

Any typos should be emailed to aaa@princeton.edu.

This lecture covers some applications of SDPs in combinatorial optimization. Historically, this is the first appearance of SDPs in optimization.

1 The independent set problem

- Throughout this lecture, we consider an undirected graph $G = (V, E)$ with $|V| = n$.
- A *stable set* (or *independent set*) of G is a subset of the nodes, no two of which are connected by an edge. Finding large stable sets has many applications in scheduling.
- The size of the largest stable set of a graph G is denoted by $\alpha(G)$ and is called the *stability number of the graph*.
- The problem of testing if $\alpha(G)$ is larger than a given integer k is NP-hard. (We will prove this soon.)

In this lecture, we will see an SDP relaxation for this problem, due to László Lovász [3].

One natural integer programming formulation of $\alpha(G)$ is the following:

$$\begin{aligned} \alpha(G) = \max_x \quad & \sum_{i=1}^n x_i \\ \text{s.t.} \quad & x_i + x_j \leq 1, \text{ if } \{i, j\} \in E, \\ & x_i \in \{0, 1\}, \quad i = 1, \dots, n. \end{aligned} \tag{1}$$

This problem has an obvious LP relaxation:

$$\begin{aligned} LP_{opt} := \max_x \quad & \sum_{i=1}^n x_i \\ \text{s.t.} \quad & x_i + x_j \leq 1, \text{ if } \{i, j\} \in E, \\ & 0 \leq x_i \leq 1, \quad i = 1, \dots, n. \end{aligned} \tag{2}$$

Clearly, we have $\alpha(G) \leq LP_{opt}$.

Consider now the following SDP due to Lovász from 1979 [3]:

$$\begin{aligned} \vartheta(G) &= \max_{X \in S^{n \times n}} \text{Tr}(JX) \\ \text{s.t. } & \text{Tr}(X) = 1, \\ & X_{i,j} = 0, \text{ if } \{i, j\} \in E, \\ & X \succeq 0, \end{aligned}$$

where J is the $n \times n$ matrix of all ones.

Theorem 1. For any graph G ,

$$\alpha(G) \underset{(1)}{\leq} \vartheta(G) \underset{(2)}{\leq} LP_{opt}.$$

Proof:

We start by showing inequality (1). Let S be a maximum stable set of G and let η be its indicator vector; i.e., a zero-one vector of length n which has a 1 in the i^{th} entry if and only if node i is in S . Define $x = \frac{1}{\sqrt{|S|}}\eta$ where $|S|$ denotes the cardinality of S . Let $X = xx^T$ (i.e., $X_{ij} = x_i x_j$). Then

$$\begin{aligned} X_{ij} &= 0 \text{ if } \{i, j\} \in E, \\ X &\succeq 0, \\ \text{Tr}(X) &= \text{Tr}\left(\frac{1}{|S|}\eta\eta^T\right) = \frac{1}{|S|}\text{Tr}(\eta^T\eta) = 1. \end{aligned}$$

Hence X is feasible for the above SDP and

$$\text{Tr}(JX) = \text{Tr}(11^T X) = \frac{1}{|S|}\text{Tr}(1^T \eta \eta^T 1) = \frac{1}{|S|}(\eta^T 1)^2 = \frac{|S|^2}{|S|} = |S|.$$

Therefore, the optimal value of the SDP can only be larger than the objective value at this one particular feasible solution. This proves inequality (1). You are asked to prove inequality (2) (even a stronger statement) on the homework. \square

To get a better (tighter) LP relaxation than (2), we can add *valid inequalities* to the constraints of (1). These are inequalities that are valid for the feasible points in the integer program (1), but hopefully violated by some feasible points of our original LP. A well-known family of valid inequalities for this problem are the *clique inequalities*. Recall that a clique of a graph G is a complete subgraph of G . The size of a clique is the number of nodes in this subgraph.

- For a 2-clique (i.e., a clique of size 2), the clique inequalities, denoted by C_2 , are $x_i + x_j \leq 1$ everytime there is an edge between i and j (this is what appears in our LP relaxation (2)).
- For a 3-clique, the clique inequalities, denoted by C_3 , are $x_i + x_j + x_k \leq 1$ if $\{i, j, k\}$ is a triangle.
- For a 4-clique, the clique inequalities denoted by C_4 , are $x_i + x_j + x_k + x_l \leq 1$ if $\{i, j, k, l\}$ forms a clique.
- ...

More generally, the clique inequalities of order k , denoted by C_k , are given by

$$x_{i_1} + x_{i_2} + \dots + x_{i_k} \leq 1$$

for $\{i_1, \dots, i_k\}$ defining a clique of size k .

These inequalities lead to new LP relaxations:

$$\begin{aligned} \eta_{LP}^{(k)} := \max_x \quad & \sum_{i=1}^n x_i \\ \text{s.t.} \quad & 0 \leq x_i \leq 1, \quad i = 1, \dots, n \\ & C_1, \dots, C_k. \end{aligned}$$

It is clear that clique inequalities are valid inequalities: any stable set satisfies the clique inequalities (only one node in the clique can belong to the stable set). However, the feasible set of the LP with C_1, \dots, C_k is contained in the feasible set of the LP with only C_1, \dots, C_{k-1} . We are hoping that this inclusion is strict; which may lead to the bound improving in every iteration.

In the homework, you are required to prove that

$$\vartheta(G) \leq \eta_{LP}^{(k)} \quad \forall k \geq 2.$$

In particular, $\eta_{LP}^2 = LP_{OPT}$ proving the second part of Theorem (1). Note that the number of cliques in a graph is exponential in the size of the graph. The above inequality is interesting because it shows that a polynomial-size semidefinite program does better than an exponential-size linear program.

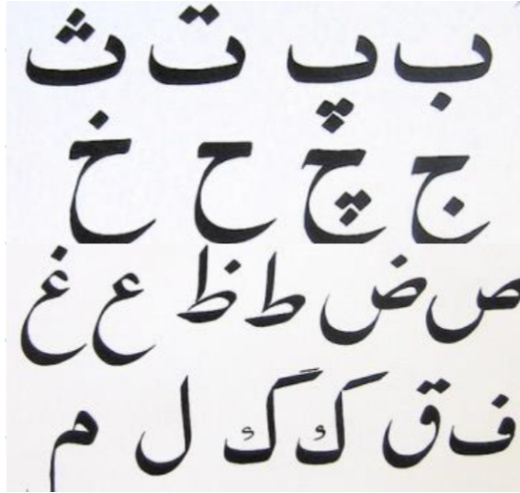


Figure 1: Some letters from the Persian alphabet

2 The Shannon capacity of a graph

The Lovász SDP that we presented previously was in fact introduced to tackle a (rather difficult) problem in coding theory, put forward by Claude Shannon [5].

Suppose you have an alphabet with a finite number of letters v_1, \dots, v_m . You want to transmit messages from this alphabet over a noisy channel. Some of your letters look similar and can get *confused* at the receiver end because of noise. Think for example of the two upperleft letters in Figure 1.

Consider a graph G whose nodes are the letters and which has an edge between two nodes if and only if the two letters can get confused. How many 1-letter words can we send from our alphabet so that we are guaranteed to have no confusion at the receiver? Well, this would be exactly $\alpha(G)$, the stability number of the graph.

But how many 2-letter words can we send with no confusion? How many 3-letter words? And so on...

Note that two k -letter words can be confused if and only if each of their letters can be confused or are equal. It is not hard to see that the number of k -letter words that can be sent without confusion is exactly

$$\alpha(G^K),$$

where $G^k := G \otimes G \otimes \dots \otimes G$ (k times) and \otimes denotes the *strong graph product* defined below.

Definition 1 (Strong graph product). Consider two graphs $G_A(V_A, E_A)$ and $G_B(V_B, E_B)$, with $|V_A| = n$ and $|V_B| = m$. Then their strong graph product $G_A \otimes G_B$ is a graph with nm nodes $V_A \times V_B$ where two nodes (i, k) and (j, l) are connected if and only if

$$(i - j \text{ is an edge in } G_A \text{ or } i = j) \text{ and } (k - l \text{ is an edge in } G_B \text{ or } k = l).$$

Practice: Draw $G_A \otimes G_B$ if G_A and G_B are as given below.

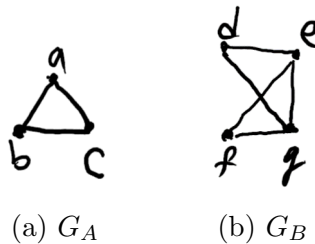


Figure 2: Graphs G_A and G_B

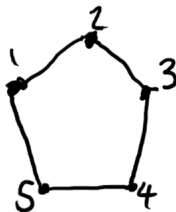
Lemma 1.

$$\alpha(G_A \otimes G_B) \geq \alpha(G_A) \cdot \alpha(G_B).$$

In particular, $\alpha(G^k) \geq \alpha^k(G)$.

Proof: Let $S_1 = \{u_1, \dots, u_s\}$ be a maximum stable set in G_A and $S_2 = \{v_1, \dots, v_r\}$ be a maximum stable set in G_B . Then $S_1 \times S_2 = \{u_i v_j : u_i \in S_1, v_j \in S_2\}$ is a stable set in $G_A \otimes G_B$. \square

It is quite possible however to have $\alpha(G_A \otimes G_B) > \alpha(G_A) \cdot \alpha(G_B)$. Here is an example: let $G_A = G_B = C_5$ (i.e., a cycle on five nodes).



Then $\alpha(G_A) \cdot \alpha(G_B) = 2 \cdot 2 = 4$, but $\alpha(G_A \otimes G_B) = 5$ since the set $\{a_1 b_1, a_2 b_3, a_3 b_5, a_5 b_4, a_4 b_2\}$ e.g. is a stable set in $G_A \otimes G_B$.

Definition 2 (Shannon capacity). *The Shannon capacity of a graph G , denoted by $\Theta(G)$, is defined as*

$$\Theta(G) = \lim_{k \rightarrow \infty} \alpha^{1/k}(G^k).$$

One can show (e.g., by using Fekete's lemma) that the limit always exists and can be equivalently written as

$$\Theta(G) = \sup_k \alpha^{1/k}(G^k).$$

Lemma 2 (Fekete's lemma). *Consider a sequence $\{\alpha_k\}$ that is superadditive; i.e., satisfies*

$$a_{n+m} \geq a_n + a_m, \quad \forall m, n$$

then $\lim_{k \rightarrow \infty} \frac{a_k}{k}$ exists in $\mathbb{R} \cup \{+\infty\}$ and is equal to $\sup_k \frac{a_k}{k}$.

To see the claim for $\Theta(G)$, apply this lemma to the sequence $\{\log \alpha(G^k)\}$ (the limit cannot be infinite here as we see below that the sequence is always upperbounded) and observe that the strong graph product is associative.

The definition of $\Theta(G)$ itself suggests a natural way of obtaining lower bounds: for any k , we have

$$\alpha^{1/k}(G^k) \leq \Theta(G).$$

But how can we obtain an upper bound?

Note that for a graph G with n vertices, we always have

$$\Theta(G) \leq n$$

because clearly $\alpha(G^k) \leq n^k$, $\forall k$. But an upperbound of n is almost always very loose.

In 1979, Lóvasz [3] gave an algorithm for computing upper bounds on the Shannon capacity that resolved the exact value for many more graphs than previously known. In the process, he invented semidefinite programming. (Of course, he didn't call it that.)

We now give a proof of his result — that $\Theta(G) \leq \vartheta(G)$ — in the language of SDP.

Theorem 2 (Lovász, [3]).

$$\Theta(G) \leq \vartheta(G)$$

Here, $\vartheta(G)$ is the optimal solution of the SDP seen before

$$\begin{aligned} \vartheta(G) &:= \max_X \operatorname{Tr}(JX) \\ \text{s.t. } &\operatorname{Tr}(X) = 1, X \succeq 0, \\ &X_{ij} = 0, \{i, j\} \in E. \end{aligned} \tag{3}$$

We have already shown that for any graph G :

$$\alpha(G) \leq \vartheta(G), \tag{4}$$

$$\Theta(G) = \sup_k \alpha^{1/k}(G^k). \tag{5}$$

So if we also show that

$$\vartheta(G^k) \leq \vartheta^k(G), \tag{6}$$

then we get

$$\Theta(G) = \sup_k \alpha^{1/k}(G^k) \leq \sup_k \vartheta^{1/k}(G^k) \leq \sup_k \vartheta(G) = \vartheta(G),$$

where the first equality is from (5), the first inequality is from (4), the second inequality is from (6), hence proving Theorem 2.2. The inequality in (6) is a direct corollary of the following theorem.

Theorem 3. *For any two graphs G_A and G_B , we have*

$$\vartheta(G_A \otimes G_B) \leq \vartheta(G_A) \cdot \vartheta(G_B).$$

To prove this theorem, we need to take a feasible solution to SDP (3) applied to $G_A \otimes G_B$ and from it produce feasible solutions to SDP (3) applied to G_A and to G_B . This doesn't seem like a straightforward thing to do since we should somehow apply a "reverse Kronecker product" ¹ operation to our original solution. It would have been much nicer if we could turn the feasibility implication around in the other direction — and we can, by taking the dual!

¹See proof of Theorem 3 to see why we are mentioning Kronecker products.

What is the dual of SDP (3)? Recall the standard primal dual pair that we derived before:

$$\begin{aligned}
 (P) \quad & \min_{X \in S^{n \times n}} \text{Tr}(CX) \\
 & \text{Tr}(A_i X) = b_i, i = 1, \dots, m \\
 & X \succeq 0 \\
 (D) \quad & \max_{y \in \mathbb{R}^m} b^T y \\
 & \sum_{i=1}^m y_i A_i \preceq C
 \end{aligned}$$

Just by pattern matching we obtain the dual of SDP (3) in Figure 3.

$$\begin{array}{ccc}
 \begin{array}{l}
 - \max_{t, y_{ij}} t \\
 tI + \sum y_{ij} E_{ij} \preceq -J
 \end{array} & \xleftrightarrow{\text{dual}} & \begin{array}{l}
 - \min \text{Tr}(-JX) \\
 \text{Tr}(X) = 1, X_{ij} = 0, i, j \in E, X \succeq 0
 \end{array} \\
 \downarrow & & \\
 \begin{array}{l}
 \min_{t, y_{ij}} -t \\
 -tI - \sum y_{ij} E_{ij} \succeq J
 \end{array} & \xleftrightarrow{\quad} & \begin{array}{l}
 \min_{t, y_{ij}} t \\
 tI + \sum y_{ij} E_{ij} \succeq J,
 \end{array} \quad \text{SDP1d}
 \end{array}$$

Figure 3: Definition of SDP1d.

Remark: Here, E_{ij} is a matrix with a one in $(i, j)^{th}$ and $(j, i)^{th}$ position and zero elsewhere.

Let us rewrite SDP1d slightly:

$$\begin{aligned}
 & \min_{t \in \mathbb{R}, Z \in S^{n \times n}} t \\
 & \text{s.t. } tI + Z - J \succeq 0, \\
 & Z_{ij} = 0, \text{ if } i = j \text{ or } \{i, j\} \notin E.
 \end{aligned} \tag{7}$$

Unlike LPs, SDPs do not always enjoy the property of having zero duality gap. However, since SDP1 in (3) and SDP1d in (7) are both *strictly feasible* (why?), there is indeed no

duality gap. Hence, $\vartheta(G)$ equals the optimal value of (7).

We now show that $\vartheta(G_A \otimes G_B) \leq \vartheta(G_A)\vartheta(G_B)$. First we prove a simple lemma from linear algebra.

Lemma 3. *Consider two matrices $X \in S^{n \times n}$, $Y \in S^{m \times m}$. Then,*

$$X \succeq 0 \text{ and } Y \succeq 0 \Rightarrow X \otimes Y \succeq 0,$$

where \otimes denotes the matrix Kronecker product ².

Proof: Let X have eigenvalues $\lambda_1, \dots, \lambda_n$ with eigenvectors v_1, \dots, v_n and Y have eigenvalues μ_1, \dots, μ_m with eigenvectors w_1, \dots, w_m . Then

$$\begin{aligned} (X \otimes Y)(v_i \otimes w_j) &= Xv_i \otimes Yw_j \\ &= \lambda_i \mu_j v_i \otimes w_j. \end{aligned}$$

where you can check the first equality by writing the expressions out. Hence $X \otimes Y$ has mn eigenvalues given by $\lambda_i \mu_j$, $i = 1, \dots, n$, $j = 1, \dots, m$, which must all be nonnegative. \square

Proof of Theorem 3: Let G_A and G_B be graphs on n and m nodes respectively. Consider a feasible solution $(t_A \in \mathbb{R}, A \in S^{n \times n})$ to SDP1d for G_A and $(t_B \in \mathbb{R}, B \in S^{m \times m})$ to SDP1d for G_B .

We claim that the pair $(t_A t_B, C)$ with

$$C := t_A I_n \otimes B + t_B A \otimes I_m + A \otimes B$$

is feasible for SDP1d applied to $G_A \otimes G_B$ (and obviously has objective value $t_A t_B$). This would finish the proof. By assumption, we have

$$\begin{aligned} t_A I_n + A - J_n \succeq 0 &\Rightarrow t_A I_n + A + J_n \succeq 0 \text{ (because } 2J_n \text{ is psd)} \\ t_B I_m + B - J_m \succeq 0 &\Rightarrow t_B I_m + B + J_m \succeq 0. \end{aligned}$$

By Lemma 3 we have

$$(t_A I_n + A - J_n) \otimes (t_B I_m + B + J_m) \succeq 0, \tag{8}$$

$$(t_A I_n + A + J_n) \otimes (t_B I_m + B - J_m) \succeq 0. \tag{9}$$

²If T is an $m \times n$ matrix and V is a $p \times q$ matrix, then the Kronecker product $T \otimes V$ is the $mp \times nq$ matrix: $\mathbf{T} \otimes \mathbf{V} = \begin{bmatrix} t_{11}\mathbf{V} & \cdots & t_{1n}\mathbf{V} \\ \vdots & \ddots & \vdots \\ t_{m1}\mathbf{V} & \cdots & t_{mn}\mathbf{V} \end{bmatrix}$.

$$(8) \Rightarrow (t_A I_n + A) \otimes (t_B I_n + B) + (t_A I_n + A) \otimes J_m - J_n(t_B I_m + B) - J_n \otimes J_m \succeq 0$$

$$(9) \Rightarrow (t_A I_n + A) \otimes (t_B I_n + B) - (t_A I_n + A) \otimes J_m + J_n(t_B I_m + B) - J_n \otimes J_m \succeq 0$$

Averaging both LMIs, we get

$$(t_A I_n + A) \otimes (t_B I_m + B) - J_n \otimes J_m \succeq 0.$$

This implies that

$$t_A t_B I_{nm} + t_A I_n \otimes B + t_B A \otimes I_m + A \otimes B - J_{nm} \succeq 0.$$

If we let

$$C := t_A I_n \otimes B + t_B A \otimes I_m + A \otimes B.$$

We see that the required LMI is met. Lastly, we need to check that $C_{ij} = 0$ if $\{i, j\} \notin E$, or if $i = j$.

Let us reindex $\{i, j\}$ in $G_A \otimes G_B$ as $\{\tilde{i}, \tilde{j}\}, \{k, l\}$ where \tilde{i}, \tilde{j} are nodes in G_A and k, l are nodes in G_B . The fact that there is no edge between the super node $\{\tilde{i}, k\}$ and $\{\tilde{j}, l\}$ in $G_A \otimes G_B$ means that either $\tilde{i} - \tilde{j}$ is not an edge in G_A or $k - l$ is not an edge in G_B or both.

- First observe that the $C_{ii} = 0$ because $A_{ii} = 0$ and $B_{ii} = 0$.
- Now consider $\{i, j\} = \{\tilde{i}, \tilde{j}\}, \{k, l\} \notin E$. We have

$$C_{(\tilde{i}, \tilde{j}), (k, l)} = A_{\tilde{i}, \tilde{j}} B_{k, l} + t_A I_{\tilde{i}, \tilde{j}} B_{k, l} + t_B A_{\tilde{i}, \tilde{j}} I_{k, l}.$$

$$\begin{bmatrix} \cdot & \textcircled{\cdot} \\ \cdot & \textcircled{\cdot} \end{bmatrix} \otimes \begin{bmatrix} \cdot & \cdot & \textcircled{\cdot} \\ \cdot & \textcircled{\cdot} & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \textcircled{\cdot} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

$$A \otimes B$$

Note that either $A_{\tilde{i}, \tilde{j}}$ or $B_{k, l}$ must be zero. Wlog, assume that $B_{k, l} = 0$. We only have to worry about $A_{\tilde{i}, \tilde{j}} I_{k, l}$. If $k \neq l$, then $I_{k, l} = 0$ and we are done. If $k = l$, then $\tilde{i} - \tilde{j}$ cannot be an edge or else the letters would get confused (and hence $\{\tilde{i}, \tilde{j}\}, \{k, l\} \in E$). Hence if $k = l$, we must have $A_{\tilde{i}, \tilde{j}} = 0$. \square

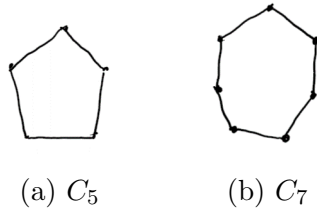


Figure 4: The odd cycles of length 5 and 7.

Example 1: What is $\Theta(C_5)$?

- $\alpha^{1/2}(C_5^2) = \sqrt{5} \leq \Theta(C_5)$. (We already presented a stable set of size 5 to C_5^2 right after the proof of Lemma 1.)
- $\Theta(C_5) \leq \vartheta(C_5) = \sqrt{5}$. (You can see this by solving the SDP, whose solution is easy enough to work out analytically.)

These two results imply that $\Theta(C_5) = \sqrt{5} = 2.236\dots$

Lovász [3] settled the exact value of $\Theta(C_5)$ more than 20 years after Shannon’s paper [5].

Example 2: What is $\Theta(C_7)$? This is an open problem! The best bounds we know so far are

$$\alpha(C_7^5)^{1/5} \leq 3.2271 \leq \Theta(C_7) \leq 3.3177 = \vartheta(C_7).$$

(See [4] for a proof of the lowerbound.)

- The exact value of C_7 = an automatic A in ORF523.
- Showing that $\Theta(C_7) < \vartheta(C_7)$ (if true) = an automatic 100/100 on the final exam.

To improve the upper bound via semidefinite programming one needs to come up with an SDP that produces a sharper bound than the Lovász SDP. While we have many SDPs that provide better upper bounds on the stability number of a graph (e.g., via the “*sum of squares hierarchy*”, which we will see soon), it is not known whether these stronger bounds are also valid upper bounds on the Shannon capacity number. For this to be the case, one needs to prove that the SDP optimal value “*tensorizes*”; i.e., satisfies the relation in Theorem 3.

Notes

Further reading for this lecture can include Chapter 2 of [2] and Chapter 3 of [1].

References

- [1] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, volume 2. SIAM, 2001.
- [2] M. Laurent and F. Vallentin. *Semidefinite Optimization*. 2012. Available at http://www.mi.uni-koeln.de/opt/wp-content/uploads/2015/10/laurent_vallentin_sdo_2012_05.pdf.
- [3] L. Lovász. On the shannon capacity of a graph. *Information Theory, IEEE Transactions on*, 25(1):1–7, 1979.
- [4] K.A. Mathew and P.R.J. Östergård. New lower bounds for the Shannon capacity of odd cycles. 2015. Available at <http://arxiv.org/pdf/1504.01472v1.pdf>.
- [5] C. E. Shannon. The zero error capacity of a noisy channel. *Information Theory, IRE Transactions on*, 2(3):8–19, 1956.