets and minimum colorings in graphs.

We introduce the graph parameter $\vartheta(G)$, known as the *theta number* of the graph G. This parameter was introduced by L. Lovász in his seminal paper [7]. We present several equivalent formulations and explain how $\vartheta(G)$ can be used to compute maximum stable sets and minimum colorings in perfect graphs in polynomial time, whereas these problems are NP-hard for general graphs.

Here are some definitions that we use in this chapter. Let $G = (V, E)$ be a graph; often we let $V = [n] = \{1, \ldots, n\}$. Then, $\overline{E}$ denotes the set of pairs $\{i, j\}$ of distinct nodes that are not adjacent in G. The graph $\overline{G} = (V, \overline{E})$ is called the *complementary graph* of G. G is *self-complementary* if G and $\overline{G}$ are isomorphic graphs. Given a subset $S \subseteq V$, $G[S]$ denotes the *subgraph induced by* S: its node set is S and its edges are all pairs $\{i, j\} \in E$ with $i, j \in S$. The graph C_n is the circuit (or cycle) of length n, with node set $[n]$ and edges the pairs $\{i, i+1\}$ (for $i \in [n]$, indices taken modulo n). For a set $S \subseteq V$, its *characteristic vector* is the vector $\chi^S \in \{0, 1\}^V$, whose i-th entry is 1 if $i \in S$ and 0 otherwise. We let $e = (1, \ldots, 1)^{\mathsf{T}}$ denote the all-ones vector.

3.1 Preliminaries on graphs

3.1.1 Stability and chromatic numbers

A subset $S \subseteq V$ of nodes is said to be *stable* (or *independent*) if no two nodes of S are adjacent in G. Then the *stability number* of G is the parameter $\alpha(G)$ defined as the maximum cardinality of an independent set in G.

A subset $C \subseteq V$ of nodes is called a *clique* if every two distinct nodes in C are adjacent. The maximum cardinality of a clique in G is denoted $\omega(G)$, the *clique number* of G. Clearly,

$$\omega(G) = \alpha(\overline{G}).$$

Computing the stability number of a graph is a hard problem: Given a graph G and an integer k, deciding whether $\alpha(G) \geq k$ is an $\mathcal{N}P$-complete problem.

Given an integer $k \geq 1$, a k-*coloring* of G is an assignment of numbers (view them as *colors*) from $\{1, \cdots, k\}$ to the nodes in such a way that two adjacent nodes receive distinct colors. In other words, this corresponds to a partition of V into k stable sets: $V = S_1 \cup \cdots \cup S_k$, where S_i is the stable set consisting of all nodes that received the i-th color. The *coloring* (or *chromatic*) *number* is the smallest integer k for which G admits a k-coloring, it is denoted as $\chi(G)$.

Again it is an $\mathcal{N}P$-complete problem to decide whether a graph is k-colorable. In fact, it is $\mathcal{N}P$-complete to decide whether a planar graph is 3-colorable. On the other hand, it is known that every planar graph is 4-colorable – this is the celebrated 4-color theorem. Moreover, observe that one can decide in polynomial time whether a graph is 2-colorable, since one can check in polynomial time whether a graph is bipartite.

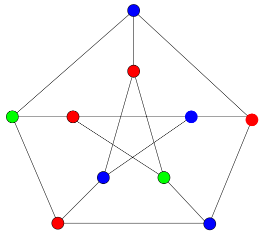


Figure 3.1: The Petersen graph has $\alpha(G) = 4$, $\omega(G) = 2$ and $\chi(G) = 3$

Clearly, any two nodes in a clique of G must receive distinct colors. Therefore, for any graph, the following inequality holds:

$$\omega(G) \leq \chi(G). \tag{3.1}$$

This inequality is strict, for example, when G is an odd circuit, i.e., a circuit of odd length at least 5, or its complement. Indeed, for an odd circuit C_{2n+1} ($n \geq 2$), $\omega(C_{2n+1}) = 2$ while $\chi(C_{2n+1}) = 3$. Moreover, for the complement

$G = \overline{C_{2n+1}}$, $\omega(G) = n$ while $\chi(G) = n + 1$. For an illustration see the cycle of length 7 and its complement in Figure 6.2.

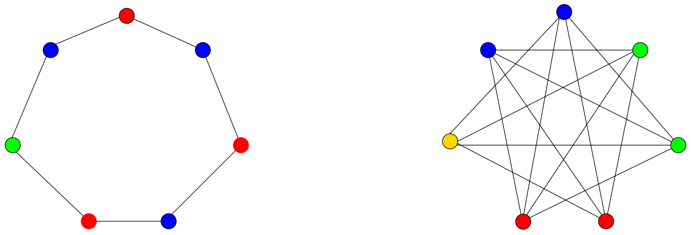


Figure 3.2: For C_7 and its complement $\overline{C_7}$: $\omega(C_7) = 2$, $\chi(C_7) = 3$, $\omega(\overline{C_7}) = \alpha(C_7) = 3$, $\chi(\overline{C_7}) = 4$

3.1.2 Perfect graphs

It is intriguing to understand for which graphs equality $\omega(G) = \chi(G)$ holds. Note that any graph G with $\omega(G) < \chi(G)$ can be embedded in a larger graph $\hat{G}$ with $\omega(\hat{G}) = \chi(\hat{G})$, simply by adding to G a clique of size $\chi(G)$ (disjoint from V). This justifies the following definition, introduced by C. Berge in the early sixties, which makes the problem well posed.

Definition 3.1.1. *A graph G is said to be* perfect *if equality*

$$\omega(H) = \chi(H)$$

holds for all induced subgraphs H of G (including $H = G$).

Here are some classes of perfect graphs. For each of them the relation $\omega(G) = \chi(G)$ gives a combinatorial min-max relation.

1. Bipartite graphs (the relation $\omega(G) = \chi(G) = 2$ is clear).

2. Line graphs of bipartite graphs (the min-max relation claims that the maximum cardinality of a matching is equal to the minimum cardinality of a vertex cover, which is König's theorem).

3. Comparability graphs (the min-max relation corresponds to Diilworth's theorem).

It follows from the definition and the above observation about odd circuits that if G is a perfect graph then it does not contain an odd circuit of length at least 5 or its complement as an induced subgraph. Berge already conjectured that *all* perfect graphs arise in this way. Resolving this conjecture has haunted generations of graph theorists. It was finally settled in 2002 by Chudnovsky, Robertson, Seymour and Thomas who proved the following result, known as the *strong perfect graph theorem*:

Theorem 3.1.2. (The strong perfect graph theorem)*[1] A graph G is perfect if and only if it does not contain an odd circuit of length at least 5 or its complement as an induced subgraph.*

This implies the following structural result about perfect graphs, known as the *perfect graph theorem*, already proved by Lovász in 1972.

Theorem 3.1.3. (The perfect graph theorem)*[6] If G is a perfect graph, then its complement $\overline{G}$ too is a perfect graph.*

We give a direct proof of Theorem 3.1.3 in the next section and we will mention later some other, more geometric, characterizations of perfect graphs (see, e.g., Theorem 3.2.5).

3.1.3 The perfect graph theorem

Lovász [6] proved the following result, which implies the perfect graph theorem (Theorem 3.1.3). The proof given below follows the elegant linear-algebraic argument of Gasparian [3].

Theorem 3.1.4. *A graph G is perfect if and only if $|V(G')| \leq \omega(G')\chi(G')$ for each induced subgraph G' of G.*

Proof. Necessity is easy: Assume that G is perfect and let G' be an induced subgraph of G. Then $\chi(G') = \omega(G')$ and thus $V(G')$ can be covered by $\omega(G')$ stable sets, which implies that $|V(G')| \leq \omega(G')\alpha(G')$.

To show sufficiency, assume for a contradiction that there exists a graph G which satisfies the condition but is not perfect; choose such a graph with $|V(G)|$ minimal. Then, $n \leq \alpha(G)\omega(G)$, $\omega(G) < \chi(G)$ and $\omega(G') = \chi(G')$ for each induced subgraph $G' \neq G$ of G. Set $\omega = \omega(G)$ and $\alpha = \alpha(G)$ for simplicity. Our first claim is:

Claim 1: There exist $\alpha\omega + 1$ stable sets $S_0, \ldots, S_{\alpha\omega}$ such that each vertex of G is covered by exactly α of them.

Proof of the claim: Let S_0 be a stable set of size α in G. For each node $v \in S_0$, as $G \backslash v$ is perfect (by the minimality assumption on G), $\chi(G \backslash v) = \omega(G \backslash v) \leq \omega$. Hence, $V \setminus \{v\}$ can be partitioned into ω stable sets. In this way we obtain a collection of $\alpha\omega$ stable sets which together with S_0 satisfy the claim. □

Our next claim is:

Claim 2: For each $i = 0, 1, \ldots, \alpha\omega$, there exists a clique K_i of size ω such that $K_i \cap S_i = \emptyset$ and $K_i \cap S_j \neq \emptyset$ for $j \neq i$.

Proof of the claim: For each $i = 0, 1, \ldots, \alpha\omega$, as $G \setminus S_i$ is perfect we have that $\chi(G \backslash S_i) = \omega(S_i) \leq \omega$. This implies that $\chi(G \backslash S_i) = \omega$ since, if $\chi(G \backslash S_i) \leq \omega - 1$, then one could color G with ω colors, contradicting our assumption on G. Hence there exists a clique K_i disjoint from S_i and with $|K_i| = \omega$. Moreover K_i meets all the other $\alpha\omega$ stable sets S_j for $j \neq i$. This follows from the fact that each of the ω elements of K_i belongs to α stable sets among the S_j's (Claim 1) and these $\omega\alpha$ sets are pairwise distinct. $\square$

We can now conclude the proof. Define the matrices $M, N \in \mathbb{R}^{n\times(\alpha\omega+1)}$, whose columns are $\chi^{S_0}, \ldots, \chi^{S_{\alpha\omega}}$ (the incidence vectors of the stable sets S_i), and the vectors $\chi^{K_0}, \ldots, \chi^{\alpha\omega+1}$ (the incidence vectors of the cliques K_i), respectively. By Claim 2, we have that $M^{\mathsf{T}}N = J - I$ (where J is the all-ones matrix and I the identity). As $J - I$ is nonsingular, we obtain that that rank$(M^{\mathsf{T}}N) =$ rank$(J - I) = \alpha\omega + 1$. On the other hand, rank$(M^{\mathsf{T}}N) \leq$ rank$N \leq n$. Thus we obtain that $n \geq \alpha\omega + 1$, contradicting our assumption on G. $\square$

3.2 Linear programming bounds

3.2.1 Fractional stable sets and colorings

Let ST(G) denote the polytope in $\mathbb{R}^V$ defined as the convex hull of the characteristic vectors of the stable sets of G:

$$\text{ST}(G) = \text{conv}\{\chi^S : S \subseteq V,\ S \text{ is a stable set in } G\},$$

called the *stable set polytope* of G. Hence, computing $\alpha(G)$ is linear optimization over the stable set polytope:

$$\alpha(G) = \max\{e^{\mathsf{T}}x : x \in \text{ST}(G)\}.$$

We have now defined the stable set polytope by listing explicitly its extreme points. Alternatively, it can also be represented by its hyperplanes representation, i.e., in the form

$$\text{ST}(G) = \{x \in \mathbb{R}^V : Ax \leq b\}$$

for some matrix A and some vector b. As computing the stability number is a hard problem one cannot hope to find the full linear inequality description of the stable set polytope (i.e., the explicit A and b). However some partial information is known: several classes of valid inequalities for the stable set polytope are known. For instance, if C is a clique of G, then the *clique inequality*

$$x(C) = \sum_{i\in C} x_i \leq 1 \tag{3.2}$$

is valid for ST(G): any stable set can contain at most one vertex from the clique C. The clique inequalities define the polytope

$$\text{QST}(G) = \left\{x \in \mathbb{R}^V : x \geq 0,\ x(C) \leq 1\ \forall C \text{ clique of } G\right\} \tag{3.3}$$

and maximizing the linear function $e^{\mathsf{T}}x$ over it gives the parameter

$$\alpha^*(G) = \max\{e^{\mathsf{T}}x : x \in \text{QST}(G)\}, \tag{3.4}$$

known as the *fractional stability number* of G. Clearly, $\text{QST}(G)$ is a relaxation of the stable set polytope:

$$\text{ST}(G) \subseteq \text{QST}(G). \tag{3.5}$$

Analogously, $\chi^*(G)$ denotes the *fractional coloring number* of G, defined by the following linear program:

$$\chi^*(G) = \min \left\{ \sum_{S \text{ stable in } G} \lambda_S : \sum_{S \text{ stable in } G} \lambda_S \chi^S = e,\ \lambda_S \geq 0\ \forall S \text{ stable in } G \right\}. \tag{3.6}$$

If we add the constraint that all λ_S should be integral we obtain the coloring number of G. Thus, $\chi^*(G) \leq \chi(G)$. In fact the fractional stability number of G coincides with the fractional coloring number of its complement: $\alpha^*(G) = \chi^*(\overline{G})$, and it is nested between $\alpha(G)$ and $\chi(\overline{G})$.

Lemma 3.2.1. *For any graph G, we have*

$$\alpha(G) \leq \alpha^*(G) = \chi^*(\overline{G}) \leq \chi(\overline{G}), \tag{3.7}$$

where $\chi^*(\overline{G})$ is the optimum value of the linear program:

$$\min \left\{ \sum_{C \text{ clique of } G} y_C : \sum_{C \text{ clique of } G} y_C \chi^C = e,\ y_C \geq 0\ \forall C \text{ clique of } G \right\}. \tag{3.8}$$

Proof. The inequality $\alpha(G) \leq \alpha^*(G)$ in (3.7) follows from the inclusion (3.5) and the inequality $\chi^*(\overline{G}) \leq \chi(\overline{G})$ was observed above. We now show that $\alpha^*(G) = \chi^*(\overline{G})$. For this, we first observe that in the linear program (3.4) the condition $x \geq 0$ can be removed without changing the optimal value; that is,

$$\alpha^*(G) = \max\{e^{\mathsf{T}} x : x(C) \leq 1\ \forall C \text{ clique of } G\} \tag{3.9}$$

(check it). Now, it suffices to observe that the dual LP of the above linear program (3.9) coincides with the linear program (3.8). □

For instance, for an odd circuit C_{2n+1} ($n \geq 2$), $\alpha^*(C_{2n+1}) = \frac{2n+1}{2}$ (check it) lies strictly between $\alpha(C_{2n+1}) = n$ and $\chi(\overline{C_{2n+1}}) = n + 1$.

When G is a perfect graph, equality holds throughout in relation (3.7). As we see in the next section, there is a natural extension of this result to weighted graphs, which permits to show the equality $\text{ST}(G) = \text{QST}(G)$ when G is a perfect graph. Moreover, it turns out that this geometric property characterizes perfect graphs.

3.2.2 Polyhedral characterization of perfect graphs

For any graph G, the factional stable set polytope is a linear relaxation of the stable set polytope: $\text{ST}(G) \subseteq \text{QST}(G)$. Here we show a geometric characterization of perfect graphs: G is perfect if and only if both polytopes coincide: $\text{ST}(G) = \text{QST}(G)$.

The following operation of *duplicating a node* will be useful. Let $G = (V, E)$ be a graph and let $v \in V$. Add to G a new node, say v', which is adjacent to v and to all neighbours of v in G. In this way we obtain a new graph H, which we say is obtained from G by *duplicating* v. Repeated duplicating is called *replicating*.

Lemma 3.2.2. *Let H arise from G by duplicating a node. If G is perfect then H too is perfect.*

Proof. First we show that $\alpha(H) = \chi(\overline{H})$ if H arises from G by duplicating node v. Indeed, by construction, $\alpha(H) = \alpha(G)$, which is equal to $\chi(\overline{G})$ since G is perfect. Now, if $C_1, \dots, C_t$ are cliques in G that cover V with (say) $v \in C_1$, then $C_1 \cup \{v'\}, \dots, C_t$ are cliques in H covering $V(H)$. This shows that $\chi(\overline{G}) = \chi(\overline{H})$, which implies that $\alpha(H) = \chi(\overline{H})$.

From this we can conclude that, if H arises from G by duplicating a node v, then $\alpha(H') = \chi(\overline{H'})$ for any induced subgraph H' of H, using induction on the number of nodes of G. Indeed, either H' is an induced subgraph of G (if H' does not contain both v and v'), or H' is obtained by duplicating v in an induced subgraph of G; in both cases we have that $\alpha(H') = \chi(\overline{H'})$.

Hence, if H arises by duplicating a node in a perfect graph G, then $\overline{H}$ is perfect which, by Theorem 3.1.3, implies that H is perfect. $\square$

Given node weights $w \in \mathbb{R}^V_+$, we define the following weighted analogues of the (fractional) stability and chromatic numbers:

$$\alpha(G, w) = \max_{x \in \mathrm{ST}(G)} w^{\mathsf{T}} x,$$

$$\alpha^*(G, w) = \max_{x \in \mathrm{QST}(G)} w^{\mathsf{T}} x,$$

$$\chi^*(\overline{G}, w) = \min_y \left\{ \sum_{C \text{ clique of } G} y_C : \sum_{C \text{ clique of } G} y_C \chi^C = w,\ y_C \geq 0\ \forall C \text{ clique of } G \right\},$$

$$\chi(\overline{G}, w) = \min_y \left\{ \sum_{C \text{ clique of } G} y_C : \sum_{C \text{ clique of } G} y_C \chi^C = w,\ y_C \in \mathbb{Z},\ y_C \geq 0\ \forall C \text{ clique of } G \right\}.$$

When w is the all-ones weight function, we find again $\alpha(G)$, $\alpha^*(G)$, $\chi^*(\overline{G})$ and $\chi(\overline{G})$, respectively. The following analogue of (3.7) holds for arbitrary node weights:

$$\alpha(G, w) \leq \alpha^*(G, w) = \chi^*(\overline{G}, w) \leq \chi(\overline{G}, w). \tag{3.10}$$

Lemma 3.2.3. *Let G be a perfect graph and let $w \in \mathbb{Z}^V_{\geq 0}$ be nonnegative integer node weights. Then, $\alpha(\overline{G}, w) = \chi(G, w)$.*

Proof. Let H denote the graph obtained from G by duplicating node i w_i times if $w_i \geq 1$ and deleting node i if $w_i = 0$. Then, by construction, $\alpha(\overline{G}, w) = \omega(H)$, which is equal to $\chi(H)$ since H is perfect (by Lemma 3.2.2). Say, $\tilde{S}_1, \dots, \tilde{S}_t$ are

$t = \chi(H)$ stable sets in H partitioning $V(H)$. Each stable set $\tilde{S}_k$ corresponds to a stable set S_k in G (since $\tilde{S}_k$ contains at most one of the w_i copies of each node i of G). Now, these stable sets $S_1, \dots, S_t$ have the property that each node i of G belongs to exactly w_i of them, which shows that $\chi(G, w) \leq t = \chi(H)$. This implies that $\chi(G, w) \leq \chi(H) = \alpha(\overline{G}, w)$, giving equality $\chi(G, w) = \alpha(\overline{G}, w)$. □

We will also use the following geometric property of down-monotone polytopes. A polytope $P \subseteq \mathbb{R}^n_{\geq 0}$ is said to be *down-monotone* if $x \in P$ and $0 \leq y \leq x$ (coordinate-wise) implies $y \in P$.

Lemma 3.2.4. *Let $P, Q \subseteq \mathbb{R}^n$ be polytopes such that $P \subseteq Q$.*

(i) $P = Q$ if and only if the following equality holds for all weights $w \in \mathbb{R}^n$:

$$\max_{x \in P} w^{\mathsf{T}} x = \max_{x \in Q} w^{\mathsf{T}} x. \tag{3.11}$$

(ii) Assume that $P \subseteq Q \subseteq \mathbb{R}^n_{\geq 0}$ are down-monotone. Then $P = Q$ if and only if (3.11) holds for all nonnegative weights $w \in \mathbb{R}^n_{\geq 0}$.

Moreover, in (i) and (ii) it suffices to show that (3.11) holds for integer weights w.

Proof. (i) The 'only if' part is clear. The 'if part' follows using the 'hyperplane separation' theorem: Assume that $P \subset Q$ and that there exists $z \in Q \setminus P$. Then there exists a hyperplane separating z from P, i.e., there exists a nonzero vector $w \in \mathbb{R}^n$ and a scalar $w_0 \in \mathbb{R}$ such that $w^{\mathsf{T}} z > w_0$ and $w^{\mathsf{T}} x \leq w_0$ for all $x \in P$. These two facts contradict the condition (3.11).

(ii) The 'only if' part is clear. For the 'if part', it suffices to show that the equality (3.11) holds for all weights w if it holds for all nonnegative weights w'. This follows from the following claim (applied to both P and Q).

Claim: Let $P \subseteq \mathbb{R}^n_{\geq 0}$ be a down-monotone polytope, let $w \in \mathbb{R}^n$ and define the nonnegative vector $w' \in \mathbb{R}^n_{\geq 0}$ by $w'_i = \max\{w_i, 0\}$ for $i \in [n]$. Then, $\max_{x \in P} w^{\mathsf{T}} x = \max_{x \in P} (w')^{\mathsf{T}} x$.

Proof of the claim: Suppose $x \in P$ maximizes $w^{\mathsf{T}} x$ over P; we claim that $x_i = 0$ at all positions i for which $w_i < 0$. Indeed, if $x_i > 0$ and $w_i < 0$ then, by setting $y_i = 0$ and $y_j = x_j$ for $j \neq i$, one obtains another point $y \in P$ (since $0 \leq y \leq x$ and P is down-monotone) with $w^{\mathsf{T}} y > w^{\mathsf{T}} x$. Therefore, $w^{\mathsf{T}} x = (w')^{\mathsf{T}} x$ and thus x maximizes w' over P. □

The last part of the lemma follows using a continuity argument (if (3.11) holds for all integer weights w, it holds for all rational weights (by scaling) and thus for all real weights (taking limits)). □

We can now show the following geometric characterization of perfect graphs, due to Chvátal [2].

Theorem 3.2.5. *[2] A graph G is perfect if and only if* $\mathrm{ST}(G) = \mathrm{QST}(G)$.

Proof. First assume that G is perfect, we show that $\mathrm{ST}(G) = \mathrm{QST}(G)$. As $\mathrm{ST}(G)$ and $\mathrm{QST}(G)$ are down-monotone in $\mathbb{R}^V_{\geq 0}$, we can apply Lemma 3.2.4. Hence, it suffices to show that, for any $w \in \mathbb{Z}^V_{\geq 0}$, $\alpha(G, w) = \max_{x \in \mathrm{ST}(G)} w^{\mathsf{T}} x$ is equal to $\alpha^*(G, w) = \max_{x \in \mathrm{QST}(G)} w^{\mathsf{T}} x$, which follows from Lemma 3.2.3 (applied to $\overline{G}$).

Conversely, assume that $\mathrm{ST}(G) = \mathrm{QST}(G)$ and that G is not perfect. Pick a minimal subset $U \subseteq V$ for which the subgraph G' of G induced by U satisfies $\alpha(G') < \chi(\overline{G'})$. Setting $w = \chi^U$, we have that $\alpha(G') = \alpha(G, w)$ which, by assumption, is equal to $\max_{x \in \mathrm{QST}(G)} w^{\mathsf{T}} x = \alpha^*(G, w)$. Consider the dual of the linear program defining $\alpha^*(G, w)$ with an optimal solution $y = (y_C)$. Pick a clique C of G for which $y_C > 0$. Using complementary slackness, we deduce that $x(C) = 1$ for any optimal solution $x \in \mathrm{QST}(G)$ and thus $|C \cap S| = 1$ for any maximum cardinality stable set $S \subseteq U$. Let G'' denote the subgraph of G induced by $U \setminus C$. Then, $\alpha(G'') \leq \alpha(G') - 1 < \chi(\overline{G'}) - 1 \leq \chi(\overline{G''})$, which contradicts the minimality assumption made on U. □

When G is a perfect graph, an explicit linear inequality description is known for its stable set polytope, given by the clique inequalities. However, it is not clear how to use this information in order to give an efficient algorithm for optimizing over the stable set polytope of a perfect graph. As we see later in Section 3.5 there is yet another description of $\mathrm{ST}(G)$ – in terms of semidefinite programming, using the theta body $\mathrm{TH}(G)$ – that will allow to give an efficient algorithm.

3.3 Semidefinite programming bounds

3.3.1 The theta number

Definition 3.3.1. *Given a graph $G = (V, E)$, consider the following semidefinite program*

$$\max_{X \in \mathcal{S}^n} \left\{ \langle J, X \rangle : \mathrm{Tr}(X) = 1,\ X_{ij} = 0 \ \forall \{i, j\} \in E,\ X \succeq 0 \right\}. \tag{3.12}$$

Its optimal value is denoted as $\vartheta(G)$, and called the theta number *of G.*

This parameter was introduced by Lovász [7]. He proved the following simple, but crucial result – called the Sandwich Theorem by Knuth [5] – which shows that $\vartheta(G)$ provides a bound for both the stability number of G and the chromatic number of the complementary graph $\overline{G}$.

Theorem 3.3.2. (Lovász' sandwich theorem) *For any graph G, we have that*

$$\alpha(G) \leq \vartheta(G) \leq \chi(\overline{G}).$$

Proof. Given a stable set S of cardinality $|S| = \alpha(G)$, define the matrix

$$X = \frac{1}{|S|} \chi^S (\chi^S)^{\mathsf{T}} \in \mathcal{S}^n.$$

Then X is feasible for (3.12) with objective value $\langle J, X\rangle = |S|$ (check it). This shows the inequality $\alpha(G) \leq \vartheta(G)$.

Now, consider a matrix X feasible for the program (3.12) and a partition of V into k cliques: $V = C_1 \cup \cdots \cup C_k$. Our goal is now to show that $\langle J, X\rangle \leq k$, which will imply $\vartheta(G) \leq \chi(\overline{G})$. For this, using the relation $e = \sum_{i=1}^k \chi^{C_i}$, observe that

$$Y := \sum_{i=1}^{k} \left(k\chi^{C_i} - e\right)\left(k\chi^{C_i} - e\right)^{\mathsf{T}} = k^2 \sum_{i=1}^{k} \chi^{C_i}(\chi^{C_i})^{\mathsf{T}} - kJ.$$

Moreover,

$$\left\langle X, \sum_{i=1}^{k} \chi^{C_i}(\chi^{C_i})^{\mathsf{T}} \right\rangle = \mathrm{Tr}(X).$$

Indeed the matrix $\sum_i \chi^{C_i}(\chi^{C_i})^{\mathsf{T}}$ has all its diagonal entries equal to 1 and it has zero off-diagonal entries outside the edge set of G, while X has zero off-diagonal entries on the edge set of G. As $X, Y \succeq 0$, we obtain

$$0 \leq \langle X, Y\rangle = k^2\mathrm{Tr}(X) - k\langle J, X\rangle$$

and thus $\langle J, X\rangle \leq k\ \mathrm{Tr}(X) = k$. □

An alternative argument for the inequality $\vartheta(G) \leq \chi(\overline{G})$, showing an even more transparent link to coverings by cliques, will be given in the paragraph after the proof of Lemma 3.4.2.

3.3.2 Computing maximum stable sets in perfect graphs

Assume that G is a graph satisfying $\alpha(G) = \chi(\overline{G})$. Then, as a direct application of Theorem 3.3.2, $\alpha(G) = \chi(\overline{G}) = \vartheta(G)$ can be computed by solving the semidefinite program (3.12), it suffices to solve this semidefinite program with precision $\epsilon < 1/2$ as one can then find $\alpha(G)$ by rounding the optimal value to the nearest integer. In particular, combining with the perfect graph theorem (Theorem 3.1.3):

Theorem 3.3.3. *If G is a perfect graph then $\alpha(G) = \chi(\overline{G}) = \vartheta(G)$ and $\omega(G) = \chi(G) = \vartheta(\overline{G})$.*

Hence one can compute the stability number and the chromatic number in polynomial time for perfect graphs. Moreover, one can also find a maximum stable set and a minimum coloring in polynomial time for perfect graphs. We now indicate how to construct a maximum stable set – we deal with minimum graph colorings in the next section.

Let $G = (V, E)$ be a perfect graph. Order the nodes of G as $v_1, \cdots, v_n$. Then we construct a sequence of induced subgraphs $G_0, G_1, \cdots, G_n$ of G. Hence each G_i is perfect, also after removing a node, so that we can compute in polynomial time the stability number of such graphs. The construction goes as follows: Set $G_0 = G$. For each $i = 1, \cdots, n$ do the following:

1. Compute $\alpha(G_{i-1}\backslash v_i)$.
2. If $\alpha(G_{i-1}\backslash v_i) = \alpha(G)$, then set $G_i = G_{i-1}\backslash v_i$.
3. Otherwise, set $G_i = G_{i-1}$.

By construction, $\alpha(G_i) = \alpha(G)$ for all i. In particular, $\alpha(G_n) = \alpha(G)$. Moreover, the node set of the final graph G_n is a stable set and, therefore, it is a maximum stable set of G. Indeed, if the node set of G_n is not stable then it contains a node v_i for which $\alpha(G_n\backslash v_i) = \alpha(G_n)$. But then, as G_n is an induced subgraph of G_{i-1}, one would have that $\alpha(G_n\backslash v_i) \leq \alpha(G_{i-1}\backslash v_i)$ and thus $\alpha(G_{i-1}\backslash v_i) = \alpha(G)$, so that node v_i would have been removed at Step 2.

Hence, the above algorithm permits to construct a maximum stable set in a perfect graph G in polynomial time – namely by solving $n + 1$ semidefinite programs for computing $\alpha(G)$ and $\alpha(G_{i-1}\backslash v_i)$ for $i = 1, \cdots, n$.

More generally, given integer node weights $w \in \mathbb{Z}^V_{\geq 0}$, the above algorithm can also be used to find a stable set S of maximum weight $w(S)$. For this, construct the new graph G' in the following way: Duplicate each node $i \in V$ w_i times, i.e., replace node $i \in V$ by a set W_i of w_i nodes pairwise non-adjacent, and make two nodes $x \in W_i$ and $y \in W_j$ adjacent if i and j are adjacent in G. By Lemma 3.2.2, the graph G' is perfect. Moreover, $\alpha(G')$ is equal to the maximum weight $w(S)$ of a stable set S in G. From this it follows that, if the weights w_i are bounded by a polynomial in n, then one can compute $\alpha(G, w)$ in polynomial time. (More generally, one can compute $\alpha(G, w)$ in polynomial time, e.g. by optimizing the linear function $w^{\mathsf{T}}x$ over the theta body TH(G), introduced in Section 3.5 below.)

3.3.3 Minimum colorings of perfect graphs

We now describe an algorithm for computing a minimum coloring of a perfect graph G in polynomial time. This will be reduced to several computations of the theta number which we will use for computing the clique number of some induced subgraphs of G.

Let $G = (V, E)$ be a perfect graph. Call a clique of G *maximum* if it has maximum cardinality $\omega(G)$. The crucial observation is that *it suffices to find a stable set S in G which meets all maximum cliques.*

First of all, such a stable set S exists: in a $\omega(G)$-coloring, any color class S must meet all maximum cliques, since $\omega(G \setminus S) = \chi(G \setminus S) = \omega(G) - 1$.

Now, if we have found such a stable set S, then one can recursively color $G\backslash S$ with $\omega(G\backslash S) = \omega(G) - 1$ colors (in polynomial time), and thus one obtains a coloring of G with $\omega(G)$ colors.

The algorithm goes as follows: For $t \geq 1$, we grow a list $\mathcal{L}$ of t maximum cliques $C_1, \cdots, C_t$. Suppose $C_1, \cdots, C_t$ have been found. Then do the following:

1. We find a stable set S meeting each of the cliques $C_1, \cdots, C_t$ (see below).

2. Compute $\omega(G\backslash S)$.

3. If $\omega(G\backslash S) < \omega(G)$ then S meets all maximum cliques and we are done.

4. Otherwise, compute a maximum clique C_{t+1} in $G\backslash S$, which is thus a new maximum clique of G, and we add it to the list $\mathcal{L}$.

The first step can be done as follows: Set $w = \sum_{i=1}^{t} \chi^{C_i} \in \mathbb{Z}^V_{\geq 0}$. As G is perfect, we know that $\alpha(G,w) = \chi(G,w)$, which in turn is equal to t. (Indeed, $\chi(G,w) \leq t$ follows from the definition of w. Moreover, if $y = (y_C)$ is feasible for the program defining $\chi(G,w)$ then, on the one hand, $w^{\mathsf{T}}e = \sum_C y_C|C| \leq \sum_C y_C\omega(G)$ and, on the other hand, $w^{\mathsf{T}}e = t\omega(G)$, thus implying $t \leq \chi(G,w)$.) Now we compute a stable set S having maximum possible weight $w(S)$. Hence, $w(S) = t$ and thus S meets each of the cliques $C_1, \cdots, C_t$.

The above algorithm has polynomial running time, since the number of iterations is bounded by $|V|$. To see this, define the affine space $L_t \subseteq \mathbb{R}^V$ defined by the equations $x(C_1) = 1, \cdots, x(C_t) = 1$ corresponding to the cliques in the current list $\mathcal{L}$. Then, L_t contains strictly L_{t+1}, since $\chi^S \in L_t \setminus L_{t+1}$ for the set S constructed in the first step, and thus the dimension decreases at least by 1 at each iteration.

3.4 Other formulations of the theta number

3.4.1 Dual formulation

We now give several equivalent formulations for the theta number obtained by applying semidefinite programming duality and some further elementary manipulations.

Lemma 3.4.1. *The theta number can be expressed by any of the following programs:*

$$\vartheta(G) = \min_{t\in\mathbb{R}, A\in\mathcal{S}^n} \{t : tI + A - J \succeq 0,\ A_{ij} = 0\ (i = j \ \text{ or } \ \{i,j\} \in \overline{E})\}, \quad (3.13)$$

$$\vartheta(G) = \min_{t\in\mathbb{R}, B\in\mathcal{S}^n} \left\{t : tI - B \succeq 0,\ B_{ij} = 1\ (i = j \text{ or } \ \{i,j\} \in \overline{E})\right\}, \quad (3.14)$$

$$\vartheta(G) = \min_{t\in\mathbb{R}, C\in\mathcal{S}^n} \{t : C - J \succeq 0,\ C_{ii} = t\ (i \in V),\ C_{ij} = 0\ (\{i,j\} \in \overline{E})\}, \quad (3.15)$$

$$\vartheta(G) = \min_{B\in\mathcal{S}^n} \left\{\lambda_{\max}(B) : B_{ij} = 1\ (i = j \text{ or } \ \{i,j\} \in \overline{E})\right\}. \quad (3.16)$$

Proof. First we build the dual of the semidefinite program (3.12), which reads:

$$\min_{t\in\mathbb{R}, y\in\mathbb{R}^E} \left\{ t : tI + \sum_{\{i,j\}\in E} y_{ij}E_{ij} - J \succeq 0 \right\}. \quad (3.17)$$

As both programs (3.12) and (3.17) are strictly feasible, there is no duality gap: the optimal value of (3.17) is equal to $\vartheta(G)$, and the optimal values are attained in both programs – here we have applied the duality theorem (Theorem 2.1.6).

Setting $A = \sum_{\{i,j\}\in E} y_{ij}E_{ij}$, $B = J - A$ and $C = tI + A$ in (3.17), it follows that the program (3.17) is equivalent to each of the programs (3.13), (3.14) and (3.15). Finally the formulation (3.16) follows directly from (3.14) after recalling that $\lambda_{\max}(B)$ is the smallest scalar t for which $tI - B \succeq 0$. □

3.4.2 Two more (lifted) formulations

We give here two more formulations for the theta number. They rely on semidefinite programs involving symmetric matrices of order $1 + n$ which we will index by the set $\{0\} \cup V$, where 0 is an additional index that does not belong to V.

Lemma 3.4.2. *The theta number $\vartheta(G)$ is equal to the optimal value of the following semidefinite program:*

$$\min_{Z\in\mathcal{S}^{n+1}}\{Z_{00} : Z \succeq 0,\ Z_{0i} = Z_{ii} = 1\ (i \in V),\ Z_{ij} = 0\ (\{i,j\} \in \overline{E})\}. \tag{3.18}$$

Proof. We show that the two semidefinite programs in (3.13) and (3.18) are equivalent. For this, observe that

$$tI + A - J \succeq 0 \Longleftrightarrow Z := \begin{pmatrix} t & e^{\mathsf{T}} \\ e & I + \frac{1}{t}A \end{pmatrix} \succeq 0,$$

which follows by taking the Schur complement of the upper left corner t in the block matrix Z. Hence, if (t, A) is feasible for (3.13), then Z is feasible for (3.18) with same objective value: $Z_{00} = t$. The construction can be reversed: if Z is feasible for (3.18), then one can construct (t, A) feasible for (3.13) with $t = Z_{00}$. Hence both programs are equivalent. □

From the formulation (3.18), the link of the theta number to the (fractional) chromatic number is even more transparent.

Lemma 3.4.3. *For any graph G, we have that $\vartheta(G) \le \chi^*(\overline{G})$.*

Proof. Let $y = (y_C)$ be feasible for the linear program (3.8) defining $\chi^*(\overline{G})$. For each clique C define the vector $z_C = (1\ \chi^C) \in \mathbb{R}^{1+n}$, obtained by appending an entry equal to 1 to the characteristic vector of C. Define the matrix $Z = \sum_{C \text{ clique of } G} y_C z_C z_C^{\mathsf{T}}$. One can verify that Z is feasible for the program (3.18) with objective value $Z_{00} = \sum_C y_C$ (check it). This shows $\vartheta(G) \le \chi^*(\overline{G})$. □

Applying duality to the semidefinite program (3.18), we obtain[1] the following formulation for $\vartheta(G)$.

[1] Of course there is more than one road leading to Rome: one can also show directly the equivalence of the two programs (3.12) and (3.19).

Lemma 3.4.4. *The theta number $\vartheta(G)$ is equal to the optimal value of the following semidefinite program:*

$$\max_{Y \in \mathcal{S}^{n+1}} \left\{ \sum_{i \in V} Y_{ii} : Y \succeq 0,\ Y_{00} = 1,\ Y_{0i} = Y_{ii}\ (i \in V),\ Y_{ij} = 0\ (\{i,j\} \in E) \right\}. \tag{3.19}$$

Proof. First we write the program (3.18) in standard form, using the elementary matrices E_{ij} (with entries 1 at positions (i,j) and (j,i) and 0 elsewhere):

$$\inf\{\langle E_{00}, Z\rangle : \langle E_{ii}, Z\rangle = 1,\ \langle E_{0i}, Z\rangle = 2\ (i \in V),\ \langle E_{ij}, Z\rangle = 0\ (\{i,j\} \in \overline{E}),\ Z \succeq 0\}.$$

Next we write the dual of this sdp:

$$\sup \left\{ \sum_{i \in V} y_i + 2z_i : Y = E_{00} - \sum_{i \in V} y_i E_{ii} + z_i E_{0i} + \sum_{\{i,j\} \in \overline{E}} u_{ij} E_{ij} \succeq 0 \right\}.$$

Observe now that the matrix $Y \in \mathcal{S}^{n+1}$ occurring in this program can be equivalently characterized by the conditions: $Y_{00} = 1$, $Y_{ij} = 0$ if $\{i,j\} \in E$ and $Y \succeq 0$. Moreover the objective function reads: $\sum_{i \in V} y_i + 2z_i = -\left(\sum_{i \in V} Y_{ii} + 2Y_{0i}\right)$. Therefore the dual can be equivalently reformulated as

$$\max \left\{ -\left(\sum_{i \in V} Y_{ii} + 2Y_{0i} \right) : Y \succeq 0,\ Y_{00} = 1,\ Y_{ij} = 0\ (\{i,j\} \in E) \right\}. \tag{3.20}$$

As (3.18) is strictly feasible (check it) there is no duality gap, the optimal value of (3.20) is attained and it is equal to $\vartheta(G)$.

Let Y be an optimal solution of (3.20). We claim that $Y_{0i} + Y_{ii} = 0$ for all $i \in V$. Indeed, assume that $Y_{0i} + Y_{ii} \neq 0$ for some $i \in V$. Then, $Y_{ii} \neq 0$. Let us multiply the i-th column and the i-th row of the matrix Y by the scalar $-\frac{Y_{0i}}{Y_{ii}}$. In this way we obtain a new matrix Y' which is still feasible for (3.20), but now with a better objective value: Indeed, by construction, $Y'_{ii} = Y_{ii}\left(-\frac{Y_{0i}}{Y_{ii}}\right)^2 = \frac{Y_{0i}^2}{Y_{ii}}$ and $Y'_{0i} = Y_{0i}\left(-\frac{Y_{0i}}{Y_{ii}}\right) = -\frac{Y_{0i}^2}{Y_{ii}} - Y'_{ii}$. Moreover, the i-th term in the new objective value is

$$-(Y'_{ii} + 2Y'_{0i}) = \frac{Y_{0i}^2}{Y_{ii}} > -(Y_{ii} + 2Y_{0i}).$$

This contradicts optimality of Y and thus we have shown that $Y_{0i} = -Y_{ii}$ for all $i \in V$. Therefore, we can add w.l.o.g. the condition $Y_{0i} = -Y_{ii}$ $(i \in V)$ to (3.20), so that its objective function can be replaced by $\sum_{i \in V} Y_{ii}$.

Finally, to get the program (3.19), it suffices to observe that one can change the signs on the first row and column of Y (indexed by the index 0). In this way we obtain a matrix $\tilde{Y}$ such that $\tilde{Y}_{0i} = -Y_{0i}$ for all i and $\tilde{Y}_{ij} = Y_{ij}$ at all other positions. Thus $\tilde{Y}$ now satisfies the conditions $Y_{ii} = Y_{0i}$ for $i \in V$ and it is an optimal solution of (3.19). □

3.5 The theta body $\mathrm{TH}(G)$

It is convenient to introduce the following set of matrices $X \in \mathcal{S}^{n+1}$, where columns and rows are indexed by the set $\{0\} \cup V$:

$$\mathcal{M}_G = \{Y \in \mathcal{S}^{n+1} : Y_{00} = 1,\ Y_{0i} = Y_{ii}\ (i \in V),\ Y_{ij} = 0\ (\{i,j\} \in E),\ Y \succeq 0\}, \tag{3.21}$$

which is thus the feasible region of the semidefinite program (3.19). Now let $\mathrm{TH}(G)$ denote the convex set obtained by projecting the set $\mathcal{M}_G$ onto the subspace $\mathbb{R}^V$ of the diagonal entries:

$$\mathrm{TH}(G) = \{x \in \mathbb{R}^V : \exists Y \in \mathcal{M}_G \text{ such that } x_i = Y_{ii}\ \forall i \in V\}, \tag{3.22}$$

called the *theta body* of G. It turns out that $\mathrm{TH}(G)$ is nested between $\mathrm{ST}(G)$ and $\mathrm{QST}(G)$.

Lemma 3.5.1. *For any graph G, we have that* $\mathrm{ST}(G) \subseteq \mathrm{TH}(G) \subseteq \mathrm{QST}(G)$.

Proof. The inclusion $\mathrm{ST}(G) \subseteq \mathrm{TH}(G)$ follows from the fact that the characteristic vector of any stable set S lies in $\mathrm{TH}(G)$. To see this, define the vector $y = (1\ \chi^S) \in \mathbb{R}^{n+1}$ obtained by adding an entry equal to 1 to the characteristic vector of S, and define the matrix $Y = yy^{\mathsf{T}} \in \mathcal{S}^{n+1}$. Then $Y \in \mathcal{M}_G$ and $\chi^S = (Y_{ii})_{i \in V}$, which shows that $\chi^S \in \mathrm{TH}(G)$.

We now show the inclusion $\mathrm{TH}(G) \subseteq \mathrm{QST}(G)$. For this take $x \in \mathrm{TH}(G)$ and let $Y \in \mathcal{M}_G$ such that $x = (Y_{ii})_{i \in V}$. Then $x \geq 0$ (as the diagonal entries of a psd matrix are nonnegative). Moreover, for any clique C of G, we have that $x(C) \leq 1$ (cf. Exercise 1.1). □

In view of Lemma 3.4.4, maximizing the all-ones objective function over $\mathrm{TH}(G)$ gives the theta number:

$$\vartheta(G) = \max_{x \in \mathbb{R}^V} \{e^{\mathsf{T}} x : x \in \mathrm{TH}(G)\}.$$

As maximizing $e^{\mathsf{T}} x$ over $\mathrm{QST}(G)$ gives the LP bound $\alpha^*(G)$, Lemma 3.5.1 implies directly that the SDP bound $\vartheta(G)$ dominates the LP bound $\alpha^*(G)$:

Corollary 3.5.2. *For any graph G, we have that* $\alpha(G) \leq \vartheta(G) \leq \alpha^*(G)$.

Combining the inclusion from Lemma 3.5.1 with Theorem 3.2.5, we deduce that $\mathrm{TH}(G) = \mathrm{ST}(G) = \mathrm{QST}(G)$ for perfect graphs. It turns out that these equalities characterize perfect graphs.

Theorem 3.5.3. *(see [2]) For any graph G the following assertions are equivalent.*

1. *G is perfect.*
2. $\mathrm{TH}(G) = \mathrm{ST}(G)$
3. $\mathrm{TH}(G) = \mathrm{QST}(G)$.

4. TH(G) *is a polytope.*

We also mention the following beautiful relationship between the theta bodies of a graph G and of its complementary graph $\overline{G}$:

Theorem 3.5.4. *For any graph G,*

$$\text{TH}(G) = \{x \in \mathbb{R}^V_{\geq 0} : x^T z \leq 1 \ \forall z \in \text{TH}(\overline{G})\}.$$

In other words, we know an explicit linear inequality description of TH(G); moreover, the normal vectors to the supporting hyperplanes of TH(G) are precisely the elements of TH$(\overline{G})$. One inclusion is easy:

Lemma 3.5.5. *If $x \in$ TH(G) and $z \in$ TH$(\overline{G})$ then $x^\mathsf{T} z \leq 1$.*

Proof. Let $Y \in \mathcal{M}_G$ and $Z \in \mathcal{M}_{\overline{G}}$ such that $x = (Y_{ii})$ and $z = (Z_{ii})$. Let Z' be obtained from Z by changing signs in its first row and column (indexed by 0). Then $\langle Y, Z' \rangle \geq 0$ as $Y, Z' \succeq 0$. Moreover, $\langle Y, Z' \rangle = 1 - x^\mathsf{T} z$ (check it), thus giving $x^\mathsf{T} z \leq 1$. □

3.6 The theta number for vertex-transitive graphs

First we mention an inequality relating the theta numbers of a graph and its complement. (You will show it in Exercise 6.1.)

Proposition 3.6.1. *For any graph $G = (V, E)$, we have that $\vartheta(G)\vartheta(\overline{G}) \geq |V|$.*

We now show that equality $\vartheta(G)\vartheta(\overline{G}) = |V|$ holds for certain symmetric graphs, namely for vertex-transitive graphs. In order to show this, one exploits in a crucial manner the symmetry of G, which permits to show that the semidefinite program defining the theta number has an optimal solution with a special (symmetric) structure. We need to introduce some definitions.

Let $G = (V, E)$ be a graph. A permutation σ of the node set V is called an *automorphism* of G if it preserves edges, i.e., $\{i, j\} \in E$ implies $\{\sigma(i), \sigma(j)\} \in E$. Then the set Aut$(G)$ of automorphisms of G is a group. The graph G is said to be *vertex-transitive* if for any two nodes $i, j \in V$ there exists an automorphism $\sigma \in$ Aut(G) mapping i to j: $\sigma(i) = j$.

The group of permutations of V acts on symmetric matrices X indexed by V. Namely, if σ is a permutation of V and P_σ is the corresponding permutation matrix (with $P_\sigma(i, j) = P_{\sigma(i),\sigma(j)}$ for all $i, j \in V$), then one can build the new symmetric matrix

$$\sigma(X) := P_\sigma X P_\sigma^\mathsf{T} = (X_{\sigma(i),\sigma(j)})_{i,j \in V}.$$

If σ is an automorphism of G, then it preserves the feasible region of the semidefinite program (3.12) defining the theta number $\vartheta(G)$. This is an easy, but very useful fact.

Lemma 3.6.2. *If X is feasible for the program (3.12) and σ is an automorphism of G, then $\sigma(X)$ is again feasible for (3.12), moreover with the same objective value as X.*

Proof. Directly from the fact that $\langle J, \sigma(X)\rangle = \langle J, X\rangle$, $\text{Tr}(\sigma(X)) = \text{Tr}(X)$ and $\sigma(X)_{ij} = X_{\sigma(i)\sigma(j)} = 0$ if $\{i,j\} \in E$ (since σ is an automorphism of G). □

Lemma 3.6.3. *The program (3.12) has an optimal solution X^* which is invariant under action of the automorphism group of G, i.e., satisfies $\sigma(X^*) = X^*$ for all $\sigma \in \text{Aut}(G)$.*

Proof. Let X be an optimal solution of (3.12). By Lemma 3.6.2, $\sigma(X)$ is again an optimal solution for each $\sigma \in \text{Aut}(G)$. Define the matrix

$$X^* = \frac{1}{|\text{Aut}(G)|} \sum_{\sigma \in \text{Aut}(G)} \sigma(X),$$

obtained by averaging over all matrices $\sigma(X)$ for $\sigma \in \text{Aut}(G)$. As the set of optimal solutions of (3.12) is convex, X^* is still an optimal solution of (3.12). Moreover, by construction, X^* is invariant under action of $\text{Aut}(G)$. □

Corollary 3.6.4. *If G is a vertex-transitive graph then the program (3.12) has an optimal solution X^* satisfying $X^*_{ii} = 1/n$ for all $i \in V$ and $X^* e = \frac{\vartheta(G)}{n} e$.*

Proof. By Lemma 3.6.3, there is an optimal solution X^* which is invariant under action of $\text{Aut}(G)$. As G is vertex-transitive, all diagonal entries of X^* are equal: Indeed, let $i, j \in V$ and $\sigma \in \text{Aut}(G)$ such that $\sigma(i) = j$. Then, $X_{jj} = X_{\sigma(i)\sigma(i)} = X_{ii}$. As $\text{Tr}(X^*) = 1$ we must have $X^*_{ii} = 1/n$ for all i. Analogously, the invariance of X^* implies that $\sum_{k \in V} X^*_{ik} = \sum_{k \in V} X^*_{jk}$ for all i, j, i.e., $X^* e = \lambda e$ for some scalar λ. Combining with the condition $\langle J, X^*\rangle = \vartheta(G)$ we obtain that $\lambda = \frac{\vartheta(G)}{n}$. □

Proposition 3.6.5. *If G is a vertex-transitive graph, then $\vartheta(G)\vartheta(\overline{G}) = |V|$.*

Proof. By Corollary 3.6.4, there is an optimal solution X^* of the program (3.12) defining $\vartheta(G)$ which satisfies $X^*_{ii} = 1/n$ for $i \in V$ and $X^* e = \frac{\vartheta(G)}{n} e$. Then $\frac{n^2}{\vartheta(G)} X^* - J \succeq 0$ (check it). Hence, $t = \frac{n}{\vartheta(G)}$ and $C = \frac{n^2}{\vartheta(G)} X^*$ define a feasible solution of the program (3.15) defining $\vartheta(\overline{G})$, which implies $\vartheta(\overline{G}) \leq n/\vartheta(G)$. Combining with Proposition 3.6.1 we get the equality $\vartheta(G)\vartheta(\overline{G}) = |V|$. □

For instance, the cycle C_n is vertex-transitive, so that

$$\vartheta(C_n)\vartheta(\overline{C_n}) = n. \tag{3.23}$$

In particular, as C_5 is isomorphic to $\overline{C_5}$, we deduce that

$$\vartheta(C_5) = \sqrt{5}. \tag{3.24}$$

For n even, C_n is bipartite (and thus perfect), so that $\vartheta(C_n) = \alpha(C_n) = \frac{n}{2}$ and $\vartheta(\overline{C_n}) = \omega(C_n) = 2$. For n odd, one can compute $\vartheta(C_n)$ using the above symmetry reduction:

Proposition 3.6.6. *For any odd $n \geq 3$,*

$$\vartheta(C_n) = \frac{n\cos(\pi/n)}{1+\cos(\pi/n)} \quad \textit{and} \quad \vartheta(\overline{C_n}) = \frac{1+\cos(\pi/n)}{\cos(\pi/n)}.$$

Proof. As $\vartheta(C_n)\vartheta(\overline{C_n}) = n$, it suffices to compute $\vartheta(C_n)$. We use the formulation (3.16). As C_n is vertex-transitive, there is an optimal solution B whose entries are all equal to 1, except $B_{ij} = 1 + x$ for some scalar x whenever $|i - j| = 1$ (modulo n). In other words, $B = J + xA_{C_n}$, where A_{C_n} is the adjacency matrix of the cycle C_n. Thus $\vartheta(C_n)$ is equal to the minimum value of $\lambda_{\max}(B)$ for all possible x. The eigenvalues of A_{C_n} are known: They are $\omega^k + \omega^{-k}$ (for $k = 0, 1, \cdots, n-1$), where $\omega = e^{\frac{2i\pi}{n}}$ is an n-th root of unity. Hence the eigenvalues of B are

$$n + 2x \;\text{ and } x(\omega^k + \omega^{-k}) \;\text{ for } k = 1, \cdots, n-1. \tag{3.25}$$

We minimize the maximum of the values in (3.25) when choosing x such that

$$n + 2x = -2x\cos(\pi/n)$$

(check it). This gives $\vartheta(C_n) = \lambda_{\max}(B) = -2x\cos(\pi/n) = \frac{n\cos(\pi/n)}{1+\cos(\pi/n)}$. □

3.7 Bounding the Shannon capacity

The theta number was introduced by Lovász [7] in connection with the problem of computing the Shannon capacity of a graph, a problem in coding theory considered by Shannon. We need some definitions.

Definition 3.7.1. (Strong product) *Let $G = (V, E)$ and $H = (W, F)$ be two graphs. Their* strong product *is the graph denoted as $G \cdot H$ with node set $V \times W$ and with edges the pairs of distinct nodes $\{(i, r), (j, s)\} \in V \times W$ with ($i = j$ or $\{i, j\} \in E$) and ($r = s$ or $\{r, s\} \in F$).*

If $S \subseteq V$ is stable in G and $T \subseteq W$ is stable in H then $S \times T$ is stable in $G \cdot H$. Hence, $\alpha(G \cdot H) \geq \alpha(G)\alpha(H)$. Let G^k denote the strong product of k copies of G, we have that

$$\alpha(G^k) \geq (\alpha(G))^k.$$

Based on this, one can verify that

$$\Theta(G) := \sup_{k \geq 1} \sqrt[k]{\alpha(G^k)} = \lim_{k \to \infty} \sqrt[k]{\alpha(G^k)}. \tag{3.26}$$

The parameter $\Theta(G)$ was introduced by Shannon in 1956, it is called the *Shannon capacity* of the graph G. The motivation is as follows. Suppose V is a finite alphabet, where some pairs of letters could be confused when they are transmitted over some transmission channel. These pairs of confusable letters can be seen as the edge set E of a graph $G = (V, E)$. Then the stability number of

G is the largest number of one-letter messages that can be sent without danger of confusion. Words of length k correspond to k-tuples in V^k. Two words $(i_1, \cdots, i_k)$ and $(j_1, \cdots, j_k)$ can be confused if at every position $h \in [k]$ the two letters i_h and j_h are equal or can be confused, which corresponds to having an edge in the strong product G^k. Hence the largest number of words of length k that can be sent without danger of confusion is equal to the stability number of G^k and the Shannon capacity of G represents the rate of correct transmission of the graph.

For instance, for the 5-cycle C_5, $\alpha(C_5) = 2$, but $\alpha((C_5)^2) \geq 5$. Indeed, if $1, 2, 3, 4, 5$ are the nodes of C_5 (in this cyclic order), then the five 2-letter words $(1, 1)$, $(2, 3)$, $(3, 5)$, $(4, 2)$, $(5, 4)$ form a stable set in G^2. This implies that $\Theta(C_5) \geq \sqrt{5}$.

Determining the exact Shannon capacity of a graph is a very difficult problem in general, even for small graphs. For instance, the exact value of the Shannon capacity of C_5 was not known until Lovász [7] showed how to use the theta number in order to upper bound the Shannon capacity: Lovász showed that $\Theta(G) \leq \vartheta(G)$ and $\vartheta(C_5) = \sqrt{5}$, which implies that $\Theta(C_5) = \sqrt{5}$. For instance, although the exact value of the theta number of C_{2n+1} is known (cf. Proposition 3.6.6), the exact value of the Shannon capacity of C_{2n+1} is not known, already for C_7.

Theorem 3.7.2. *For any graph G, we have that* $\Theta(G) \leq \vartheta(G)$.

The proof is based on the multiplicative property of the theta number from Lemma 3.7.3 – which you will prove in Exercise 6.2 – combined with the fact that the theta number upper bounds the stability number: For any integer k, $\alpha(G^k) \leq \vartheta(G^k) = (\vartheta(G))^k$ implies $\sqrt[k]{\alpha(G^k)} \leq \vartheta(G)$ and thus $\Theta(G) \leq \vartheta(G)$.

Lemma 3.7.3. *The theta number of the strong product of two graphs G and H satisfies* $\vartheta(G \cdot H) = \vartheta(G)\vartheta(H)$.

As an application one can compute the Shannon capacity of any graph G which is vertex-transitive and self-complementary (e.g., like C_5).

Theorem 3.7.4. *If $G = (V, E)$ is a vertex-transitive graph, then $\Theta(G \cdot \overline{G}) = |V|$. If, moreover, G is self-complementary, then $\Theta(G) = \sqrt{|V|}$.*

Proof. We have $\Theta(G \cdot \overline{G}) \geq \alpha(G \cdot \overline{G}) \geq |V|$, since the set of diagonal pairs $\{(i, i) : i \in V\}$ is stable in $G \cdot \overline{G}$. The reverse inequality follows from Lemma 3.7.3 combined with Proposition 3.6.5: $\Theta(G \cdot \overline{G}) \leq \vartheta(G \cdot \overline{G}) = \vartheta(G)\vartheta(\overline{G}) = |V|$. If G is isomorphic to $\overline{G}$ then $\Theta(G \cdot \overline{G}) = \Theta(G^2) = (\Theta(G))^2$ (check the rightmost equality). This gives $\Theta(G) = \sqrt{|V|}$. □

3.8 Exercises

3.1. Show: $\vartheta(G)\vartheta(\overline{G}) \geq n$ for any graph G on n nodes.

Show: $\vartheta(C_5) \geq \sqrt{5}$.

Hint: Let C be a feasible solution for the program (3.15) defining $\vartheta(G)$, and let C' be a feasible solution of the analogous program defining $\vartheta(\overline{G})$. Use the fact that $\langle C - J, C' - J\rangle \geq 0$, $\langle C - J, J\rangle \geq 0$, $\langle C' - J, J\rangle \geq 0$ (why is this true?), and compute $\langle C, C'\rangle$.

3.2 The goal is to show the result of Lemma 3.7.3 about the theta number of the strong product of two graphs $G = (V, E)$ and $H = (W, F)$:

$$\vartheta(G \cdot H) = \vartheta(G)\vartheta(H).$$

(a) Show that $\vartheta(G \cdot H) \geq \vartheta(G)\vartheta(H)$.

(b) Show that $\vartheta(G \cdot H) \leq \vartheta(G)\vartheta(H)$.

Hint: Use the primal formulation (3.12) for (a), and the dual formulation (Lemma 3.4.1) for (b), and think of using Kronecker products of matrices in order to build feasible solutions.

3.3 Let $G = (V = [n], E)$ be a graph. A set of vectors $u_1, \ldots, u_n \in \mathbb{R}^n$ is said to be an *orthonormal representation* of G if they satisfy:

$$\|u_i\| = 1 \ \forall i \in V, \ u_i^{\mathsf{T}} u_j = 0 \ \forall \{i, j\} \in \overline{E}.$$

Consider the graph parameter

$$\vartheta_1(G) = \min_{c, u_i} \max_{i \in V} \frac{1}{(c^{\mathsf{T}} u_i)^2},$$

where the minimum is taken over all unit vectors c and all orthonormal representations $u_1, \cdots, u_n$ of G.

(a) Show: $\vartheta(G) \leq \vartheta_1(G)$.
Hint: Consider unit vectors $c, u_1, \ldots, u_n$ forming an orthonormal representation of G. Set $t = \max_i \frac{1}{(c^{\mathsf{T}} u_i)^2}$ and define the symmetric matrix A with entries $A_{ij} = \frac{u_i^{\mathsf{T}} u_j}{(c^{\mathsf{T}} u_i)(c^{\mathsf{T}} u_j)}$ for $\{i, j\} \in E$ and $A_{ij} = 0$ otherwise. It might also be useful to consider the matrix M, defined as the Gram matrix of the vectors $c - \frac{u_i}{c^{\mathsf{T}} u_i}$ for $i \in V$. Show that (t, A) is feasible for the formulation (3.13) for $\vartheta(G)$.

(b) Show: $\vartheta_1(G) \leq \vartheta(G)$.
Hint: Consider an optimum solution $(t = \vartheta(G), B)$ to the program (3.14) defining $\vartheta(G)$. Say $tI - B$ is the Gram matrix of the vectors $x_1, \ldots, x_n$. Show that there exists a unit vector c orthogonal to $x_1, \ldots, x_n$. Show that the vectors $u_i = \frac{c + x_i}{\sqrt{t}}$ $(i \in V)$ form an orthonormal representation of G.

(c) Show: $\vartheta(C_5) \leq \sqrt{5}$.

Hint: Use the formulation via $\vartheta_1(G)$ and the following construction (known as *Lovász' umbrella construction*).

Set $c = (0, 0, 1) \in \mathbb{R}^3$ and, for $k = 1, 2, 3, 4, 5$, define the vector $u_k = (s\cos(2k\pi/5), s\sin(2k\pi/5), t) \in \mathbb{R}^3$, where s and t are scalars to be determined. Show that one can choose s and t in such a way that $u_1, \ldots, u_5$ form an orthonormal representation of C_5.

Recall: $\cos(2\pi/5) = \frac{\sqrt{5}-1}{4}$.

BIBLIOGRAPHY

[1] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas. The strong perfect graph theorem, *Annals of Mathematics* **164 (1):** 51229, 2006.

[2] V. Chvátal. On certain polytopes associated with graphs. *Journal of Combinatorial Theory, Series B* **18**:138–154, 1975.

[3] G.S. Gasparian. Minimal imperfect graphs: a simple approach. *Combinatorica*, **16**:209–212, 1996.

[4] M. Grötschel, L. Lovász, A. Schrijver. *Geometric Algorithms in Combinatorial Optimization*, Springer, 1988. `http://www.zib.de/groetschel/pubnew/paper/groetschellovaszschrijver1988.pdf`

[5] D.E. Knuth. The Sandwich Theorem. *The Electronic Journal of Combinatorics* **1, A1**, 1994. `http://www.combinatorics.org/ojs/index.php/eljc/article/view/v1i1a1`

[6] L. Lovász. A characterization of perfect graphs. *Journal of Combinatorial Theory, Series B* **13**:95–98, 1972.

[7] L. Lovász. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory* **IT-25**:1–7, 1979.

CHAPTER 4

APPROXIMATING THE MAX CUT PROBLEM

4.1 Introduction

4.1.1 The MAX CUT problem

The maximum cut problem (MAX CUT) is the following problem in combinatorial optimization. Let $G = (V, E)$ be a graph and let $w = (w_{ij}) \in \mathbb{R}^E_+$ be nonnegative weights assigned to the edges. Given a subset $S \subseteq V$, the *cut* $\delta_G(S)$ consists of the edges $\{u, v\} \in E$ having exactly one endnode in S, i.e., with $|\{i, j\} \cap S| = 1$. In other words, $\delta_G(S)$ consists of the edges that are *cut* by the partition $(S, \overline{S} = V \setminus S)$ of V. The cut $\delta_G(S)$ is called *trivial* if $S = \emptyset$ or V (in which case it is empty). Then the weight of the cut $\delta_G(S)$ is $w(\delta_G(S)) = \sum_{\{i,j\} \in \delta_G(S)} w_{ij}$ and the MAX CUT problem asks for a cut of maximum weight, i.e., compute

$$\text{mc}(G, w) = \max_{S \subseteq V} w(\delta_G(S)).$$

It is sometimes convenient to extend the weight function $w \in \mathbb{R}^E$ to all pairs of nodes of V, by setting $w_{ij} = 0$ if $\{i, j\}$ is not an edge of G. Given disjoint subsets $S, T \subseteq V$, it is also convenient to use the following notation:

$$w(S, T) = \sum_{i \in S, j \in T} w_{ij}.$$

Thus,

$$w(S, \overline{S}) = w(\delta_G(S)) \text{ for all } S \subseteq V.$$

To state its complexity, we formulate MAX CUT as a decision problem:

MAX CUT: *Given a graph $G = (V, E)$, edge weights $w \in \mathbb{Z}_+^E$ and an integer $k \in \mathbb{N}$, decide whether there exists a cut of weight at least k.*

It is well known that MAX CUT is an NP-complete problem. In fact, MAX CUT is one of Karp's 21 NP-complete problems. So unless the complexity classes P and NP coincide there is no efficient polynomial-time algorithm which solves MAX CUT exactly. We give here a reduction of MAX CUT from the PARTITION problem, defined below, which is one the first six basic NP-complete problems in Garey and Johnson [3]:

PARTITION: *Given natural numbers $a_1, \dots, a_n \in \mathbb{N}$, decide whether there exists a subset $S \subseteq [n]$ such that $\sum_{i \in S} a_i = \sum_{i \notin S} a_i$.*

Theorem 4.1.1. *The MAX CUT problem is NP-complete.*

Proof. It is clear that MAX CUT to the class NP. We now show a reduction from PARTITION. Let $a_1, \dots, a_n \in \mathbb{N}$ be given. Construct the following weights $w_{ij} = a_i a_j$ for the edges of the complete graph K_n. Set $\sigma = \sum_{i=1}^n a_i$ and $k = \sigma^2/4$. For any subset $S \subseteq [n]$, set $a(S) = \sum_{i \in S} a_i$. Then, we have

$$w(S, \overline{S}) = \sum_{i \in S, j \in \overline{S}} w_{ij} = \sum_{i \in S, j \in \overline{S}} a_i a_j = (\sum_{i \in S} a_i)(\sum_{j \in \overline{S}} a_j) = a(S)(\sigma - a(S)) \leq \sigma^2/4,$$

with equality if and only if $a(S) = \sigma/2$ or, equivalently, $a(S) = a(\overline{S})$. From this it follows that there is a cut of weight at least k if and only if the sequence $a_1, \dots, a_n$ can be partitioned. This concludes the proof. □

This hardness result for MAX CUT is in sharp contrast to the situation of the MIN CUT problem, which asks for a *nontrivial* cut of minimum weight, i.e., to compute

$$\min_{S \subseteq V : S \neq \emptyset, V} w(S, \overline{S}).$$

(For MIN CUT the weights of edges are usually called *capacities* and they also assumed to be nonnegative). It is well known that the MIN CUT problem can be solved in polynomial time (together with its dual MAX FLOW problem), using the Ford-Fulkerson algorithm. Specifically, the Ford-Fulkerson algorithm permits to find in polynomial time a minimum cut $(S, \overline{S})$ separating a given source s and a given sink t, i.e., with $s \in S$ and $t \in \overline{S}$. Thus a minimum weight nontrivial cut can be obtained by applying this algorithm $|V|$ times, fixing any $s \in V$ and letting t vary over all nodes of $V \setminus \{s\}$. Details can be found in Chapter 4 of the Lecture Notes [7].

Even stronger, Håstad in 2001 showed that it is NP-hard to approximate MAX CUT within a factor of $\frac{16}{17} \sim 0.941$.

On the positive side, one can compute a 0.878-approximation of MAX CUT in polynomial time, using semidefinite programming. This algorithm, due to

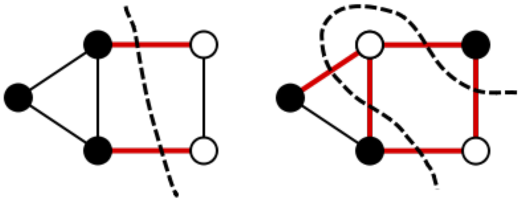


Figure 4.1: Minimum and maximum weight cuts

Goemans and Williamson [4], is one of the most influential approximation algorithms which are based on semidefinite programming. We will explain this result in detail in Section 4.2.1.

Before doing that we recall some results for MAX CUT based on using linear programming.

4.1.2 Linear programming relaxation

In order to define linear programming bounds for MAX CUT, one needs to find some linear inequalities that are satisfied by all cuts of G, i.e., some valid inequalities for the cut polytope of G. Large classes of such inequalities are known (cf. e.g. [2] for an overview and references).

We now present some simple but important valid inequalities for the cut polytope of the complete graph K_n, which is denoted as CUT$_n$, and defined as the convex hull of the incidence vectors of the cuts of K_n:

$$\text{CUT}_n = \text{conv}\{\chi^{\delta_{K_n}(S)} : S \subseteq [n]\}.$$

For instance, for $n = 2$, CUT$_n = [0,1]$ and, for $n = 3$, CUT$_3$ is a simplex in $\mathbb{R}^3$ (indexed by the edges of K_3 ordered as $\{1,2\},\{1,3\},\{2,3\}$) with as vertices the incidence vectors of the four cuts $(S,\overline{S})$ of K_3: $(0,\ 0,\ 0)$, $(1,\ 1,\ 0)$, $(1,\ 0,\ 1)$, and $(0\ 1\ 1)$ (for $S = \emptyset, \{1\}, \{2\}$ and $\{3\}$, respectively).

As a first easy observation it is important to realize that in order to compute the maximum cut mc(G,w) in a weighted graph G on n nodes, one can as well deal with the complete graph K_n. Indeed, any cut $\delta_G(S)$ of G can be obtained from the corresponding cut $\delta_{K_n}(S)$ of K_n, simply by ignoring the pairs that are not edges of G, in other words, by projecting onto the edge set of G. Hence one can reformulate any maximum cut problem as a linear optimization problem over the cut polytope of K_n:

$$\text{mc}(G,w) = \max_{x \in \text{CUT}_n} \sum_{\{i,j\} \in E} w_{ij}x_{ij};$$

the graph G is taken into account by the objective function of this LP.

The following *triangle inequalities* are valid for the cut polytope CUT$_n$:

$$x_{ij} - x_{ik} - x_{jk} \leq 0,\ x_{ij} + x_{jk} + x_{jk} \leq 2, \tag{4.1}$$

for all distinct $i, j, k \in [n]$. This is easy to see, just verify that these inequalities hold when x is equal to the incidence vector of a cut. The triangle inequalities (4.1) imply the following bounds (check it):

$$0 \leq x_{ij} \leq 1 \tag{4.2}$$

on the variables. Let MET$_n$ denote the polytope in $\mathbb{R}^{E(K_n)}$ defined by the triangle inequalities (4.1). Thus, MET$_n$ is a linear relaxation of CUT$_n$, tighter than the trivial relaxation by the unit hypercube:

$$\text{CUT}_n \subseteq \text{MET}_n \subseteq [0,1]^{E(K_n)}.$$

It is known that equality holds for $n \leq 4$, but the inclusion is strict for $n \geq 5$. Indeed, the inequality:

$$\sum_{1 \leq i < j \leq 5} x_{ij} \leq 6 \tag{4.3}$$

is valid for CUT$_5$ (as any cut of K_5 has cardinality 0, 4 or 6), but it is not valid for MET$_5$. For instance, the vector $(2/3, \ldots, 2/3) \in \mathbb{R}^{10}$ belongs to MET$_5$ but it violates the inequality (4.3) (since $10.2/3 > 6$).

We can define the following linear programming bound:

$$\text{lp}(G, w) = \max \left\{ \sum_{\{i,j\} \in E(G)} w_{ij} x_{ij} : x \in \text{MET}_n \right\} \tag{4.4}$$

for the maximum cut:

$$\text{mc}(G, w) \leq \text{lp}(G, w).$$

The graphs for which this bound is tight have been characterized by Barahona [1]:

Theorem 4.1.2. *Let G be a graph. Then,* $\text{mc}(G, w) = \text{lp}(G, w)$ *for all weight functions $w \in \mathbb{R}^E$ if and only if the graph G has no K_5 minor.*

In particular, if G is a planar graph, then $\text{mc}(G, w) = \text{lp}(G, w)$ so that the maximum cut can be computed in polynomial time using linear programming.

A natural question is how good the LP bound is for general graphs. Here are some easy bounds.

Lemma 4.1.3. *Let G be a graph with nonnegative weights w. The following holds.*

(i) $\text{mc}(G, w) \leq \text{lp}(G, w) \leq w(E)$.

(ii) $\text{mc}(G, w) \geq w(E)/2$.

Proof. (i) follows from the fact that $\text{MET}_n \subseteq [0,1]^{E(K_n)}$ and $w \geq 0$. For (ii) pick $S \subseteq V$ for which $(S, \overline{S})$ is a cut of maximum weight: $w(S, \overline{S}) = \text{mc}(G, w)$. Thus if we move one node $i \in S$ to $\overline{S}$, or if we move one node $j \in \overline{S}$ to S, then we obtain another cut whose weight is at most $w(S, \overline{S})$. This gives:

$$w(S \setminus \{i\}, \overline{S} \cup \{i\}) - w(S, \overline{S}) = w(S \setminus \{i\}, \{i\}) - w(\{i\}, \overline{S}) \geq 0,$$

$$w(S \cup \{j\}, \overline{S} \setminus \{j\}) - w(S, \overline{S}) = w(\{j\}, \overline{S} \setminus \{j\}) - w(S, \{j\}) \geq 0.$$

Summing the first relation over $i \in S$ and using the fact that $2w(E(S)) = \sum_{i \in S} w(S \setminus \{i\}, \{i\})$, where $E(S)$ is the set of edges contained in S, and the fact that $\sum_{i \in S} w(\{i\}, \overline{S}) = w(S, \overline{S})$, we obtain:

$$2w(E(S)) \geq w(S, \overline{S}).$$

Analogously, summing over $j \in \overline{S}$, we obtain:

$$2w(E(\overline{S})) \geq w(S, \overline{S}).$$

Summing these two relations yields: $w(E(S)) + w(E(\overline{S})) \geq w(S, \overline{S})$. Now adding $w(S, \overline{S})$ to both sides implies: $w(E) \geq 2w(S, \overline{S}) = 2\text{mc}(G, w)$, which shows (ii). □

As an application of Lemma 4.1.3, we obtain that

$$\frac{1}{2} \leq \frac{\text{mc}(G, w)}{\text{lp}(G, w)} \leq 1 \quad \text{for all nonnegative weights } w \geq 0.$$

It turns out that there are graphs for which the ratio $mc(G, w)/\text{lp}(G, w)$ can be arbitrarily close to 1/2 [6]. This means that for these graphs, the metric polytope does not provide a better approximation of the cut polytope than its trivial relaxation by the hypercube $[0, 1]^E$.

We now provide another argument for the lower bound $\text{mc}(G, w) \geq w(E)/2$. This argument is probabilistic and based on the following simple randomized algorithm: Construct a random partition $(S, \overline{S})$ of V by assigning, independently, with probability 1/2, each node $i \in V$ to either side of the partition. Then the probability that an edge $\{i, j\}$ is cut by the partition is equal to

$$\mathbb{P}(\{i, j\} \text{ is cut}) = \mathbb{P}(i \in S, j \in \overline{S}) + \mathbb{P}(i \in \overline{S}, j \in S) = \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2}.$$

Hence, the expected weight of the cut produced by this random partition is equal to

$$\mathbb{E}(w(S, \overline{S})) = \sum_{\{i,j\} \in E} w_{ij} \mathbb{P}(\{i, j\} \text{ is cut}) = \sum_{\{i,j\} \in E} w_{ij} \frac{1}{2} = \frac{w(E)}{2}.$$

Here we have used the linearity of the expectation.

In the next section, we will see another probabilistic argument, due to Goemans and Williamson, which permits to construct a much better random cut. Namely we will get a random cut whose expected weight satisfies:

$$E(w(S, \overline{S})) \geq 0.878 \cdot w(E),$$

thus improving the above factor 0.5. The crucial tool will be to use a semidefinite relaxation for MAX CUT combined with a simple, but ingenious randomized "hyperplane rounding" technique.

4.2 The algorithm of Goemans and Williamson

4.2.1 Semidefinite programming relaxation

We now want to describe the Goemans-Williamson algorithm.

For this we first reformulate MAX CUT as a (non-convex) quadratic optimization problem having quadratic equality constraints. With every vertex $i \in V$, we associate a binary variable $x_i \in \{-1, +1\}$ which indicates whether i lies in S or in $\overline{S}$, say, $i \in S$ if $x_i = -1$ and $i \in \overline{S}$ if $x_i = +1$. We model the binary constraint $x_i \in \{-1, +1\}$ as a quadratic equality constraint

$$x_i^2 = 1 \text{ for } i \in V.$$

For two vertices $i, j \in V$ we have

$$1 - x_i x_j \in \{0, 2\}.$$

This value equals to 0 if i and j lie on the same side of the cut $(S, \overline{S})$ and the value equals to 2 if i and j lie on different sides of the cut. Hence, one can express the weight of the cut $(S, \overline{S})$ by

$$w(S, \overline{S}) = \sum_{\{i,j\} \in E} w_{ij} \frac{1 - x_i x_j}{2}.$$

Now, the MAX CUT problem can be equivalently formulated as

$$\text{mc}(G, w) = \max \left\{ \frac{1}{2} \sum_{\{i,j\} \in E} w_{ij}(1 - x_i x_j) : x_i^2 = 1 \; \forall i \in V \right\}. \tag{4.5}$$

Next, we introduce a matrix variable $X = (x_{ij}) \in \mathcal{S}^n$, whose entries x_{ij} model the pairwise products $x_i x_j$. Then, as the matrix $(x_i x_j)_{i,j=1}^n = xx^{\mathsf{T}}$ is positive semidefinite, we can require the condition that X should be positive semidefinite. Moreover, the constraints $x_i^2 = 1$ give the constraints $X_{ii} = 1$ for all $i \in [n]$. Therefore we can formulate the following semidefinite programming relaxation:

$$\text{sdp}(G, w) = \max \left\{ \frac{1}{2} \sum_{\{i,j\} \in E} w_{ij}(1 - X_{ij}) : X \succeq 0, \; X_{ii} = 1 \; \forall i \in [n] \right\}. \tag{4.6}$$

By construction, we have:

$$\text{mc}(G, w) \leq \text{sdp}(G, w). \tag{4.7}$$

The feasible region of the above semidefinite program is the convex (non-polyhedral) set

$$\mathcal{E}_n = \{X \in \mathcal{S}^n : X \succeq 0,\ X_{ii} = 1\ \forall i \in [n]\},$$

called the *elliptope* (and its members are known as *correlation matrices*). One can visualize the elliptope $\mathcal{E}_3$. Indeed, for a 3×3 symmetric matrix X with an all-ones diagonal, we have:

$$X = \begin{pmatrix} 1 & x & y \\ x & 1 & z \\ y & z & 1 \end{pmatrix} \succeq 0 \Longleftrightarrow 1 + 2xyz - x^2 - y^2 - z^2 \geq 0,\ x, y, z \in [-1, 1],$$

which expresses the fact that the determinant of X is nonnegative as well as the three 2×2 principal subdeterminants. The following Figure 4.2.1 visualizes the set of triples (x, y, z) for which $X \in \mathcal{E}_3$. Notice that the elliptope $\mathcal{E}_3$ looks like an “inflated” tetrahedron, while the underlying tetrahedron corresponds to the linear relaxation MET$_3$.

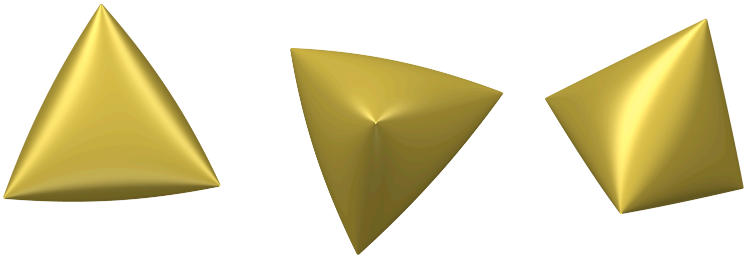


Figure 4.2: Views on the convex set $\mathcal{E}_3$ behind the semidefinite relaxation.

4.2.2 The Goemans-Williamson algorithm

Goemans and Williamson [4] show the following result for the semidefinite programming bound sdp(G, w).

Theorem 4.2.1. *Given a graph G with nonnegative edge weights w, the following inequalities hold:*

$$\text{sdp}(G, w) \geq \text{mc}(G, w) \geq 0.878 \cdot \text{sdp}(G, w).$$

The proof is algorithmic and it gives an approximation algorithm which approximates the MAX CUT problem within a ratio of 0.878. The Goemans-Williamson algorithm has five steps:

1. Solve the semidefinite program (4.6); let X be an optimal solution, so that sdp$(G, w) = \sum_{\{i,j\} \in E} w_{ij}(1 - X_{ij})/2$.

2. Perform a Cholesky decomposition of X to find unit vectors $v_i \in \mathbb{R}^{|V|-1}$ for $i \in V$, so that $X = (v_i^\mathsf{T} v_j)_{i,j \in V}$.

3. Choose a random unit vector $r \in \mathbb{R}^{|V|-1}$, according to the rotationally invariant probability distribution on the unit sphere.

4. Define a cut $(S, \overline{S})$ by setting $x_i = \operatorname{sign}(v_i^\mathsf{T} r)$ for all $i \in V$. That is, $i \in S$ if and only if $\operatorname{sign}(v_i^\mathsf{T} r) \leq 0$.

5. Check whether $\sum_{\{i,j\} \in E} w_{ij}(1 - x_i x_j)/2 \geq 0.878 \cdot \text{sdp}(G, w)$. If not, go to step 3.

The steps 3 and 4 in the algorithm are called a *randomized rounding procedure* because a solution of a semidefinite program is “rounded” (or better: projected) to a solution of the original combinatorial problem with the help of randomness.

Note also that because the expectation of the constructed solution is at least $0.878 \cdot \text{sdp}(G, w)$ the algorithm eventually terminates; it will pass step 5 and without getting stuck in an endless loop. One can show that with high probability we do not have to wait long until the condition in step 5 is fulfilled.

The following lemma (also known as *Grothendieck’s identity*, since it came up in work of Grothendieck in the 50’s, however in the different context of functional analysis) is the key to the proof of Theorem 4.2.1.

Lemma 4.2.2. *Let $u, v \in \mathbb{R}^d$ (for some $d \geq 1$) be unit vectors and let $r \in \mathbb{R}^d$ be a random unit vector chosen according to the rotationally invariant probability distribution on the unit sphere. The following holds.*

(i) The probability that $\operatorname{sign}(u^\mathsf{T} r) \neq \operatorname{sign}(v^\mathsf{T} r)$ is equal to

$$\mathbb{P}(\operatorname{sign}(u^\mathsf{T} r) \neq \operatorname{sign}(v^\mathsf{T} r)) = \frac{\arccos(u^\mathsf{T} v)}{\pi}. \tag{4.8}$$

(ii) The expectation of the random variable $\operatorname{sign}(u^\mathsf{T} r) \operatorname{sign}(v^\mathsf{T} r) \in \{-1, +1\}$ is equal to

$$\mathbb{E}[\operatorname{sign}(u^\mathsf{T} r) \operatorname{sign}(v^\mathsf{T} r)] = \frac{2}{\pi} \arcsin(u^\mathsf{T} v). \tag{4.9}$$

Proof. (i) Since the probability distribution from which we sample the unit vector r is rotationally invariant we can assume that u, v and r lie in a common plane. Hence we can assume that they lie on a unit circle in $\mathbb{R}^2$ and that r is chosen according to the uniform distribution on this circle. Then the probability that $\operatorname{sign}(u^\mathsf{T} r) \neq \operatorname{sign}(v^\mathsf{T} r)$ depends only on the angle between u and v. Using a figure (draw one!) it is easy to see that

$$\mathbb{P}[\operatorname{sign}(u^\mathsf{T} r) \neq \operatorname{sign}(v^\mathsf{T} r)] = 2 \cdot \frac{1}{2\pi} \arccos(u^\mathsf{T} v) = \frac{1}{\pi} \arccos(u^\mathsf{T} v).$$

(ii) By definition, the expectation $\mathbb{E}[\operatorname{sign}(u^\mathsf{T} r)\operatorname{sign}(v^\mathsf{T} r)]$ can be computed as

$$(+1)\cdot \mathbb{P}[\operatorname{sign}(u^\mathsf{T} r) = \operatorname{sign}(v^\mathsf{T} r)] + (-1)\cdot \mathbb{P}[\operatorname{sign}(u^\mathsf{T} r) \neq \operatorname{sign}(v^\mathsf{T} r)]$$

$$= 1 - 2\cdot \mathbb{P}[\operatorname{sign}(u^\mathsf{T} r) \neq \operatorname{sign}(v^\mathsf{T} r)] = 1 - 2\cdot \tfrac{\arccos(u^\mathsf{T} v)}{\pi},$$

where we have used (i) for the last equality. Now use the trigonometric identity

$$\arcsin t + \arccos t = \frac{\pi}{2},$$

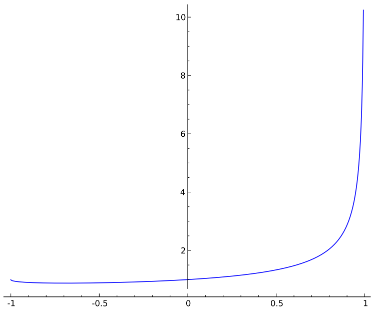
to conclude the proof of (ii). □

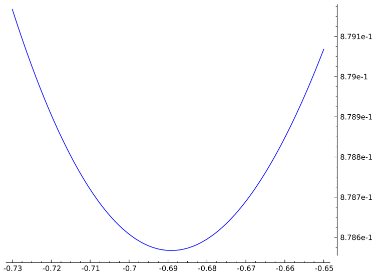
Using elementary univariate calculus one can show the following fact.

Lemma 4.2.3. *For all $t \in [-1, 1)]$, the following inequality holds:*

$$\frac{2}{\pi}\frac{\arccos t}{1-t} \geq 0.878. \tag{4.10}$$

One can also "see" this on the following plots of the function in (4.10), where t varies in $[-1, 1)$ in the first plot and in $[-0.73, -0.62]$ in the second plot.





Proof. (of Theorem 4.2.1) Let X be the optimal solution of the semidefinite program (4.6) and let $v_1, \ldots, v_n$ be unit vectors such that $X = (v_i^\mathsf{T} v_j)_{i,j=1}^n$, as in Steps 1,2 of the GW algorithm. Let $(S, \overline{S})$ be the random partition of V, as in Steps 3,4 of the algorithm. We now use Lemma 4.2.2(i) to compute the expected value of the cut $(S, \overline{S})$:

$$\mathbb{E}(w(S,\overline{S})) = \textstyle\sum_{\{i,j\}\in E} w_{ij}\mathbb{P}(\{i,j\} \text{ is cut}) = \sum_{\{i,j\}\in E} w_{ij}\mathbb{P}(x_i \neq x_j)$$

$$= \textstyle\sum_{\{i,j\}\in E} w_{ij}\mathbb{P}(\operatorname{sign}(v_i^\mathsf{T} r) \neq \operatorname{sign}(v_j^\mathsf{T} r)) = \sum_{\{i,j\}\in E} w_{ij}\, \frac{\arccos(v_i^\mathsf{T} v_j)}{\pi}$$

$$= \textstyle\sum_{\{i,j\}\in E} w_{ij}\left(\frac{1-v_i^\mathsf{T} v_j}{2}\right)\cdot\left(\frac{2}{\pi}\frac{\arccos(v_i^\mathsf{T} v_j)}{1-v_i^\mathsf{T} v_j}\right).$$

By Lemma 4.2.3, each term $\frac{2}{\pi}\frac{\arccos(v_i^\mathsf{T} v_j)}{1-v_i^\mathsf{T} v_j}$ can be lower bounded by the constant 0.878. Since all weights are nonnegative, each term $w_{ij}(1-v_i^\mathsf{T} v_j)$ is nonnegative.

Therefore, we can lower bound $\mathbb{E}(w(S,\overline{S}))$ in the following way:

$$\mathbb{E}(w(S,\overline{S})) \geq 0.878 \cdot \sum_{\{i,j\}\in E} w_{ij} \left(\frac{1 - v_i^{\mathsf{T}} v_j}{2}\right).$$

Now we recognize that the objective value sdp(G,w) of the semidefinite program is appearing in the right hand side and we obtain:

$$\mathbb{E}(w(S,\overline{S})) \geq 0.878 \cdot \sum_{\{i,j\}\in E} w_{ij} \left(\frac{1 - v_i^{\mathsf{T}} v_j}{2}\right) = 0.878 \cdot \text{sdp}(G,w).$$

Finally, it is clear that the maximum weight of a cut is at least the expected value of the random cut $(S,\overline{S})$:

$$\text{mc}(G,w) \geq \mathbb{E}(w(S,\overline{S})).$$

Putting things together we can conclude that

$$\text{mc}(G,w) \geq \mathbb{E}(w(S,\overline{S})) \geq 0.878 \cdot \text{sdp}(G,w).$$

This concludes the proof, since the other inequality mc$(G,w) \leq$ sdp(G,w) holds by (4.7). □

4.2.3 Remarks on the algorithm

It remains to give a procedure which samples a random vector from the unit sphere. This can be done if one can sample random numbers from the standard normal (Gaussian) distribution (with mean zero and variance one) which has probability density

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}.$$

Many software packages include a procedure which produces random numbers from the standard normal distribution.

If we sample n real numbers $x_1, \ldots, x_n$ independently uniformly at random from the standard normal distribution, then, the vector

$$r = \frac{1}{\sqrt{x_1^2 + \cdots + x_n^2}} (x_1, \ldots, x_n)^{\mathsf{T}} \in S^{n-1}$$

is distributed according to the rotationally invariant probability measure on the unit sphere.

Finally we mention that one can modify the Goemans-Williamson algorithm so that it becomes an algorithm which runs deterministically (without the use of randomness) in polynomial time and which gives the same approximation ratio. This was done by Mahajan and Ramesh in 1995.

4.3 Extensions

4.3.1 Reformulating MAX CUT using the Laplacian matrix

Given a graph G with edge weights w, its *Laplacian matrix* L_w is the symmetric $n \times n$ matrix with entries:

$$(L_w)_{ii} = \sum_{j:\{i,j\} \in E} w_{ij} \ (i \in [n]),$$

$$(L_w)_{ij} = -w_{ij} \ (\{i,j\} \in E), \ (L_w)_{ij} = 0 \ (i \neq j, \{i,j\} \notin E).$$

The following can be checked (Exercise 4.2).

Lemma 4.3.1. *The following properties hold for the Laplacian matrix L_w:*

(i) For any vector $x \in \{\pm 1\}^n$, $\frac{1}{4} x^\mathsf{T} L_w x = \frac{1}{2} \sum_{\{i,j\} \in E} w_{ij}(1 - x_i x_j)$.

(ii) For any nonnegative edge weights $w \geq 0$, $L_w \succeq 0$.

This permits to reformulate the quadratic formulation (4.5) of MAX CUT as

$$\text{mc}(G,w) = \max \left\{ \frac{1}{4} x^\mathsf{T} L_w x : x_i^2 = 1 \ \forall i \in V \right\}$$

and its semidefinite relaxation (4.6) as

$$\text{sdp}(G,w) = \max \left\{ \frac{1}{4} \langle L_w, X \rangle : X \succeq 0, \ X_{ii} = 1 \ \forall i \in V \right\}.$$

A property of the above programs is that the matrix $L_w/4$ occurring in the objective function is positive semidefinite. In the next section we consider general quadratic programs, where L_w is replaced by an arbitrary positive semidefinite matrix A. Then one can still show an approximation algorithm, however with performance ration $\frac{2}{\pi} \sim 0.636$, thus weaker than the 0.878 ratio of Goemans and Williamson for the case when $A = L_w$ for some $w \geq 0$.

4.3.2 Nesterov's approximation algorithm

Nesterov [5] considers the class of quadratic problems:

$$\text{qp}(A) = \max \left\{ \sum_{i,j=1}^{n} A_{ij} x_i x_j : x_i^2 = 1 \ \forall i \in [n] \right\}, \tag{4.11}$$

where $A \in \mathcal{S}^n$ is a symmetric matrix. (Thus, $\text{qp}(A) = \text{mc}(G,w)$ for $A = L_w/4$). Analogously define the semidefinite programming relaxation:

$$\text{sdp}(A) = \max \left\{ \langle A, X \rangle : X \succeq 0, \ X_{ii} = 1 \ \forall i \in [n] \right\}. \tag{4.12}$$

The following inequality holds:

$$\text{qp}(A) \leq \text{sdp}(A)$$

for any symmetric matrix A. In the special case when A is positive semidefinite, Nesterov shows that sdp(A) is a $\frac{2}{\pi}$-approximation for qp(A). The proof is based on the same rounding technique of Goemans-Williamson, but the analysis is different. It relies on the following property of the function $\arcsin t$: There exist positive scalars $a_k > 0$ ($k \geq 0$) such that

$$\arcsin t = t + \sum_{k\geq 0} a_k t^{2k+1} \text{ for all } t \in [-1,1]. \tag{4.13}$$

Based on this one can show the following result.

Lemma 4.3.2. *Given a matrix $X = (x_{ij}) \in \mathcal{S}^n$, define the new matrix*

$$\tilde{X} = (\arcsin X_{ij} - X_{ij})_{i,j=1}^n,$$

whose entries are the images of the entries of X under the map $t \mapsto \arcsin t - t$. Then, $X \succeq 0$ implies $\tilde{X} \succeq 0$.

Proof. The proof uses the following fact: If $X = (x_{ij})_{i,j=1}^n$ is positive semidefinite then, for any integer $k \geq 1$, the matrix $(X_{ij}^k)_{i,j=1}^n$ (whose entries are the k-th powers of the entries of X) is positive semidefinite as well. (Recall Section 1.2.2 of Chapter 1.)
Using this fact, the form of the series decomposition (4.13), and taking limits, implies the result of the lemma. □

Theorem 4.3.3. *Assume A is a positive semidefinite matrix. Then,*

$$\text{sdp}(A) \geq \text{qp}(A) \geq \frac{2}{\pi}\text{sdp}(A).$$

Proof. Let X be an optimal solution of the semidefinite program (4.12) and let $v_1, \ldots, v_n$ be unit vectors such that $X = (v_i^{\mathsf{T}} v_j)_{i,j=1}^n$ (as in Steps 1,2 of the GW algorithm). Pick a random unit vector r and set $x_i = \text{sign}(v_i^{\mathsf{T}} r)$ for $i \in V$ (as in Steps 3,4 of the GW algorithm). We now use Lemma 4.2.2(ii) to compute the expected value of $\sum_{i,j=1}^n A_{ij} x_i x_j$:

$$\begin{aligned}
&\mathbb{E}(\textstyle\sum_{i,j=1}^n A_{ij} x_i x_j) = \sum_{i,j=1}^n A_{ij} \mathbb{E}(x_i x_j) \\
&= \tfrac{2}{\pi} \textstyle\sum_{i,j=1}^n A_{ij} \arcsin(v_i^{\mathsf{T}} v_j) = \tfrac{2}{\pi} \sum_{i,j=1}^n A_{ij} \arcsin X_{ij} \\
&= \tfrac{2}{\pi} \left(\textstyle\sum_{i,j=1}^n A_{ij} X_{ij} + \sum_{i,j=1}^n A_{ij} (\arcsin X_{ij} - X_{ij}) \right).
\end{aligned}$$

By Lemma 4.3.2, the second term is equal to $\langle A, \tilde{X} \rangle \geq 0$, since $\tilde{X} \succeq 0$. Moreover, we recognize in the first term the objective value of the semidefinite program

(4.12). Combining these facts, we obtain:

$$\mathbb{E}(\sum_{i,j=1}^{n} A_{ij}x_ix_j) \geq \frac{2}{\pi}\text{sdp}(A).$$

On the other hand, it is clear that

$$\text{qp}(A) \geq \mathbb{E}(\sum_{i,j=1}^{n} A_{ij}x_ix_j).$$

This concludes the proof. □

4.3.3 Quadratic programs modeling MAX 2SAT

Here we consider another class of quadratic programs, of the form:

$$\text{qp}(a,b) = \max\left\{\sum_{ij\in E_1} a_{ij}(1-x_ix_j) + \sum_{ij\in E_2} b_{ij}(1+x_ix_j) : x \in \{\pm 1\}^n\right\}, \tag{4.14}$$

where $a_{ij}, b_{ij} \geq 0$ for all ij. Write the semidefinite relaxation:

$$\text{sdp}(a,b) = \max\left\{\sum_{ij\in E_1} a_{ij}(1-X_{ij}) + \sum_{ij\in E_2} b_{ij}(1+X_{ij}) : X \succeq 0,\ X_{ii} = 1\ \forall i \in [n]\right\}. \tag{4.15}$$

Goemans and Williamson [4] show that the same approximation result holds as for MAX CUT:

Theorem 4.3.4. *Assume that $a, b \geq 0$. Then,*

$$\text{sdp}(a,b) \geq \text{qp}(a,b) \geq 0.878 \cdot \text{sdp}(a,b).$$

In the proof we will use the following variation of Lemma 4.2.3.

Lemma 4.3.5. *For any $z \in [-1,1]$, the following inequality holds:*

$$\frac{2}{\pi}\frac{\pi - \arccos z}{1+z} \geq 0.878.$$

Proof. Set $t = -z \in [-1,1]$. Using the identity $\arccos(-t) = \pi - \arccos t$ and applying (4.10), we get: $\frac{2}{\pi}\frac{\pi-\arccos z}{1+z} = \frac{2}{\pi}\frac{\arccos t}{1-t} \geq 0.878$. □

Proof. (of Theorem 4.3.4) We apply the GW algorithm: Let $X = (v_i^{\mathsf{T}} v_j)$ be an optimal solution of (4.15). Pick a random unit vector r and set $x_i = \text{sign}(v_i^{\mathsf{T}} r)$ for $i \in [n]$. Using the fact that $\mathbb{E}(x_ix_j) = 1-2\cdot\mathbb{P}(x_i \neq x_j) = 1-2\cdot\frac{\arccos(v_i^{\mathsf{T}} v_j)}{\pi}$, we can compute the expected value of the quadratic objective of (4.14) evaluated at x:

$$\begin{aligned}
&\mathbb{E}\left(\textstyle\sum_{ij\in E_1} a_{ij}(1-x_ix_j)+\sum_{ij\in E_2} b_{ij}(1+x_ix_j)\right)\\
&= 2\cdot\textstyle\sum_{ij\in E_1} a_{ij}\frac{\arccos(v_i^\mathsf{T}v_j)}{\pi}+2\cdot\sum_{ij\in E_2} b_{ij}\left(1-\frac{\arccos(v_i^\mathsf{T}v_j)}{\pi}\right)\\
&= \textstyle\sum_{ij\in E_1}\underbrace{a_{ij}(1-v_i^\mathsf{T}v_j)}_{\geq 0}\underbrace{\frac{2}{\pi}\frac{\arccos(v_i^\mathsf{T}v_j)}{1-v_i^\mathsf{T}v_j}}_{\geq 0.878}+\sum_{ij\in E_2}\underbrace{b_{ij}(1+v_i^\mathsf{T}v_j)}_{\geq 0}\underbrace{\frac{2}{\pi}\frac{\pi-\arccos(v_i^\mathsf{T}v_j)}{1+v_i^\mathsf{T}v_j}}_{\geq 0.878}\\
&\geq 0.878\cdot \mathrm{sdp}(a,b).
\end{aligned}$$

Here we have used Lemmas 4.2.3 and 4.3.5. From this we can conclude that $\mathrm{qp}(a,b)\geq 0.878\cdot\mathrm{sdp}(a,b)$. □

In the next section we indicate how to use the quadratic program (4.14) in order to formulate MAX 2 SAT.

4.3.4 Approximating MAX 2-SAT

An instance of MAX SAT is given by a collection of Boolean clauses $C_1,\ldots,C_m$, where each clause C_j is a disjunction of literals, drawn from a set of variables $\{z_1,\ldots,z_n\}$. A literal is a variable z_i or its negation $\overline{z}_i$. Moreover there is a weight w_j attached to each clause C_j. The MAX SAT problem asks for an assignment of truth values to the variables $z_1,\ldots,z_n$ that maximizes the total weight of the clauses that are satisfied. MAX 2SAT consists of the instances of MAX SAT where each clause has at most two literals. It is an NP-complete problem [3] and analogously to MAX CUT it is also hard to approximate.

Goemans and Williamson show that their randomized algorithm for MAX CUT also applies to MAX 2SAT and yields again a 0.878-approximation algorithm. Prior to their result, the best approximation was 3/4, due to Yannakakis (1994).

To show this it suffices to model MAX 2SAT as a quadratic program of the form (4.14). We now indicate how to do this. We introduce a variable $x_i\in\{\pm 1\}$ for each variable z_i of the SAT instance. We also introduce an additional variable $x_0\in\{\pm 1\}$ which is used as follows: z_i is true if $x_i=x_0$ and false otherwise.

Given a clause C, define its value $v(C)$ to be 1 if the clause C is true and 0 otherwise. Thus,

$$v(z_i)=\frac{1+x_0x_i}{2},\ v(\overline{z}_i)=1-v(z_i)=\frac{1-x_0x_i}{2}.$$

Based on this one can now express $v(C)$ for a clause with two literals:

$$\begin{aligned}
v(z_i\vee z_j)=1-v(\overline{z}_i\wedge\overline{z}_j)=1-v(\overline{z}_i)v(\overline{z}_j)=1-\tfrac{1-x_0x_i}{2}\tfrac{1-x_0x_j}{2}\\
=\tfrac{1+x_0x_i}{4}+\tfrac{1+x_0x_j}{4}+\tfrac{1-x_ix_j}{4}.
\end{aligned}$$

Analogously, one can express $v(z_i \vee \overline{z}_j)$ and $v(\overline{z}_i \vee \overline{z}_j)$, by replacing x_i by $-x_i$ when z_i is negated. In all cases we see that $v(C)$ is a linear combination of terms of the form $1 + x_i x_j$ and $1 - x_i x_j$ with *nonnegative* coefficients.

Now MAX 2SAT can be modelled as

$$\max\{\sum_{j=1}^{m} w_j v(C_j) : x_1^2 = \ldots = x_n^2 = 1\}.$$

This quadratic program is of the form (4.14). Hence Theorem 4.3.4 applies. Therefore, the approximation algorithm of Goemans and Williamson gives a 0.878 approximation for MAX 2SAT.

4.4 Further reading and remarks

We start with an anecdote. About the finding of the approximation ratio 0.878, Knuth writes in the article "Mathematical Vanity Plates":

> Sometimes people obtain mathematically significant license plates purely by accident, without making a personal selection. A striking example of this phenomenon is the case of Michel Goemans, who received the following innocuous-looking plate from the Massachusetts Registry of Motor Vehicles when he and his wife purchased a Subaru at the beginning of September 1993:

> Two weeks later, Michel got together with his former student David Williamson, and they suddenly realized how to solve a problem that they had been working on for some years: to get good approximations for maximum cut and satisfiability problems by exploiting semidefinite programming. Lo and behold, their new method—which led to a famous, award-winning paper [15]—yielded the approximation factor .878! There it was, right on the license, with `C`, `S`, and `W` standing respectively for cut, satisfiability, and Williamson.

For their work [4], Goemans and Williamson won in 2000 the Fulkerson prize (sponsored jointly by the Mathematical Programming Society and the AMS) which recognizes outstanding papers in the area of discrete mathematics for this result.

How good is the MAX CUT algorithm? Are there graphs where the value of the semidefinite relaxation and the value of the maximal cut are a factor of 0.878 apart or is this value 0.878, which maybe looks strange at first sight, only an artefact of our analysis? It turns out that the value is optimal. In 2002 Feige and Schechtmann gave an infinite family of graphs for which the ratio mc/sdp converges to exactly $0.878\ldots$. This proof uses a lot of nice mathematics (continuous graphs, Voronoi regions, isoperimetric inequality) and it is explained in detail in the Chapter 8 of the book *Approximation Algorithms and Semidefinite Programming* of Gärtner and Matoušek.

In 2007, Khot, Kindler, Mossel, O'Donnell showed that the algorithm of Goemans and Williamson is optimal in the following sense: If the unique games conjecture is true, then there is no polynomial time approximation algorithm achieving a better approximation ratio than 0.878 unless $\mathrm{P} = \mathrm{NP}$. Currently, the validity and the implications of the unique games conjecture are under heavy investigation. The book of Gärtner and Matoušek also contains an introduction to the unique games conjecture.

4.5 Exercises

4.1 The goal of this exercise is to show that the maximum weight stable set problem can be formulated as an instance of the maximum cut problem.

Let $G = (V, E)$ be a graph with node weights $c \in \mathbb{R}^V_+$. Define the new graph $G' = (V', E')$ with node set $V' = V \cup \{0\}$, with edge set $E' = E \cup \{\{0, i\} : i \in V\}$, and with edge weights $w \in \mathbb{R}^{E'}_+$ defined by

$$w_{0i} = c_i - \deg_G(i)M \text{ for } i \in V, \text{ and } w_{ij} = M \text{ for } \{i,j\} \in E.$$

Here, $\deg_G(i)$ denotes the degree of node i in G, and M is a constant to be determined.

(a) Let $S \subseteq V$. Show: $w(S, V' \setminus S) = c(S) - 2M|E(S)|$.

(b) Show: If M is sufficiently large, then $S \subseteq V$ is a stable set of maximum weight in (G, c) if and only if $(S, V' \setminus S)$ is a cut of maximum weight in (G', w).

Give an explicit value of M for which the above holds.

4.2 Let $G = (V = [n], E)$ be a graph with edge weights $w \in \mathbb{R}^E$. Define the Laplacian matrix $L_w \in \mathcal{S}^n$ by: $L_{ii} = \sum_{j \in V : \{i,j\} \in E} w_{ij}$ for $i \in V$, $L_{ij} = -w_{ij}$ if $\{i,j\} \in E$, and $L_{ij} = 0$ otherwise.

(a) Show: $x^{\mathsf{T}} L_w x = 2 \cdot \sum_{\{i,j\} \in E} w_{ij}(1 - x_i x_j)$ for any vector $x \in \{\pm 1\}^n$.

(b) Show: If $w \geq 0$ then $L_w \succeq 0$.

(c) Given an example of weights w for which L_w is not positive semidefinite.

4.3 Let $G = (V = [n], E)$ be a graph and let $w \in \mathbb{R}^E_+$ be nonnegative edge weights.

(a) Show the following reformulation for the MAX CUT problem:

$$\mathrm{mc}(G, w) = \max \left\{ \sum_{\{i,j\} \in E} w_{ij} \frac{\arccos(v_i^\mathsf{T} v_j)}{\pi} : v_1, \ldots, v_n \text{ unit vectors in } \mathbb{R}^n \right\}.$$

Hint: Use the analysis of the Goemans-Williamson algorithm.

(b) Let $v_1, \ldots, v_7$ be unit vectors. Show:

$$\sum_{1 \leq i < j \leq 7} \arccos(v_i^\mathsf{T} v_j) \leq 12\pi.$$

4.4 For a matrix $A \in \mathbb{R}^{m \times n}$ we define the following quantities:

$$f(A) = \max_{I \subseteq [m], J \subseteq [n]} |\sum_{i \in I} \sum_{j \in J} A_{ij}|,$$

called the cut norm of A, and

$$g(A) = \max \left\{ \sum_{i \in [m]} \sum_{j \in [n]} A_{ij} x_i y_j : x_1, \ldots, x_m, y_1, \ldots, y_n \in \{\pm 1\} \right\}.$$

(a) Show: $f(A) \leq g(A) \leq 4f(A)$.

(b) Assume that all row sums and all column sums of A are equal to 0. Show: $g(A) = 4f(A)$.

(c) Formulate a semidefinite programming relaxation for $g(A)$.

(d) Show:

$$g(A) = \max \left\{ \sum_{i \in [m]} \sum_{j \in [n]} A_{ij} x_i y_j : x_1, \ldots, x_m, y_1, \ldots, y_n \in [-1, 1] \right\}.$$

(e) Assume that A is a symmetric positive semidefinite $n \times n$ matrix. Show:

$$g(A) = \max \left\{ \sum_{i=1}^n \sum_{j=1}^n A_{ij} x_i x_j : x_1, \ldots, x_n \in \{\pm 1\} \right\}.$$

(f) Show that the maximum cut problem in a graph $G = ([n], E)$ with nonnegative edge weights can be formulated as an instance of computing the cut norm $f(A)$ of some matrix A.

4.5 Let $G = C_5$ denote the cycle on 5 nodes. Compute the Goemans-Williamson semidefinite relaxation for max-cut (where all edge weights are taken equal to 1):

$$\text{sdp}(C_5) = \max \left\{ \frac{1}{2} \sum_{i=1}^{5} (1 - X_{i,i+1}) : X \in \mathcal{S}^5, X \succeq 0, X_{ii} = 1 \ \forall i \in [5] \right\}.$$

How does the ratio $\frac{\text{mc}(C_5)}{\text{sdp}(C_5)}$ compare to the GW ratio 0.878?

BIBLIOGRAPHY

[1] F. Barahona. The max-cut problem in graphs not contractible to K_5. *Operations Research Letters* **2**:107–111, 1983.

[2] M. Deza and M. Laurent. *Geometry of Cuts and Metrics.* Springer, 1997.

[3] M.R. Garey and D.S. Johnson. *Computers and Intractability - A guide to the Theory of NP-Completeness*. Freeman, 1979.

[4] M.X. Goemans, D.P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. ACM **42**:1115–1145, 1995.

[5] Y. Nesterov. Quality of semidefinite relaxation for nonconvex quadratic optimization. *CORE Discussion Paper*, Number 9719, 1997.

[6] S. Poljak and Z. Tuza. The expected relative error of the polyhedral approximation of the max-cut problem. *Operations Research Letters* **16**:191–1998, 1994.

[7] A. Schrijver. *A Course in Combinatorial Optimization*. Lecture Notes. Available at `http://homepages.cwi.nl/~lex/files/dict.pdf`