Chapter VIII: SDP in Combinatorial Optimization

1 Stable Sets and Graph Coloring

In this section, we consider a simple, undirected graph $G = (V, E)$. Formally, an edge between the edges $u \in V$ and $v \in V$ should be represented by an unordered pair $\{u, v\} \in E$, but we will write $uv \in E$ for the sake of simplicity. In particular, $uv \in E \iff vu \in E$.

> **Definition 1** (Stable set)**.** Let G be a simple graph. A subset of vertices $S \subseteq V$ is called *stable* (or *independent*) if
> $$u \in S, v \in S \implies uv \notin E.$$

The stable set problem is defined as follows: *Given a simple graph $G = (V, E)$, find the largest stable set of G.* The *stability number* (or *independent number*) of the graph is defined as the cardinality of the largest stable set. We denote it by $\alpha(G)$.

The stable set problem admits the following formulation as an integer linear program:

$$\begin{aligned} \alpha(G) = \max_{\boldsymbol{x}} \quad & \sum_{v \in V} x_v \\ s.t. \quad & x_u + x_v \leq 1, \quad \forall uv \in E \\ & \boldsymbol{x} \in \{0,1\}^V \end{aligned} \tag{1}$$

The fractional relaxation of the above IP, where the constraints $x_v \in \{0, 1\}$ are replaced by $0 \leq x_v \leq 1$, can be shown to be half-integer (there always exists an optimal solution such that $x_v \in \{0, \frac{1}{2}, 1\}$, $\forall v \in V$). These half-integer solutions can be quite poor; for example, it is easy to see that for the complete graph K_n, the fractional solution of (1) is $x_v = \frac{1}{2}$ for all $v \in V$, which yields the fractional value of $\sum_{v \in V} x_v = \frac{n}{2}$, while $\alpha(K_n) = 1$.

We will see that semidefinite programming allows us to formulate a stronger relaxation, which even yields an exact algorithm for a large class of graphs.

> **Definition 2** (Clique number)**.** A clique of G is a subset $U \subseteq V$ such that $u, v \in U \implies uv \in E$. The largest cardinality of a clique is called the clique number, and is denoted by $\omega(G)$.

> **Definition 3** (Chromatic number)**.** A k-coloring of G is a partition of its vertices in exactly k stable sets. The chromatic number of G is the smallest number k such that a $k-$coloring exists, and is denoted by $\chi(G)$

Definition 4 (Clique cover number)**.** A k-clique cover of G is a partition of its vertices in exactly k cliques. The clique cover number of G is the smallest number k such that a $k-$clique cover exists, and is denoted by $\bar{\chi}(G)$.

To summarize,

$$\begin{aligned}
\alpha(G) &= \max\{|S| : S \text{ is a stable set of } G\} && \text{[stability number]}\\
\omega(G) &= \max\{|C| : C \text{ is a clique of } G\} && \text{[clique number]}\\
\chi(G) &= \min\{k : V = S_1 \uplus \cdots \uplus S_k, \text{where the } S_i\text{'s are stable sets of } G\} && \text{[chromatic number]}\\
\bar{\chi}(G) &= \min\{k : V = C_1 \uplus \cdots \uplus C_k, \text{where the } C_i\text{'s are cliques of } G\}. && \text{[clique cover number]}
\end{aligned}$$

Let $\bar{G}$ denote the complementary graph of G, that is, $\bar{G} = (V, \bar{E})$, where $\bar{E} = E_K \setminus E$ and E_K is the set of edges on the complete graph over V. In other words, i and j are connected in $\bar{G}$ iff they are not connected in G. It is straigtforward to show that

Proposition 1. *$\alpha(G) = \omega(\bar{G})$ and $\chi(G) = \bar{\chi}(\bar{G})$.*

Proof. This results from the simple observation

$$S \text{ is stable in } G \iff S \text{ is a clique in } \bar{G}.$$

□

It is also clear that

$$\omega(G) \leq \chi(G),$$

because each vertex of a clique must be assigned to different colors. Hence, by taking the complementary graph, we obtain

$$\alpha(G) \leq \bar{\chi}(G).$$

A graph in which the equality $\omega(G') = \chi(G')$ holds for all induced subgraphs G' of G (including $G' = G$) is called a *perfect graph*. Lovász showed in 1972 that a graph is perfect iff its complementary is perfect. Hence, in a perfect graph it also holds that $\alpha(G) = \bar{\chi}(G)$. The class of perfect graphs was finally characterized in 2006 by Chudnovsky, Robertson, Seymour, and Thomas [3], in a theorem that we will not prove:

Theorem 2. *A graph G is perfect if and only if neither G nor $\bar{G}$ contains an odd cycle of length ≥ 5 as an induced subgraph.*

We also point out that there is a polynomial-time algorithm that recognizes whether a graph is perfect.

We will now present what is often refered as the Sandwich theorem of Lovász. It states that there is a function of G, equal to the value of some SDP, that lies between $\alpha(G)$ and $\bar{\chi}(G)$. As a consequence, $\alpha(G)$ and $\bar{\chi}(G)$ can be computed in polynomial time if G is perfect. In contrast, note that computing any of $\alpha(G)$ or $\chi(G)$ is NP-hard for general graphs.

Definition 5. The theta-function of Lovász is defined for all graphs G, as follows:

$$\begin{aligned} \vartheta(G) = \max_{X\in\mathbb{S}^n} \quad & \langle J, X\rangle \\ s.t. \quad & \langle I, X\rangle = 1 \\ & X_{ij} = 0, \quad \forall ij \in E \\ & X \succeq 0. \end{aligned} \tag{2}$$

Recall that I denotes the identity matrix and $J = \mathbf{1}\mathbf{1}^T$ is the matrix of all-ones. Hence, $\langle I, X\rangle = \operatorname{trace} X$ and $\langle J, X\rangle = \sum_{i=1}^n \sum_{j=1}^n X_{ij}$.

We can also define $\vartheta(G)$ by the dual SDP:

Proposition 3.

$$\begin{aligned} \vartheta(G) = \min_{t, Z\in\mathbb{S}^n} \quad & t \\ s.t. \quad & Z \preceq tI \\ & Z_{ij} = 1, \quad \forall (i,j) \ \textit{such that} \quad (i=j \quad \textit{or} \quad ij \in \bar{E}). \end{aligned} \tag{3}$$

This SDP can also be rewritten as an eigenvalue problem:

$$\vartheta(G) = \min_{Z\in\mathcal{Z}} \ \lambda_{\max}(Z),$$

where $\mathcal{Z} := \{Z \in \mathbb{S}^n | \quad Z_{ii} = 1,\ \forall i \in [n]; \quad Z_{ij} = 1,\ \forall\, ij \in \bar{E}\}$.

Proof. We start to show that the two SDPs are dual from each other. To this end, we first rewrite the max SDP as a saddle point (max-min) problem:

$$\vartheta(G) = \sup_{X\succeq 0} \langle J, X\rangle \ + \ \inf_{t\in\mathbb{R}} \ t\cdot(1 - \langle I, X\rangle) \ + \ \sum_{ij\in E} \inf_{u_{ij}\in\mathbb{R}} u_{ij}\cdot X_{ij}$$

The dual problem is obtained by switching the order of sup and inf. By weak duality,

$$\vartheta(G) \leq \vartheta'(G) := \inf_{t\in\mathbb{R}, u_{ij}\in\mathbb{R}} \ t \ + \ \sup_{X\succeq 0} \Big\langle X,\ J - t\,I + \sum_{ij\in E} u_{ij}\ E_{i,j}\Big\rangle,$$

where $E_{i,j} = \frac{1}{2}(\boldsymbol{e}_i\boldsymbol{e}_j^T + \boldsymbol{e}_j\boldsymbol{e}_i^T)$ is the matrix with $\frac{1}{2}$ on the (i,j)- and (j,i)-coordinates and 0's elsewhere, so it holds $X_{ij} = \langle X, E_{ij}\rangle$ for $i \neq j$. The above supremum is finite (and has value 0) if and only if $J - t\,I + \sum_{ij\in E} u_{ij}\ E_{i,j} \preceq 0$. Hence,

$$\begin{aligned} \vartheta'(G) \ = \inf_{t\in\mathbb{R}, u_{ij}\in\mathbb{R}} \quad & t \\ s.t. \quad & J + \sum_{ij\in E} u_{ij}\ E_{i,j} \preceq t\,I, \end{aligned}$$

Then, the dual SDP of the proposition is obtained by making the change of variable $Z = J + \sum_{ij\in E} u_{ij}\ E_{i,j}$, which is a symmetric matrix with arbitrary entries on coordinates (i,j) where $ij \in E$, and with ones elsewhere, that is, $Z \in \mathcal{Z}$. The formulation as an eigenvalue problem follows from the SDP-representation of $\lambda_{\max}(\cdot)$.

It remains to show that strong duality holds, so $\vartheta(G) = \vartheta'(G)$. We are going to show that both SDPs are strictly feasible, which also implies that they are also bounded and attain their optimal values (so we can safely write "max" and "min" in the formulations of $\vartheta(G)$ instead of "sup" and "inf").

The matrix $X = \frac{1}{n} I_n$ is clearly strictly feasible for the primal SDP (the maximization problem). For the dual SDP, we observe that $Z \prec tI$ iff $\lambda_{\max}(Z) < t$. We can thus take an arbitrary matrix $Z \in \mathcal{Z}$, and choose $t > \lambda_{\max}(Z)$, so the pair (Z, t) is strictly feasible for the dual SDP.

$\square$

In the exercises, we will give yet another alternative SDP formulation of $\vartheta(G)$, which can be derived *in a systematic way*, as a relaxation from the integer quadratic programming formulation of $\alpha(G)$. We are now ready to prove the

Theorem 4 (Lovász's Sandwich theorem [7]).

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

Proof. For the first inequality, $\alpha(G) \leq \vartheta(G)$, let S be a maximal stable set, and denote by $\boldsymbol{e}_S$ the incidence vector of S: $\boldsymbol{e}_S$ is the $\{0,1\}$-vector of size n with a one on the ith coordinate iff $i \in S$. Then, we can see that $X = \frac{1}{|S|}\, \boldsymbol{e}_S \boldsymbol{e}_S^T$ is feasible for (2). Indeed, $X \succeq 0$ because it is defined as a tensor product of two vectors, $\operatorname{trace} X = \frac{1}{|S|} \operatorname{trace} \boldsymbol{e}_S \boldsymbol{e}_S^T = \frac{1}{|S|} \boldsymbol{e}_S^T \boldsymbol{e}_S = 1$, and for all $ij \in E$, we have ($i \notin S$ or $j \notin S$) because S is a stable, so $(\boldsymbol{e}_S \boldsymbol{e}_S^T)_{ij} = 0 \implies X_{ij} = 0$. This shows that $\vartheta(G) \geq \langle J, X \rangle$, since the optimal value of the SDP is at least as large as the value of the particular solution X:

$$\vartheta(G) \geq \langle J, X \rangle = \frac{1}{|S|} \boldsymbol{e}_S^T J \boldsymbol{e}_S = \frac{1}{|S|} (\boldsymbol{e}_S^T \mathbf{1})^2 = \frac{1}{|S|} (|S|)^2 = |S| = \alpha(G).$$

We will proceed similarly for the second inequality, by identifying a feasible solution of value $\bar{\chi}(G)$ for the dual SDP. Let $C_1, \ldots, C_k$ be a minimal $k-$clique cover of G, and denote by $\boldsymbol{e}_{C_j}$ the incidence vector of C_j. We claim that $Z = k\,I - \frac{1}{k} \sum_{j=1}^k (k \boldsymbol{e}_{C_j} - \mathbf{1})(k \boldsymbol{e}_{C_j} - \mathbf{1})^T$ and $t = k$ are feasible for the SDP (3). Indeed, $t\,I - Z = \frac{1}{k} \sum_{j=1}^k (k \boldsymbol{e}_{C_j} - \mathbf{1})(k \boldsymbol{e}_{C_j} - \mathbf{1})^T$ is a sum of rank-one positive semidefinite matrices, and $t\,I \succeq Z$. Then, by observing that $\sum_{j=1}^k \boldsymbol{e}_{C_j} = \mathbf{1}$ (because the C_j's form a partition of V), we can expand the expression of Z:

$$\begin{aligned} Z = k\,I - \frac{1}{k} \sum_{j=1}^k (k \boldsymbol{e}_{C_j} - \mathbf{1})(k \boldsymbol{e}_{C_j} - \mathbf{1})^T &= k\,I - k \sum_{j=1}^k \boldsymbol{e}_{C_j} \boldsymbol{e}_{C_j}^T + 2J - J \\ &= k\,(I - \sum_{j=1}^k \boldsymbol{e}_{C_j} \boldsymbol{e}_{C_j}^T) + J. \end{aligned}$$

Finally, the elements of the matrix $U := (I - \sum_{j=1}^k \boldsymbol{e}_{C_j} \boldsymbol{e}_{C_j}^T)$ are

$$U_{ij} = \begin{cases} -1 & \text{if } i \neq j \text{ belong to the same clique;} \\ 0 & \text{otherwise,} \end{cases}$$

so the diagonal elements of Z are $Z_{ii} = J_{ii} = 1$, and for $ij \in \bar{E}$, the vertices i and j must belong to different cliques, so $Z_{ij} = J_{ij} = 1$. This shows $Z \in \mathcal{Z}$, and $\vartheta(G) \leq t = k = \bar{\chi}(G)$. $\square$

We will see in the exercises that Semidefinite Programming can also be used to compute a maximum stable set in a perfect graph G. The algorithm needs to solve $n+1$ times the Lovász's SDP. Note that there is no known algorithm to compute a maximum stable set for perfect graphs without SDP. It is also possible to write an algorithm based on the Lovász's ϑ-function to compute a minimum coloring of perfect graphs.

2 Maxcut SDP

Let G be a simple graph with weights $w_{ij} \geq 0, \forall\, ij \in E$.

Definition 6 (cut)**.** A *cut* of G is a partition of V in two node sets S and $\bar{S}$. The weight of a cut is the sum of the weights of the cut edges:

$$\operatorname{cut}(S, \bar{S}) = \sum_{\substack{ij \in E \\ i \in S, j \notin S}} w_{ij}$$

The maximum cut problem asks to find the cut of maximum weight. From now on, we assume without loss of generality that G is the complete graph, since adding edges of weight 0 does not change the maximum cut.

We represent a cut by a vector $\boldsymbol{x} \in \{-1, 1\}^n$, where $x_i = 1$ if $i \in S$, and $x_i = -1$ if $i \in \bar{S}$. Then, we have

$$1 - x_i x_j = \begin{cases} 2 & \text{if } \{i, j\} \text{ is a cut-edge} \\ 0 & \text{otherwise.} \end{cases}$$

It follows that $\operatorname{cut}(S, \bar{S}) = \frac{1}{2} \sum_{ij \in E} w_{ij}(1 - x_i x_j) = \frac{1}{4} \sum_{1 \leq i,j \leq n} w_{ij}(1 - x_i x_j)$ (there is an additional $\frac{1}{2}$-factor because each edge is counted twice in the sum). So the maximum cut problem can be formulated as

$$\begin{aligned} \underset{\boldsymbol{x}}{\textbf{maximize}} \quad & \frac{1}{4} \sum_{i,j} w_{ij}(1 - x_i x_j) \\ & \boldsymbol{x} \in \{-1, 1\}^n. \end{aligned}$$

To formulate an SDP relaxation, we introduce a matrix variable X, and we would like that $X_{ij} = x_i x_j$. To this end, we use the following lemma

Lemma 5. *The matrix $X \in \mathbb{S}^n$ satisfies $X_{ij} = x_i x_j$ for some vector $\boldsymbol{x} \in \{-1, 1\}^n$ if and only if*

$$X \succeq 0, \quad \textbf{diag}\,(X) = \mathbf{1}, \quad \textit{and} \quad \textbf{rank}\,(X) = 1.$$

Proof. We know that positive semidefinite matrices of rank 1 are of the form $X = \boldsymbol{u}\boldsymbol{u}^T$ for some vector $\boldsymbol{u} \in \mathbb{R}^n$, that is, $X_{ij} = u_i u_j$. Moreover, we have $(\boldsymbol{u}\boldsymbol{u}^T)_{ii} = (u_i)^2$, so $X_{ii} = 1 \iff u_i \in \{-1, 1\}$. □

If X satisfies the property of this lemma, then it holds

$$\operatorname{cut}(S, \bar{S}) = \frac{1}{4} \sum_{i,j} w_{ij}(1 - x_i x_j) = \frac{1}{4} \langle W, J - X \rangle,$$

where W is the symmetric matrix such that both W_{ij} and W_{ji} are set to the weight w_{ij} of the edge $\{i, j\}$. If follows that the maximum cut problem is equivalent to the following optimization problem:

$$\begin{aligned} \underset{X \in \mathbb{S}^n}{\textbf{maximize}} \quad & \frac{1}{4} \langle W, J - X \rangle \\ & \textbf{diag}\,(X) = \mathbf{1} \\ & X \succeq 0 \\ & \textbf{rank}\,(X) = 1. \end{aligned}$$

The above problem *is not* an SDP, because of the nonconvex rank-one constraint. However, we otain an

SDP relaxation by removing that constraint:

$$\begin{aligned} \underset{X \in \mathbb{S}^n}{\textbf{maximize}} \quad & \frac{1}{4}\langle W, J - X\rangle \\ & \textbf{diag}\,(X) = \mathbf{1} \\ & X \succeq 0 \end{aligned} \tag{4}$$

Denote the optimal value of this relaxation by SDP. Since this is a relaxation (we removed a constraint, hence we optimize over a larger set of matrices), we have

$$\text{maxcut}(G) \leq \text{SDP}\,.$$

Note: In the exercises, we will give an alternative formulation for the MAXCUT SDP, which relies on the Laplacian matrix of the graph G, and can be used to derive analytic bounds on maxcut(G).

In a seminal paper, Goemans and Williamson showed that it is also possible to use the SDP (4) to derive an approximation algorithm for the maximum cut problem [5]. We next present this randomized algorithm, which relies on projections on a random hyperplane:

1. Compute a solution X^* of the SDP (4).
2. Compute a decomposition $X^* = H^T H$ (for example, a Cholesky decomposition). Denote the columns of H by $\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n$. Note that the constraint $X^*_{ii} = 1$ implies $\boldsymbol{h}_i^T \boldsymbol{h}_i = 1$. Hence, the $\boldsymbol{h}_i$'s have unit norm.
3. Draw a vector $\boldsymbol{r}$ uniformly at random over the unit sphere of $\mathbb{R}^n$. To do this, one can draw independently some $z_i \sim \mathcal{N}(0,1)$ for $i = 1, \ldots, n$, and then take $\boldsymbol{r} = \frac{1}{\|\boldsymbol{z}\|}\boldsymbol{z}$.
4. Finally, return the cut defined by $S = \{i \in V : \ \boldsymbol{r}^T \boldsymbol{h}_i > 0\}$. (Or, equivalently, define the cut through the vector $x_i = \text{sign}(\boldsymbol{r}^T \boldsymbol{h}_i)$, $\forall i \in [n]$.)

Theorem 6 (Goemans & Williamson)**.** *Let $(S, \bar{S})$ be the (random) cut returned by the above random projection algorithm. Then,*

$$\mathbb{E}[\text{cut}(S, \bar{S})] \geq \alpha \text{ SDP} \geq \alpha \text{ maxcut}(G),$$

where $\alpha \simeq 0.87856$.

The proof of this theorem is based on the following lemma:

Lemma 7. *Let $\boldsymbol{u}$ and $\boldsymbol{v}$ be two vectors on the unit sphere on $\mathbb{R}^n$, and let $\boldsymbol{r}$ be a random vector drawn uniformly at random on the sphere. Denote by H be the hyperplane $\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x}^T \boldsymbol{r} = 0\}$. Then, the probability that H separates $\boldsymbol{u}$ and $\boldsymbol{v}$ is equal to $\frac{\theta}{\pi}$, where $\theta = \arccos(\boldsymbol{u}^T \boldsymbol{v})$ is the angle between $\boldsymbol{u}$ and $\boldsymbol{v}$.*

Proof. (Sketch) We can reason in the two-dimensional subspace which contains $\boldsymbol{u}$ and $\boldsymbol{v}$, and for a suitable basis $(\boldsymbol{e}_1, \boldsymbol{e}_2)$ of this subspace, it holds $\boldsymbol{u} = \boldsymbol{e}_1$ and $\boldsymbol{v} = \cos(\theta)\boldsymbol{e}_1 + \sin(\theta)\boldsymbol{e}_2$. Then, it is easy to see that the projection of $\boldsymbol{r}$ on this subspace is a vector of the form $\boldsymbol{r} = \rho\big(\cos(\alpha)\boldsymbol{e}_1 + \sin(\alpha)\boldsymbol{e}_2\big)$, where $\rho \leq 1$ and α is drawn unifomly at random in $[0, 2\pi]$. Finally, the hyperplane defined by $\boldsymbol{r}$ separates $\boldsymbol{u}$ and $\boldsymbol{v}$ iff $\boldsymbol{r}$ lies in a two-sided cone of angle θ, which occurs with probability $\frac{2\theta}{2\pi}$. □

We are now ready to prove the theorem:

Proof. The expected weight of the cut $(S, \bar{S})$ is

$$\begin{aligned}\mathbb{E}[\mathrm{cut}(S,\bar{S})] &= \sum_{ij\in E} w_{ij}\mathbb{P}[\{i,j\} \text{ belongs to the cut set}]\\ &= \sum_{ij\in E} w_{ij}\frac{\arccos(\boldsymbol{h_i}^T\boldsymbol{h}_j)}{\pi}\end{aligned}$$

Now we multiply and divide the (i,j)th term of this sum by $\frac{1}{2}(1-\boldsymbol{h_i}^T\boldsymbol{h}_j) = \frac{1}{2}(1-X^*_{ij})$:

$$\mathbb{E}[\mathrm{cut}(S,\bar{S})] = \sum_{ij\in E}\frac{1}{2}w_{ij}(1-X^*_{ij})\frac{2\arccos(\boldsymbol{h_i}^T\boldsymbol{h}_j)}{\pi(1-\boldsymbol{h_i}^T\boldsymbol{h}_j)}$$

A straigtforward analysis shows that $\alpha := \inf_{\theta\in[0,\pi]}\frac{2}{\pi}\frac{\theta}{1-\cos(\theta)} \simeq 0.87856$. Hence,

$$\mathbb{E}[\mathrm{cut}(S,\bar{S})] \geq \alpha\sum_{ij\in E}\frac{1}{2}w_{ij}(1-X^*_{ij}) = \alpha\sum_{1\leq i,j\leq n}\frac{1}{4}w_{ij}(1-X^*_{ij}) = \alpha\,\frac{1}{4}\langle W, J-X^*\rangle = \alpha\ \mathrm{SDP}\,.$$

□

3 Further extensions of the MAXCUT result

3.1 Nesterov's $\frac{2}{\pi}$-approximation

There exists an alternative formulation of the MAXCUT SDP, in which the objective function is the scalar product between X and a positive semidefinite matrix. To see this, note that the equality $X_{ii} = 1$ holds for all i, so the objective function of (4) rewrites

$$\frac{1}{2}\sum_{i<j}w_{ij}(1-X_{ij}) = \frac{1}{4}\sum_{i<j}w_{ij}(X_{ii}+X_{jj}-2X_{ij}) = \frac{1}{4}\sum_{i<j}w_{ij}(\boldsymbol{e}_i-\boldsymbol{e}_j)^TX(\boldsymbol{e}_i-\boldsymbol{e}_j) = \frac{1}{4}\langle X, L\rangle,$$

where $L := \sum_{i<j}w_{ij}(\boldsymbol{e}_i-\boldsymbol{e}_j)(\boldsymbol{e}_i-\boldsymbol{e}_j)^T$ is the *Laplacian matrix* of the graph G. By construction, L is positive semidefinite, because it is a conic combination of rank-one positive semidefinite matrices.

Nesterov [8] generalized the result of Goemans and Williamson to the case where the matrix $Q \succeq 0$ is *any* positive semidefinite matrix (not necessarily the Laplacian matrix of some graph). This means that we are interested in solving the following binary quadratic program

$$\operatorname*{\mathbf{maximize}}_{\boldsymbol{x}\in\{-1,1\}^n}\quad \boldsymbol{x}^TQ\boldsymbol{x}, \tag{5}$$

which we relax to the SDP

$$\begin{aligned}\operatorname*{\mathbf{maximize}}_{X\in\mathbb{S}^n}\quad & \langle Q, X\rangle\\ s.t.\quad & \mathbf{diag}\,(X) = 1\\ & X\succeq 0.\end{aligned}$$

The proposed rounding scheme is the same as for MAXCUT: We solve the SDP, and get a decomposition of the optimal matrix X in order to get unit vectors $\boldsymbol{u}_i$ with $\boldsymbol{u}_i^T\boldsymbol{u}_j = X_{ij}$. Then, we draw a Gaussian vector $\boldsymbol{r}$ at random and define $x_i = \mathrm{sign}(\boldsymbol{r}^T\boldsymbol{u}_i)$. Obviously, the geometric lemma 7 still holds for this problem, so $\mathbb{P}[x_i \neq x_j] = \frac{1}{\pi}\arccos(X_{ij})$. From there, we get

$$\mathbb{E}[x_ix_j] = 1\cdot\underbrace{\mathbb{P}[x_i = x_j]}_{1-\arccos(X_{ij})/\pi} + (-1)\cdot\underbrace{\mathbb{P}[x_i\neq x_j]}_{\arccos(X_{ij})/\pi} = 1-\frac{2}{\pi}\arccos(X_{ij}) = \frac{2}{\pi}\arcsin(X_{ij}).$$

Therefore, the expected value of the rounding can be expressed by

$$\mathbb{E}[\boldsymbol{x}^T Q \boldsymbol{x}] = \langle Q, \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T]\rangle = \frac{2}{\pi}\langle Q, \arcsin(X)\rangle,$$

where the arcsin operator is applied elementwise to the matrix X. Now, we need the following results:

Proposition 8 (Schur Product Theorem)**.** *Let $X \succeq 0$ and $Y \succeq 0$. Then, $X \circ Y \succeq 0$, where $X \circ Y \in \mathbb{S}^n$ denotes the Hadamard (elementwise) product of X and Y, i.e., $(X \circ Y)_{ij} = X_{ij} Y_{ij}$.*

Proof. Consider eigendecompositions $X = \sum_i \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^T$ and $T = \sum_i \mu_i \boldsymbol{v}_i \boldsymbol{v}_i^T$. Then,

$$X \circ Y = \sum_i \sum_j \lambda_i \mu_j (\boldsymbol{u}_i \boldsymbol{u}_i^T) \circ (\boldsymbol{v}_j \boldsymbol{v}_j^T).$$

Now, we observe that the identity $(\boldsymbol{u}\boldsymbol{u}^T) \circ (\boldsymbol{v}\boldsymbol{v}^T) = (\boldsymbol{u} \circ \boldsymbol{v})(\boldsymbol{u} \circ \boldsymbol{v})^T$ holds for all vectors $\boldsymbol{u}, \boldsymbol{v}$ (both matrices have coordinates $u_i u_j v_i v_j$). So, $X \circ Y = \sum_i \sum_j \lambda_i \mu_j (\boldsymbol{u}_i \boldsymbol{v}_j^T) \circ (\boldsymbol{u}_i \boldsymbol{v}_j^T)$ is a conic combination of rank-one positive semidefinite matrices, which implies $X \circ Y \succeq 0$. ☐

Lemma 9. $\arcsin(X) \succeq X$.

Proof. The function arcsin admits the following Taylor series:

$$\arcsin(x) = x + \frac{1}{2}\frac{x^3}{3} + \frac{1 \cdot 3}{2 \cdot 4}\frac{x^5}{5} + \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6}\frac{x^7}{7} + \ldots, \qquad \forall |x| \leq 1.$$

Now, we observe that $|X_{ij}| \leq 1$ for all i, j, because we know that the diagonal elements of X are $X_{ii} = 1$, and $X \succeq 0$ implies $|X_{ij}| \leq \sqrt{X_{ii} X_{jj}} = 1$. So we can apply the above series componentwise, which gives

$$\arcsin(X) = X + \underbrace{\frac{1}{2}\frac{X^{\circ 3}}{3}}_{\succeq 0} + \underbrace{\frac{1 \cdot 3}{2 \cdot 4}\frac{X^{\circ 5}}{5}}_{\succeq 0} + \underbrace{\frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6}\frac{X^{\circ 7}}{7}}_{\succeq 0} + \ldots \succeq X,$$

where the conic inequality follows from the Schur product theorem (Proposition 8). ☐

We are now ready to prove Nesterov's result:

Theorem 10 (Nesterov)**.**

$$\mathbb{E}[\boldsymbol{x}^T Q \boldsymbol{x}] \geq \frac{2}{\pi} SDP \geq \frac{2}{\pi} OPT.$$

Proof. We know that $\arcsin(X) \succeq X$ from Lemma 9 and that $Q \succeq 0$, so $\langle Q, \arcsin(X) - X\rangle \geq 0$. Therefore,

$$\mathbb{E}[\boldsymbol{x}^T Q \boldsymbol{x}] = \frac{2}{\pi}\langle Q, \arcsin(X)\rangle \geq \frac{2}{\pi}\langle Q, X\rangle = \frac{2}{\pi} SDP.$$

Finally, $SDP \geq OPT$ holds because SDP is a relaxation of the binary quadratic program (5). ☐

3.2 A 0.651-approximation algorithm for MAX-BISECTION

The problem of MAX-BISECTION is simular to MAXCUT, except that the returned cut is constrained to partition the vertices in two sets of equal size. So, we assume that $|V| = n$ is even, and we want to maximize $\mathrm{cut}(S, \bar{S})$ subject to the constraint that $|S| = \frac{n}{2}$.

In this section, we present a 0.651-approximation algorithm due to Frieze and Jerrum [4], but we point out that this result has been improved by several authors. The best approximation factor known to date for this problem is very close to the 0.878 performance guarantee of MAXCUT [1], but it requires solving several rounds of the Lasserre hierarchy of SDPs (cf. next chapter), so its running time –although polynomial– is very bad: the authors claim estimates in the order of $O(n^{10^{100}})$... In contrast, the algorithm we present involves solving a simple SDP.

In order to refine the SDP (4), we observe that if the vector $\boldsymbol{x} \in \{-1,1\}^n$ defines a bisection, then $\mathbf{1}^T\boldsymbol{x} = 0$. Squaring this equality yields $\mathbf{1}^T\boldsymbol{x}\boldsymbol{x}^T\mathbf{1} = 0$, so the equality $\mathbf{1}^T X^*\mathbf{1} = \langle J, X^*\rangle = 0$ must be satisfied by the matrix $X^* = \boldsymbol{x}^*\boldsymbol{x}^{*T}$, where $\boldsymbol{x}^*$ is the $\{\pm 1\}$-indicator vector of the maximal bisection cut. This shows that the following SDP is a relaxation of MAX-BISECTION:

$$\begin{aligned} \underset{X\in\mathbb{S}^n}{\textbf{maximize}} \quad & \frac{1}{4}\langle W, J-X\rangle \\ & \textbf{diag}\,(X) = \mathbf{1} \\ & \langle J, X\rangle = 0 \\ & X \succeq 0 \end{aligned} \tag{6}$$

The rounding algorithm proposed by Frieze & Jerrum works as follows: As in the algorithm of Goemans and Williamson, solve the SDP and factorize the optimal matrix X, in order to obtain some unit-length vectors $\boldsymbol{u}_i$ satisfying $\boldsymbol{u}_i^T\boldsymbol{u}_j = X_{ij}$.

Then, for a vector $\boldsymbol{p} \in \mathbb{R}^n$, we define a bisection $(S(\boldsymbol{p}), \overline{S(\boldsymbol{p})})$ as follows: let $S_1 = \{i : \boldsymbol{u}_i^T\boldsymbol{p} \geq 0\}, S_2 = \{i : \boldsymbol{u}_i^T\boldsymbol{p} < 0\}$, and assume w.l.o.g that $|S_1| \geq |S_2|$ (otherwise, swap S_1 and S_2). Finally, sort the elements of S_1 as $i_1, \ldots, i_\ell$, so that $\zeta_{i_1} \geq \zeta_{i_2} \geq \ldots \geq \zeta_{i_\ell}$, where $\ell = |S_1| \geq \frac{n}{2}$ and $\zeta_i = \sum_{j\in S_2} w_{ij}$, and define $S(\boldsymbol{p}) = \{i_1, \ldots, i_{\frac{n}{2}}\}$. In simple words, the last step greedily reassigns points from the larger set S_1 to the smaller set S_2, until both sets have the same size.

Theorem 11 (Frieze & Jerrum). *Generate independent random Gaussian vectors $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_K \sim \mathcal{N}(0, I)$, and output the best bisection cut S from $S(\boldsymbol{p}_1), \ldots, S(\boldsymbol{p}_K)$. If $K \geq \frac{2}{\epsilon}\log(\frac{2}{\epsilon})$ for some $\epsilon > 0$, then*

$$\mathbb{E}[\mathrm{cut}(S, \bar{S})] \geq \big(\underbrace{2(\sqrt{2\alpha}-1)}_{>0.6511} - \epsilon\big)OPT,$$

where $\alpha \simeq 0.87856$ is the approximation factor of Theorem 6 for MAXCUT, and OPT denotes the optimal value of the maximum bisection problem.

Proof (sketch). Let $\boldsymbol{p} \sim \mathcal{N}(0, I)$ and consider the sets S_1 and S_2, as described above. We introduce the random variables $C = \mathrm{cut(S_1, S_2)}$ and $Y = |S_1|\cdot|S_2|$. The first thing to observe is that the analysis of Goemans and Williamson still holds, so

$$\mathbb{E}[C] \geq \alpha\, SDP \geq \alpha\, OPT.$$

Now, to get a bound on $\mathbb{E}[Y]$, we will use the fact that $\langle J, X\rangle = 0$ (this is a constraint of the refined SDP):

$$\begin{aligned} \mathbb{E}[Y] &= \sum_{i<j} \mathbb{P}[(i,j) \text{ are separated by cut } (S_1, S_2)] \\ &= \sum_{i<j} \frac{\arccos(X_{ij})}{\pi} && \text{(By Lemma 7)} \\ &\geq \sum_{i<j} \frac{\alpha}{2}(1 - X_{ij}) && \text{(By definition of } \alpha) \\ &= \frac{\alpha}{2}\left(\sum_{i<j} 1 - \frac{1}{2}(\underbrace{\langle J, X\rangle}_{=0} - \underbrace{\mathrm{trace}\, X}_{=n})\right) \\ &= \frac{\alpha}{2}\left(\frac{n(n-1)}{2} + \frac{n}{2}\right) = \alpha\frac{n^2}{4}. \end{aligned}$$

So $\mathbb{E}[Y]$ is at least within a fraction α of $y^* = \frac{n^2}{4}$, where y^* is the largest value Y can take (when $|S_1| = |S_2| = \frac{n}{2}$).

Then, we define the random variable $Z = \frac{C}{OPT} + \frac{Y}{y^*}$. The above inequalities show that $\mathbb{E}[Z] \geq 2\alpha$, while Z is a random variable with bounded support (we have $Z \in [0,3]$, because $Y \leq y^*$ and it can be seen that $C \leq 2OPT$). Therefore, if we draw enough samples $z_1, \ldots, z_K$ from the random variable Z, we will soon get one $z_i \geq \mathbb{E}[Z]$.

The precise analysis of Frieze and Jerrum shows that if $K \geq \epsilon^{-1}\log\epsilon^{-1}$, then $\max(z_1,\ldots,z_K) \geq (1-\epsilon)\mathbb{E}[Z]$ with probability $\geq 1-\epsilon$. For the sake of simplicity, let us forget about the ϵ, and assume that after K random draws, we obtained two sets S_1 and S_2 such that

$$\frac{\operatorname{cut}(S_1,S_2)}{OPT} + \frac{|S_1|\cdot|S_2|}{y^*} \geq \mathbb{E}[Z] \geq 2\alpha. \tag{7}$$

Now, consider the bisection cut $S = \{i_1,\ldots,i_{n/2}\}$ returned by the greedy swapping procedure applied to S_1 and S_2. By construction,

$$\operatorname{cut}(S,\bar{S}) = \sum_{i\in S}\sum_{j\in\bar{S}} w_{ij} \geq \sum_{i\in S}\sum_{j\in S_2} w_{ij} = \sum_{i\in S}\zeta_i \geq \frac{n/2}{|S_1|}\sum_{i\in S_1}\zeta_i = \frac{n/2}{|S_1|}\operatorname{cut}(S_1,S_2), \tag{8}$$

where the first inequality follows from $S_2 \subseteq \bar{S}$ and the second inequality from the fact that S retains the $n/2$ elements of S_1 with the largest ζ_i's.

Assume that $\operatorname{cut}(S_1,S_2) = \lambda\cdot OPT$ and $|S_1| = \delta n$, so that inequality (7) rewrites

$$\lambda + \frac{\delta n(n-\delta n)}{n^2/4} = \lambda + 4\delta(1-\delta) \geq 2\alpha \iff \lambda \geq 2\alpha - 4\delta(1-\delta). \tag{9}$$

Combining (8) and (9), we get

$$\operatorname{cut}(S,\bar{S}) \geq \frac{n/2}{\delta n}\ \lambda\cdot OPT \geq \frac{1}{2\delta}\big(2\alpha - 4\delta(1-\delta)\big)\cdot OPT.$$

Finally, it is easy to see that the function $\delta \mapsto \frac{1}{2\delta}\big(2\alpha-4\delta(1-\delta)\big) = \frac{\alpha}{\delta}+2\delta-2$ is minimized for $\delta = \sqrt{\frac{\alpha}{2}} \in [0,1]$, and substituting this value in the above inequality yields

$$\operatorname{cut}(S,\bar{S}) \geq 2(\sqrt{2\alpha}-1)OPT.$$

Clearly, the best bisection cut from $S(\boldsymbol{p}_1),\ldots,S(\boldsymbol{p}_K)$ is at least as good as the cut obtained from the sets (S_1,S_2) for which (7) holds. A more precise analysis with the ϵ yields the precise statement from the theorem. □

3.3 MAX-3-CUT via complex SDP

Goemans and Williamson proposed an elegant extension of their result to the case of MAX-3-CUT [6], by using *complex semidefinite programming*, which is conic programming over the cone $\mathcal{H}_+^n \subset \mathbb{C}^{n\times n}$ of positive semidefinite hermitian matrices. The problem MAX-3-CUT asks to partition the vertices of G in 3 subsets S_0, S_1, S_2, so that the weight of all edges with endpoints in two different subsets is maximized.

For a vertex $v \in V$, let $x_v \in \{0,1,2\}$ denote the index of the partition to which v is assigned, and denote $z_v = \omega^{x_v}$, where $\omega = e^{2\boldsymbol{i}\frac{\pi}{3}} = -\frac{1}{2} + \boldsymbol{i}\frac{\sqrt{3}}{2} \in \mathbb{C}$ is the principal third complex root of unity. For all $i,j \in V$, note that $z_i\bar{z}_j + z_j\bar{z}_i = 2$ if $x_i = x_j$, and $z_i\bar{z}_j + z_j\bar{z}_i = -1$ otherwise. Hence, the weight of the 3−cut defined by $\boldsymbol{x}$ can be expressed as

$$\operatorname{cut}(S_1,S_2,S_3) = \sum_{i<j} w_{ij}\mathbf{1}_{\{x_i\neq x_j\}} = \sum_{i<j}\frac{w_{ij}}{3}(2 - z_i\bar{z}_j - z_j\bar{z}_i).$$

So the problem of maximizing the above expression, subject to the constraints that $z_i \in R_3 := \{1,\omega,\omega^2\}, \forall i$, is equivalent to MAX-3-CUT. To obtain an SDP relaxation, we first introduce the Hermitian matrix $Z \in \mathcal{H}_n$ such that $Z_{ij} = z_i\bar{z}_j$. Note that $Z = \boldsymbol{z}\boldsymbol{z}^*$, so Z is positive semidefinite and has rank one (recall that $\boldsymbol{z}^* := \bar{\boldsymbol{z}}^T$ denotes the *conjugate transpose of* $\boldsymbol{z}$). The diagonal elements of Z are $Z_{ii} = z_i\bar{z}_i = |z_i|^2 = 1$, and the off-diagonal elements are $Z_{ij} = R^3$. We obtain a $\mathcal{H}_+^n$-cone programming relaxation by ignoring the rank-one constraint, and by constraining Z_{ij} in $\mathbf{conv}(R^3)$, that is, the triangle in the complex plane whose vertices are the third roots of unity:

$$\begin{aligned}
\underset{Z\in\mathcal{H}^n}{\textbf{maximize}} \quad & \sum_{i<j}\frac{w_{ij}}{3}(2 - Z_{ij} - Z_{ji}) \\
s.t. \quad & Z_{ii} = 1, \qquad \forall i \in V \\
& \alpha Z_{ij} + \bar{\alpha}Z_{ji} \geq -1, \quad \forall \alpha \in R^3,\ \forall i<j \\
& Z \succeq_{\mathcal{H}_+^n} 0.
\end{aligned}$$

In the above program, it can be seen that the three constraints involving Z_{ij} (for each $\alpha \in R^3$) are equivalent to $Z_{ij} \in \mathbf{conv}(R^3)$.

The first thing you might ask yourself is: can we solve such a cone program over $\mathcal{H}_+^n$ efficiently? In fact, it is possible to reformulate such problems as an equivalent real-valued SDP. To see this, the trick is to introduce the matrix

$$M = \begin{bmatrix} X & -Y \\ Y & X \end{bmatrix},$$

where $Z = X + \boldsymbol{i}Y$ is the decomposition of Z into real and imaginary parts. Clearly, M must be symmetric (because X is symmetric and Y is skew-symetric, so $Y^T = -Y$), and it can be seen that for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, with $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{i} \cdot \boldsymbol{y} \in \mathbb{C}^n$, it holds $\boldsymbol{z}^* Z \boldsymbol{z} = \boldsymbol{x}^T X \boldsymbol{x} + 2\boldsymbol{y}^T Y \boldsymbol{x} + \boldsymbol{y}^T X \boldsymbol{y} = [\boldsymbol{x}^T, \boldsymbol{y}^T] M [\boldsymbol{x}^T, \boldsymbol{y}^T]^T$. Therefore,

$$Z \succeq_{\mathcal{H}_+^n} 0 \iff \boldsymbol{z}^* Z \boldsymbol{z} \geq 0, \quad \forall \boldsymbol{z} \in \mathbb{C}^n \iff \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix}^T M \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} \geq 0, \quad \forall \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} \in \mathbb{R}^{2n} \iff M \succeq 0.$$

This implies that the above complex-SDP can be recast as the following equivalent SDP with real variables:

$$\begin{aligned} \underset{X \in \mathbb{S}^n, Y \in \mathbb{R}^{n \times n}}{\textbf{maximize}} \quad & \sum_{i<j} \frac{w_{ij}}{3}(2 - 2X_{ij}) \\ s.t. \quad & X_{ii} = 1, \qquad \forall i \in V \\ & 2\,\mathrm{Re}(\alpha) X_{ij} - 2\,\mathrm{Im}(\alpha) Y_{ij} \geq -1, \quad \forall \alpha \in R^3,\ \forall i < j \\ & \begin{bmatrix} X & -Y \\ Y & X \end{bmatrix} \succeq 0. \end{aligned}$$

So, the way to adapt the MAX-CUT rounding algorithm to the case of MAX-3-CUT is to solve the above complex-SDP, and to get a decomposition $Z = UU^*$ of the optimal matrix. By construction, the rows $\boldsymbol{u}_i^T$ of U are such that $Z_{ij} = \langle \boldsymbol{u}_i, \boldsymbol{u}_j \rangle$, so we have $\|\boldsymbol{u}_i\| := \sqrt{\boldsymbol{u}_i^* \boldsymbol{u}_i} = (Z_{ii})^{1/2} = 1$. Then, we generate a random vector $\boldsymbol{p} \in \mathbb{C}^n$ with independent complex Gaussian coordinates (i.e., $\boldsymbol{p} = \boldsymbol{p}_1 + \boldsymbol{i} \cdot \boldsymbol{p}_2$ for some independent Gaussian vectors $\boldsymbol{p}_1 \sim \mathcal{N}(0, I)$ and $\boldsymbol{p}_2 \sim \mathcal{N}(0, I)$). The randomized rounding is then obtained from the argument of the complex scalar products $\langle \boldsymbol{p}, \boldsymbol{u}_i \rangle = \boldsymbol{u}_i^* \boldsymbol{p} \in \mathbb{C}$:

$$x_i = \begin{cases} 0 \text{ if } \arg\langle \boldsymbol{p}, \boldsymbol{u}_i \rangle \in [0, 2\pi/3) \\ 1 \text{ if } \arg\langle \boldsymbol{p}, \boldsymbol{u}_i \rangle \in [2\pi/3, 4\pi/3) \\ 2 \text{ if } \arg\langle \boldsymbol{p}, \boldsymbol{u}_i \rangle \in [4\pi/3, 2\pi) \end{cases}.$$

Goemans and Williamson proved the following two geometric lemmas, which imply a 0.836-performance guarantee for the above algorithm:

> **Lemma 12.** *If $\langle \boldsymbol{u}_i, \boldsymbol{u}_j \rangle = re^{\boldsymbol{i}\theta}$, then the probability that $x_i \neq x_j$ is*
>
> $$P(r, \theta) = \frac{2}{3} - \frac{3}{8\pi^2}\Big[2 \arccos^2\big(-r\cos(\theta)\big) - \arccos^2\big(-r\cos(\theta + \frac{2\pi}{3})\big) - \arccos^2\big(-r\cos(\theta - \frac{2\pi}{3})\big)\Big].$$

> **Lemma 13.** *If $\boldsymbol{z} = re^{\boldsymbol{i}\theta}$, belongs to the triangle defined by R_3, then $P(r, \theta) \geq \psi \cdot \frac{2}{3}(1 - \mathrm{Re}(\boldsymbol{z}))$, where $\psi := \frac{7}{12} + \frac{3}{\pi^2} \arccos^2(-\frac{1}{4}) > 0.836008$.*

It is now easy to show that the expected value of the 3-cut S_0, S_1, S_2 is within a fraction ψ of the optimum cut:

> **Theorem 14.** *Let (S_0, S_1, S_2) be the (random) 3-cut returned by the above rounding algorithm. Then,*
>
> $$\mathbb{E}[\mathrm{cut}(S_0, S_1, S_2)] \geq \psi \text{ SDP} \geq \psi \text{ max-3-cut}(G).$$

Proof. Denote by $\boldsymbol{u}_i$ the complex vectors returned by the algorithm, and let $\langle \boldsymbol{u}_i, \boldsymbol{u}_j \rangle = r_{ij} e^{\boldsymbol{i}\theta_{ij}}$.

$$
\begin{aligned}
\mathbb{E}[\mathrm{cut}(S_0, S_1, S_2)] &= \mathbb{E}\Big[\sum_{i<j} w_{ij} \mathbf{1}_{\{x_i \neq x_j\}}\Big] \\
&= \sum_{i<j} w_{ij} \mathbb{P}[\boldsymbol{x}_i \neq \boldsymbol{x}_j] \\
&= \sum_{i<j} w_{ij} P(r_{ij}, \theta_{ij}) && \text{(by Lemma 12)} \\
&\geq \psi \sum_{i<j} w_{ij} \frac{2}{3}\Big(1 - \mathrm{Re}(\langle \boldsymbol{u}_i, \boldsymbol{u}_j \rangle)\Big) && \text{(by Lemma 13)} \\
&= \psi \sum_{i<j} \frac{w_{ij}}{3}\Big(2 - Z_{ij} - Z_{ji}\Big) = \psi \;\; \mathrm{SDP}\,.
\end{aligned}
$$

□

The last inequality, ψ SDP $\geq \psi$ max-3-cut(G) follows from the fact that the complex SDP is a relaxation of the MAX-3-CUT problem.

4 SDP relaxations for nonconvex QCQPs (with binary variables)

In the previous sections, we have derived SDP relaxations for the stable set problem and (variants of) the maximum cut problem. In fact, those relaxations could have been obtained in a systemmatic manner, by using a *general recipe* that allows to obtain an SDP relaxation for any QCQP.

Let us consider an optimization problem of the form

$$
\begin{aligned}
\underset{\boldsymbol{x}\in\mathbb{R}^n}{\textbf{minimize}} \quad & \boldsymbol{x}^T Q_0 \boldsymbol{x} + \boldsymbol{c}_0^T \boldsymbol{x} + q_0 \\
s.t. \quad & \boldsymbol{x}^T Q_i \boldsymbol{x} + \boldsymbol{c}_i^T \boldsymbol{x} + q_i \lesseqgtr 0, \quad \forall i \in [m],
\end{aligned} \tag{10}
$$

where the data $(Q_i, \boldsymbol{c}_i, q_i)_i$ is of appropriate size, and the symbol $\lesseqgtr$ replaces any of $\{\leq, =, \geq\}$. Note that this problem is not convex in general. In particular, it allows for binary constraints of the form $x_i \in \{0, 1\}$, which can be formulated as equivalent quadratic equality constraints:

$$
x_i \in \{0,1\} \iff x_i^2 = x_i \iff \boldsymbol{x}^T Q \boldsymbol{x} + \boldsymbol{c}^T \boldsymbol{x} = 0 \text{ for } Q = \boldsymbol{e}_i \boldsymbol{e}_i^T, \boldsymbol{c} = -\boldsymbol{e}_i.
$$

We can use semidefinite programming to construct a relaxation of (10). To do this, one possibility is to introduce an auxilary variable $X = \boldsymbol{x}\boldsymbol{x}^T$: Prolem (10) is equivalent to

$$
\begin{aligned}
\underset{\boldsymbol{x}\in\mathbb{R}^n, X\in\mathbb{S}^n}{\textbf{minimize}} \quad & \langle Q_0, X \rangle + \boldsymbol{c}_0^T \boldsymbol{x} + q_0 \\
s.t. \quad & \langle Q_i, X \rangle + \boldsymbol{c}_i^T \boldsymbol{x} + q_i \lesseqgtr 0, \quad \forall i \in [m], \\
& X = \boldsymbol{x}\boldsymbol{x}^T.
\end{aligned} \tag{11}
$$

this problem is not convex because of the constraint $X = \boldsymbol{x}\boldsymbol{x}^T$. However, we obtain an SDP if we relax this constraint to $X \succeq \boldsymbol{x}\boldsymbol{x}^T$, which can be expressed as an LMI by using a Schur complement:

$$
X \succeq \boldsymbol{x}\boldsymbol{x}^T \iff \begin{bmatrix} X & \boldsymbol{x} \\ \boldsymbol{x}^T & 1 \end{bmatrix} \succeq 0.
$$

Therefore, we obtain the following result:

Proposition 15. *The SDP*

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n, X\in\mathbb{S}^n}{\mathbf{minimize}} \quad & \langle Q_0, X\rangle + \boldsymbol{c}_0^T\boldsymbol{x} + q_0 \\ s.t. \quad & \langle Q_i, X\rangle + \boldsymbol{c}_i^T\boldsymbol{x} + q_i \lesseqgtr 0, \quad \forall i\in[m], \\ & \begin{bmatrix} X & \boldsymbol{x} \\ \boldsymbol{x}^T & 1 \end{bmatrix} \succeq 0 \end{aligned} \tag{12}$$

is a relaxation of Problem (10)*. Its optimal value gives a lower bound for the original nonconvex QCQP.*

We observe that binary variables $x_i \in \{0,1\} \iff x_i^2 = x_i$ result in constraints of the form

$$X_{ii} = x_i$$

in the SDP. Similarly, a binary variable $x_j \in \{-1,1\} \iff x_j^2 = 1$ yields the constraint $X_{jj} = 1$.

There is an alternative way to interprete this SDP, by proceeding as we did for MAXCUT. Indeed, note that we obtain an exact reformulation of Problem (10) when we add the nonconvex constraint $\mathbf{rank} \begin{bmatrix} X & \boldsymbol{x} \\ \boldsymbol{x}^T & 1 \end{bmatrix} = 1$ to the SDP. To see this,

$$\begin{aligned} \begin{bmatrix} X & \boldsymbol{x} \\ \boldsymbol{x}^T & 1 \end{bmatrix} \succeq 0 \text{ is of rank 1} \iff & \begin{bmatrix} X & \boldsymbol{x} \\ \boldsymbol{x}^T & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{u} \\ \alpha \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \alpha \end{bmatrix}^T \text{ for some } \boldsymbol{u}\in\mathbb{R}^n, \alpha\in\mathbb{R} \\ \iff & X = \boldsymbol{u}\boldsymbol{u}^T \text{ for } \boldsymbol{u} = \alpha\boldsymbol{x}, \quad \alpha = \pm 1 \\ \iff & X = \boldsymbol{x}\boldsymbol{x}^T. \end{aligned}$$

5 Completely positive formulation for binary QPs

It was shown by Burer that we can even obtain an exact conic reformulation for QPs with binary variables. To do this, we need to introduce the cone of *copositive matrices*:

Definition 7 (Copositive cone)**.** The cone of $n\times n$ copositive matrices is

$$\mathcal{C}_n := \{X\in\mathbb{S}^n \mid \boldsymbol{u}^T X\boldsymbol{u} \geq 0,\ \forall \boldsymbol{u}\in\mathbb{R}^n_+\}.$$

Note that the copositive cone only differs from the semidefinite cone from the restriction $\boldsymbol{u} \geq \mathbf{0}$. We also introduce the cone of *completely positive matrices*:

Definition 8 (Completely positive cone)**.** The cone of $n\times n$ completely positive matrices is

$$\mathcal{C}_n^* := \{\sum_{k=1}^q \boldsymbol{u}_k\boldsymbol{u}_k^T \mid q\in\mathbb{N},\ \boldsymbol{u}_k\in\mathbb{R}^n_+,\ \forall k\in[q]\}.$$

The next proposition gives important properties above these 2 cones:

Proposition 16 (Properties of $\mathcal{C}_n$ and $\mathcal{C}_n^*$)**.**

(i) We require at most $q \leq n(n+1)/2$ vectors to decompose a completely positive matrix:

$$X\in\mathcal{C}_n^* \iff \exists \ell \leq \frac{1}{2}n(n+1), \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_\ell \in\mathbb{R}^n_+ : X = \sum_{j=1}^{\ell} \boldsymbol{x}_i\boldsymbol{x}_i^T.$$

(ii) The following inclusions hold:

$$\mathcal{C}_n^* \subseteq \left(\mathbb{S}_+^n \cap \mathbb{R}_+^{n\times n}\right) \subseteq \mathbb{S}_+^n \subseteq \left(\mathbb{S}_+^n + \mathbb{R}_+^{n\times n}\right) \subseteq \mathcal{C}_n.$$

(iii) The cones $\mathcal{C}_n$ and $\mathcal{C}_n^*$ are proper, and dual from each other.

Proof. (i). We have $\mathcal{C}_n^* = \mathbf{cone}\,\{\boldsymbol{u}\boldsymbol{u}^T | \boldsymbol{u} \in \mathbb{R}_n^+\}$. The affine dimension of $\{\boldsymbol{u}\boldsymbol{u}^T | \boldsymbol{u} \in \mathbb{R}_n^+\} \subseteq \mathbb{S}^n$ is less than $n(n+1)/2$ (the affine dimension of $\mathbb{S}^n$), so Caratheodory's theorem tells us that every element of $\mathcal{C}_n^*$ can be expressed as a conic combination of $q \leq n(n+1)/2$ elements of $\{\boldsymbol{u}\boldsymbol{u}^T | \boldsymbol{u} \in \mathbb{R}_n^+\}$.

(ii). The two inclusions in the middle are trivial, so we only prove the first and the last inclusions. Let $X \in \mathcal{C}_n^*$. Then X is positive semidefinite, because it can be written as a sum of rank-one positive semidefinite matrices. Moreover, the elements of $X = \sum_k \boldsymbol{u}_k \boldsymbol{u}_k^T$, where $\forall k\, \boldsymbol{u}_k \geq \mathbf{0}$, are clearly nonnegative. This shows: $\mathcal{C}_n^* \subseteq \mathbb{S}_+^n \cap \mathbb{R}_+^{n\times n}$. Now, let $X \in \left(\mathbb{S}_+^n + \mathbb{R}_+^{n\times n}\right)$, that is, $X = Y + Z$ for some matrices $Y \succeq 0$ and $Z \geq 0$. Then, for all $\boldsymbol{u} \geq \mathbf{0}$, it holds $\boldsymbol{u}^T X \boldsymbol{u} = \boldsymbol{u}^T Y \boldsymbol{u} + \boldsymbol{u}^T Z \boldsymbol{u}$, and $\boldsymbol{u}^T Y \boldsymbol{u} \geq 0$ because $Y \succeq 0$, $\boldsymbol{u}^T Z \boldsymbol{u} \geq 0$ because it is a sum of products of nonnegative numbers. This shows $\left(\mathbb{S}_+^n + \mathbb{R}_+^{n\times n}\right) \subseteq \mathcal{C}_n$.

(iii). The cone $\mathcal{C}_n$ is clearly closed, and convex (it is the intersection of infinitely many halfspaces). Its interior is nonempty, which can be seen from $\mathbb{S}_+^n \subseteq \mathcal{C}_n \implies \mathbf{int}\,\mathbb{S}_+^n = \mathbb{S}_{++}^n \subseteq \mathbf{int}\,\mathcal{C}_n$. To see that the cone is pointed, assume that $X \in \mathcal{C}_n$ and $-X \in \mathcal{C}_n$, that is, $\boldsymbol{u}^T X \boldsymbol{u} = 0, \forall \boldsymbol{u} \geq \mathbf{0}$. We can choose $\boldsymbol{u} = \boldsymbol{e}_i$, which gives $\boldsymbol{u}^T X \boldsymbol{u} = X_{ii} = 0$, so the diagonal elements of X are 0. Then, choosing $\boldsymbol{u} = \boldsymbol{e}_i + \boldsymbol{e}_j$, we get $\boldsymbol{u}^T X \boldsymbol{u} = X_{ii} + X_{jj} + 2X_{ij} = 0 \implies X_{ij} = 0$, so the off-diagonal elements of X must be 0, too.

This shows that $\mathcal{C}_n$ is proper. Now, we show that $\mathcal{C}_n$ is the dual cone of $\mathcal{C}_n^*$, which also implies that $\mathcal{C}_n^*$ is the dual of $\mathcal{C}_n$, and that $\mathcal{C}_n^*$ is a proper cone, because we know that the dual cone of a proper cone is proper.

$$\begin{aligned}
Y \in \mathrm{dual}(\mathcal{C}_n^*) &\iff \forall X \in \mathcal{C}_n^*, \langle X, Y\rangle \geq 0\\
&\iff \forall q \in \mathbb{N}, \forall \boldsymbol{u}_1, \ldots, \boldsymbol{u}_q \in \mathbb{R}_+^n,\ \sum_{k=1}^{q} \boldsymbol{u}_k^T Y \boldsymbol{u}_k \geq 0\\
&\iff \forall \boldsymbol{u} \geq \mathbf{0},\ \boldsymbol{u}^T Y \boldsymbol{u} \geq 0\\
&\iff Y \in \mathcal{C}_n
\end{aligned}$$

□

Now, consider a mixed-integer QP of the form

$$\begin{aligned}
\underset{\boldsymbol{x}\in\mathbb{R}^n}{\mathbf{minimize}} \quad & \boldsymbol{x}^T Q \boldsymbol{x} + \boldsymbol{c}^T \boldsymbol{x} \\
s.t. \quad & \boldsymbol{a}_i^T \boldsymbol{x} = b_i,\ \forall i \in [m]\\
& \boldsymbol{x} \geq \mathbf{0}\\
& x_i \in \{0,1\},\ \forall i \in B,
\end{aligned} \tag{13}$$

where B is a subset of $[n]$. By applying the general recipe of Proposition 15, this problem admits the following SDP relaxation:

$$\begin{aligned}
\underset{\boldsymbol{x}\in\mathbb{R}^n}{\mathbf{minimize}} \quad & \langle Q, X\rangle + \boldsymbol{c}^T \boldsymbol{x} \\
s.t. \quad & \boldsymbol{a}_i^T \boldsymbol{x} = b_i,\ \forall i \in [m]\\
& \boldsymbol{x} \geq \mathbf{0}\\
& X_{ii} = x_i,\ \forall i \in B,\\
& \begin{bmatrix} X & \boldsymbol{x} \\ \boldsymbol{x}^T & 1 \end{bmatrix} \succeq 0
\end{aligned} \tag{14}$$

Burer proposed to make two modifications to this SDP [2]. First, he adds the (redundant) quadratic equalities $(\boldsymbol{a}_i^T \boldsymbol{x})^2 = b_i^2$ in the original problem formulation, which yields the new constraints $\boldsymbol{a}_i^T X \boldsymbol{a}_i = b_i^2$

in the SDP. Then, he observes that thanks to the constraint $\boldsymbol{x} \geq \mathbf{0}$, the matrix $\begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix}^T$ is not only positive semidefinite, but also completely positive. With these two modifications, the relaxation becomes exact !

Theorem 17 (Burer)**.** *Under some mild assumption (which can always be achieved without loss of generality), the completely positive program*

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\textbf{minimize}} \quad & \langle Q, X\rangle + \boldsymbol{c}^T\boldsymbol{x} \\ s.t. \quad & \boldsymbol{a}_i^T\boldsymbol{x} = b_i,\ \forall i \in [m] \\ & \boldsymbol{a}_i^T X \boldsymbol{a}_i = b_i^2,\ \forall i \in [m] \\ & X_{ii} = x_i,\ \forall i \in B, \\ & \begin{bmatrix} X & \boldsymbol{x} \\ \boldsymbol{x}^T & 1 \end{bmatrix} \succeq_{\mathcal{C}_n^*} 0 \end{aligned} \tag{15}$$

is equivalent *to the mixed-integer QP* (13).

We are not going to prove this result, but we will prove a special case of this result for a QP formulation of the maximum stable set problem in the exercises.

Of course, we can not expect to solve the above completely positive program in polynomial time, as (13) contains many NP-hard optimization problems as special cases. Hence, this shows that completely positive programming is intractable. This is a nice example to show that convex optimization problems can be NP-hard. The difficulty comes from the fact that it is NP-hard to *separate* the completely positive cone, that is, to decide whether a matrix X is completely positive, or to return a separating hyperplane.

Nevertheless, the completely positive formulation can be used to construct hierarchies of SDPs that converge to the optimal value of (13). The first level of this hierarchy is the well known *doubly nonnegative relaxation*, in which the constraint $\begin{bmatrix} X & \boldsymbol{x} \\ \boldsymbol{x}^T & 1 \end{bmatrix} \succeq_{\mathcal{C}_n^*} 0$ is replaced by

$$X \in \mathbb{S}_+^n \cap \mathbb{R}_+^{n\times n} \iff \begin{bmatrix} X & \boldsymbol{x} \\ \boldsymbol{x}^T & 1 \end{bmatrix} \succeq 0, \qquad \begin{bmatrix} X & \boldsymbol{x} \\ \boldsymbol{x}^T & 1 \end{bmatrix} \geq 0.$$

The doubly nonnegative relaxation is known to be exact for $n \leq 4$, but is only an approximation for $n \geq 5$.

References

[1] Austrin, P., Benabbas, S., & Georgiou, K. (2016). Better balance by being biased: A 0.8776-approximation for max bisection. ACM Transactions on Algorithms (TALG), 13(1):2.

[2] Burer, S. (2009). On the copositive representation of binary and continuous nonconvex quadratic programs. Mathematical Programming, 120(2), 479–495.

[3] Chudnovsky, M., Robertson, N., Seymour, P., & Thomas, R. (2006). The strong perfect graph theorem. Annals of mathematics, 51–229.

[4] Frieze, A., & Jerrum, M. (1995, May). Improved approximation algorithms for MAX k-CUT and MAX BISECTION. In International Conference on Integer Programming and Combinatorial Optimization (pp. 1–13).

[5] Goemans, M. X., & Williamson, D. P. (1995). Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. Journal of the ACM (JACM), 42(6), 1115–1145.

[6] Goemans, M. X., & Williamson, D. P. (2004). Approximation algorithms for MAX-3-CUT and other problems via complex semidefinite programming. Journal of Computer and System Sciences, 68(2), 442–470.

[7] Lovsz, L. (1986). An algorithmic theory of numbers, graphs, and convexity (Vol. 50). SIAM.

[8] Nesterov, Y. (1998). Semidefinite relaxation and nonconvex quadratic optimization. Optimization methods and software, 9(1-3), 141–160.