Lovász Theta Function, Semidefinite Programs and Algorithms

Parts 3,4

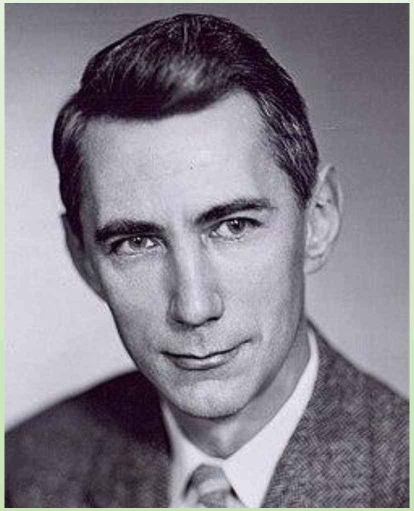
Lecturer: Rakesh Venkat

Short Term Program on “Graphs, Matrices and Applications”

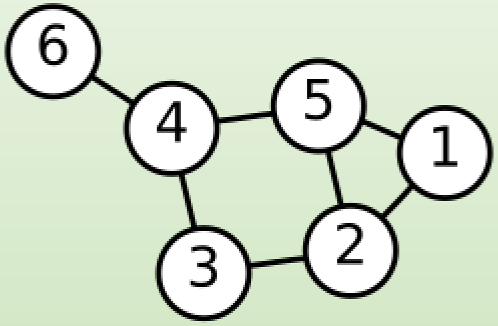
Indian Institute of Technology, Hyderabad

4th Oct 2025 (Sat)

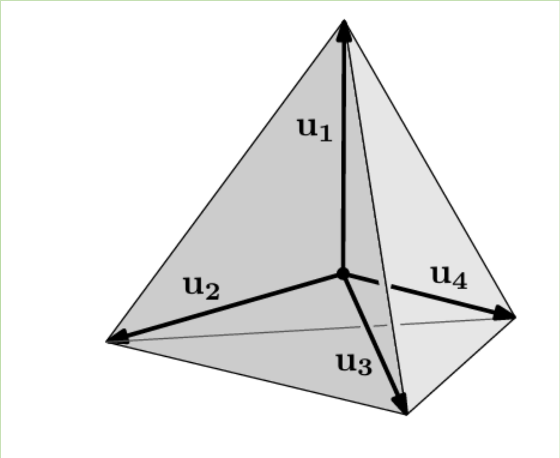
Shannon, Lovász, Graphs and Geometry



Claude Shannon



László Lovász



Recap from Lectures 1,2

- Optimization in CS
- Shannon Capacity
- The Theta Function of a Graph
- Lovasz Bound
- Shannon capacity of the 5-cycle

Outline for Today

- Linear and Semidefinite Programs
- Semidefinite programs for the Theta function
- Sandwich theorem and perfect graphs
- Relaxations and Rounding: Combinatorial optimization. Examples.
- Goemans Williamson Max-Cut algorithm
- SDPs for Coloring

Quick Recap

- Independence number $\alpha(G)$ and chromatic number $\chi(G)$ are hard-to-compute quantities of G, but important from both theoretical and practical standpoints
- Given a graph G, we are interested in finding the value of
$$S(G) \coloneqq \sup_{\{k \in \mathbb{N}\}} \alpha\left(G^k\right)^{\frac{1}{k}}$$
  - This quantity characterizes the measure of information that can be sent across a channel per symbol when edges of G show which alphabets cannot be sent together
- Lovasz formulated the *theta* function $\vartheta(G)$, that satisfies:
$$S(G) \leq \vartheta(G)$$
- $\vartheta(G)$ is a function that utilizes *orthonormal representations* of G, and is easier to analyze than $S(G)$
  - For instance, $\vartheta\left(G^k\right) \leq \vartheta(G)^k$
- Using this, Lovasz showed that $\vartheta(C_5) = \sqrt{5}$, implying $S(G) = \sqrt{5}$

Recap: OR and theta function

$V = \{1, 2, \ldots n\}$

- **Orthonormal Representation (OR)** for G: A set of unit vectors $\{u_1, \ldots, u_n\}$ satisfying:
  - $u_i^T u_j = 0 \cancel{\Leftrightarrow}$ if $\{i, j\} \in \bar{E}$

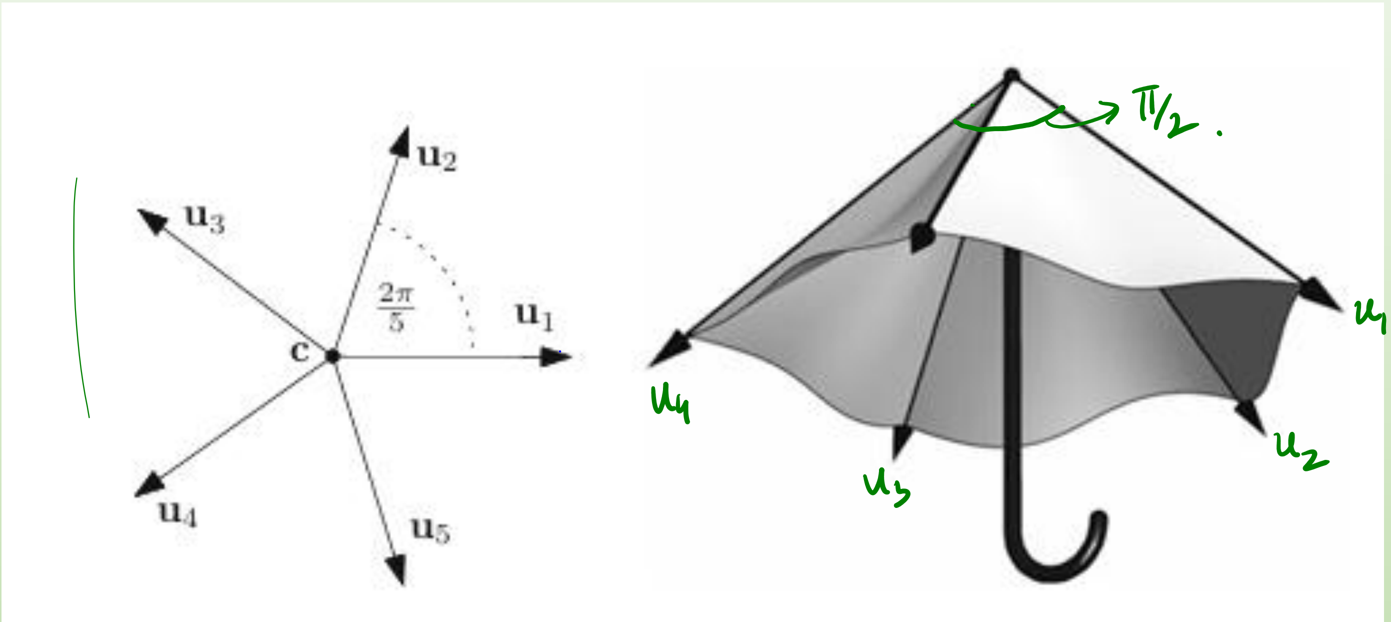
- The **value** of U is defined as

$$\vartheta(U) := \min_{c:\|c\|=1} \max_i \frac{1}{(c^T u_i)^2}$$

c = "handle" of the OR.

- $\vartheta(G) := \min_{U:\text{OR for } G} \vartheta(U)$

Illustration of OR for C_5



$$\boxed{\vartheta(u) = \frac{1-\cos 4\pi/5}{-\cos 4\pi/5} = \sqrt{5}}$$

At satisfying point of OR:

$$u_i = \left(\cos\frac{2\pi i}{5},\ \sin\frac{2\pi i}{5},\ z\right), \text{ for } i = 1, 2, 3, 4, 5$$

$$v_5^T u_2 = 0. \Rightarrow (1, 0, z)^T \begin{pmatrix} \cos 4\pi/5 \\ \sin 4\pi/5 \\ z \end{pmatrix} = 0$$

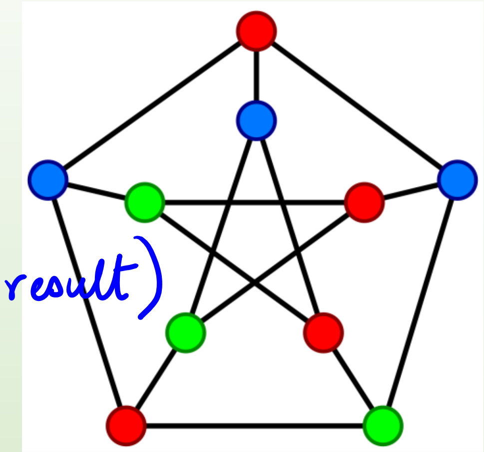
$$\Rightarrow \cos\frac{4\pi}{5} + z^2 = 0.$$

Some definitions

- $\omega(G)$: Size of the maximum clique in G

$\alpha(\bar{G}) = \omega(G)$

The Sandwich Theorem



Theorem [Lovasz, 1979]: For all graphs G, (Main result)

$$\omega(\bar{G}) \underbrace{\le}_{\text{Shown yesterday}} \vartheta(G) \overbrace{\le}^{\text{Today}} \chi(\bar{G})$$

Shown yesterday: $\underset{\omega(\bar{G})}{\underset{\parallel}{\alpha(G)}} \le \vartheta(G)$.

(Notice: $\chi(\bar{G}) \geq \omega(\bar{G})$).

$\rightarrow$ Go through "computational aspects of $\vartheta(G)$"

Computation of $\vartheta(G)$

- $\vartheta(G)$ is a *relaxation* of $\alpha(G)$

- $\vartheta(G)$ is an *optimization* problem

min over all OR's $\mathcal{U}$

of $\vartheta(\mathcal{U})$.

- Can a solution to this optimization problem be *computed* efficiently?

General form of an optimization problem

$$Z^* := \quad \min \quad \text{or} \max f(x)$$
$$\text{subject to}: \quad g_1(x) = 0$$
$$g_2(x) \geq 0$$
$$\ldots$$

x: variables
$x \in \mathbb{R}^n$ or similar

- Some optimization problems are "easy" computationally, others are hard
- If $f, g_{\mathrm{i}}'s$ are "simple", then the optimization problem may be efficiently solved
  - Will assume that an optimum exists

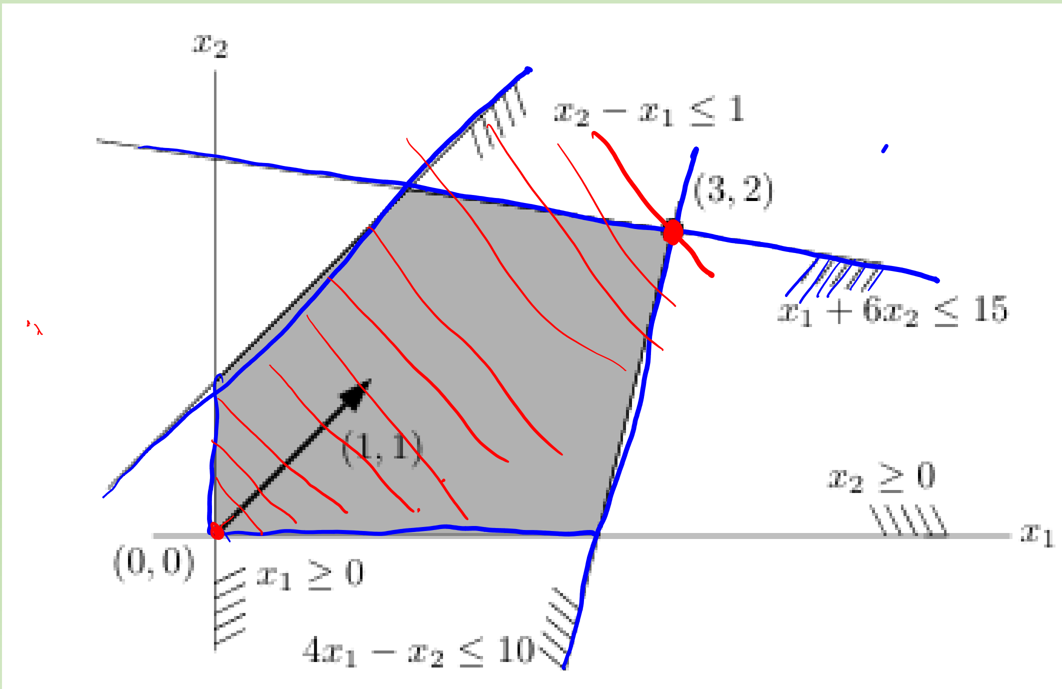
Example: f, g linear

Maximize $x_1 + x_2$ objective (2-variables)

Subject to:

$$\left.\begin{aligned} x_1, x_2 &\geq 0 \\ x_2 - x_1 &\leq 1 \\ x_1 + 6x_2 &\leq 15 \\ 4x_1 - x_2 &\leq 10 \end{aligned}\right\} \text{constraints.}$$

Feasible solutions: $(x_1, x_2) \in \mathbb{R}^2$ s.t. they satisfy all the constraints.



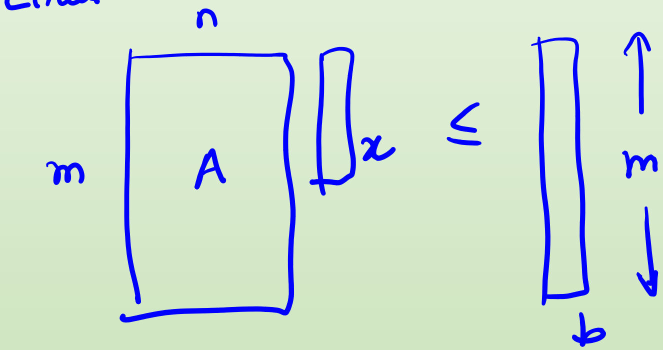
General *Linear Program*

$$\max\ c^T x \rightarrow \sum_{i=1}^{n} c_i x_i$$

$$s.t. \quad Ax \le b \rightarrow \text{Linear contraints}$$

$$x \ge 0$$

$$x \in \mathbb{R}^n$$



Here, $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$

- An optimal solution to a Linear Program can be found efficiently computationally
  - Given inputs c, A, b, we can find an x^* optimizing the above

More constraints

$$\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & & \\ & & \ddots & \\ & & & x_{nn} \end{pmatrix}$$

- Say, now, the variables are entries x_{ij} of a symmetric matrix
- The space of variables is, therefore: (n^2 variables)

$$\boxed{\mathrm{SYM}_n = \{X \in \mathbb{R}^{n \times n}: x_{ij} = x_{ji}\}}$$

- Let's generalize the previous Linear Program:

- $\max\ c^T x$

  $s.t. \quad Ax \leq b$

  $\underline{x \geq 0}$

  $x \in \mathbb{R}^n$

$\rightsquigarrow$

$\max \sum_{ij} c_{ij} x_{ij}$ (Linear)

s.t. Linear constraints.

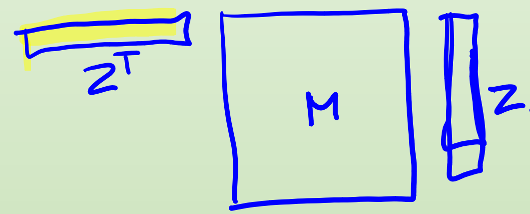
$\sum_{ij} A_{ij}^{(k)} x_{ij} \leq \underline{b^{(k)}} \quad k = 1, \ldots, m$

$X \succeq 0$ (X is psd)

$X \in \mathrm{SYM}_n$.

Positive Semidefinite Matrices

- **Fact:** Let $M \in \mathrm{SYM}_n$. The following are equivalent:
  1. M is positive semidefinite: All the eigenvalues of M are non-negative
  2. $z^T M z \geq 0$ for all $z \in \mathbb{R}^n$
  3. $M = U^T U$, for some matrix U



$\geq 0.$

$\forall z$

(3) ⇒ (2) : $z^T M z = z^T U^T U z$

$= \|Uz\|^2 \geq 0.$

Semidefinite Program (SDP)

max/min $\sum_{ij} c_{ij} x_{ij}$

s.t. Linear constraints on x_{ij}'s. (eg: $4x_{11} + 3x_{21} + 5x_{23} \geq 10$)

$X \succeq 0.$

$X \in SYM_n.$

- Omitting technical conditions*, we can *efficiently* find optimal solutions* to Semidefinite Programs!

A slight caveat

- max $-x_{11}$
- s.t. $x_{12} = 1$
- $X \underbrace{\succcurlyeq}_{X \text{ is psd.}} 0, X \in \mathrm{SYM}_2$

- What is the optimum? → Exists, but can't be attained

→ Solvers give approximately optimal solutions

Back to the Theta Function

- $\vartheta(G)$ as an optimization problem

Find vectors $u_1 \cdots u_n$: $u_i^T u_j = 0$ for $\{i,j\} \in \bar{E}$
and handle 'c'

and $\min_{\substack{c, \\ u_1 \cdots u_n}} \max_i \frac{1}{(c^T u_i)^2}$

s.t. OR constraint.

For any feasible
$u_1 \cdots u_n$
and c,
the optimum t
$= \min_{i \in [n]} (c^T u_i)$

Claim: Consider the optimization problem

$$
\begin{aligned}
Z^* := \max \quad & t \\
\text{s.t.} \quad & u_i^T u_j = 0 \quad \forall \{i,j\} \in \bar{E} \\
& c^T u_i \geq t \quad \forall i \in \{1, \cdots n\}. \\
& \forall i : \|u_i\| = 1, \ \|c\| = 1
\end{aligned}
$$

$\{u_i \in \mathbb{R}^n, \ c \in \mathbb{R}^n, \ t\} \rightarrow$ variables.

$$\boxed{Z^* = \frac{1}{\sqrt{\vartheta(G)}}}$$

or, $\vartheta(G) = \frac{1}{(Z^*)^2}$.

Previous is actually a SDP

$$\max \ t$$

s.t. $u_i^T u_j = 0 \quad \forall \{i,j\} \in \overline{E}$

$c^T u_i \geq t$

$\|u_i\|^2 = 1, \quad \|c\|^2 = 1.$

→ A "vector program"

(variables are vectors).

Introduce $Y \in SYM_{n+1}$

$$\begin{array}{c|c} & j \\ \hline \begin{array}{c} 0 \\ \vdots \\ 2 \\ \vdots \\ i \\ \vdots \\ n \end{array} & \begin{pmatrix} \cdot & \cdot & \cdot \\ & y_{ij} & \\ & & \end{pmatrix} \end{array}$$

$y_{ij} := u_i^T u_j \quad i, j > 0.$

$y_{0,j} := c^T u_j \quad \forall j > 0.$

equiv

$$\max \ t$$

s.t. $y_{ij} = 0 \quad \forall \{i,j\} \in \overline{E}$

$y_{ii} = 1 \quad \forall i$

$y_{0j} \geq t \quad \forall j$

$Y \succeq 0$ →

$t \in \mathbb{R}.$

Why does $Y \succeq 0$ make sense?

Ans: $Y \succeq 0 \iff \exists$ matrix U s.t.

$Y = U^T U$. (defn of psd)

$$\begin{bmatrix} - u_1 - \\ - u_2 - \\ \vdots \\ - u_n - \end{bmatrix} \begin{bmatrix} | & & | \\ u_1 & \cdots & u_n \\ | & & | \end{bmatrix}.$$

Summary: $\dfrac{1}{\sqrt{\vartheta(G)}}$ is the optimal value of a Semidefinite Program.

SDP #2 for the Theta Function

$G = (V, E)$

$V = \{1, \dots n\}$.

$$\min \quad t$$

$$\text{s.t.} \quad u_i^T u_j = -1 \quad \forall \{i,j\} \in E$$

$$\|u_i\|^2 = t-1. \quad \forall i.$$

Can convert into $\rightarrow$ (SDP) by introducing $y_{ij} := u_i^T u_j$

Variables: vectors $u_1, u_2, \dots u_n \in \mathbb{R}^n$.

$t \in \mathbb{R}$.

- Why is the optimal of this equal to $\vartheta(G)$?

Let Z_2^* be the optimal of the above optimization problem.

Lemma: $Z_2^* = \vartheta(G)$.

Proof that $\vartheta(G) = Z_2^*$

$$Z_2^* = \min \ t \quad \Big\} \text{ SDP2.}$$

$$\text{s.t.} \quad v_i^T v_j = -1 \quad \forall \{i,j\} \in \bar{E}$$

$$\|v_i\|^2 = t-1.$$

$$\text{vars: } t \in \mathbb{R}, \ v_i\text{'s} \in \mathbb{R}^n.$$

- Part 1: $Z_2^* \leq \underline{\underline{\vartheta(G)}}$

- Given an optimal OR and handle c, construct a feasible solution to SDP2

Let $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$. $\rightarrow$ optimal OR, handle c.

A Feasible soln to SDP2:

$$v_i := c - \frac{u_i}{(c^T u_i)}.$$

For $i,j \in \bar{E}$: $$v_i^T v_j = c^T c - \frac{c^T u_i}{c^T u_i} - \frac{c^T u_j}{(c^T u_j)} + \cancel{\frac{u_i^T u_j}{(c^T u_i)(c^T u_j)}} \rightarrow 0.$$

$$= 1 - 1 - 1 = -1.$$

Part 2: $Z_2^* \geq \vartheta(G)$

- From an optimal SDP solution, construct an OR for G

(skip the proof here).

Till now

- Optimization formulation (SDP) for the Theta Function

- Goal: To show that $\vartheta(G) \leq \chi(\bar{G})$

shown $\left\{ \begin{array}{c} \text{\rotatebox{90}{$\leq$}} \\ \omega(\bar{G}) \end{array} \right.$

- Need to relate it to a coloring in the complement graph

Proof strategy

To show: $\boxed{\vartheta(G) \le \chi(\bar{G})}$.

- Show that if $\bar{G}$ has a k-coloring, then SDP2 has a feasible solution with value at most k $\Rightarrow \vartheta(G) \le k$.

SDP2:

$Z2 = \min t$

s.t. $y_{ij} = -1$ forall $i, j \in \bar{E}$

$y_{ii} = 1$

$Y \succcurlyeq 0$

$\equiv$

SDP 2 : (vector form)

$Z_2^* = \min \quad t$

s.t. $\quad v_i^T v_j = -1 \quad \forall \{i,j\} \in \bar{E}$

$\|v_i\|^2 = 1$

Variables: v_i's and $t \in \mathbb{R}$

vectors.

k - colorings and Vector k-colorings

$G' = (V, E')$

k-coloring of $G' = (V, E')$: $\chi : V \to \{1, \dots k\}$.

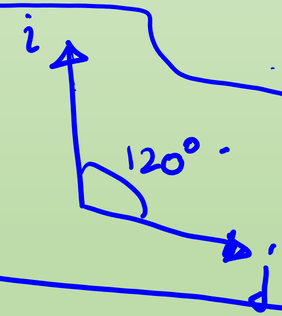
s.t. $\forall \{i, j\} \in E'$, $\chi(i) \neq \chi(j)$.

Definition: A vector k-coloring is an assignment of unit vectors to vertices

$$\gamma : V \to S^{n-1}$$

$$\text{s.t. } \forall \{i, j\} \in E', \quad \gamma(i)^T \gamma(j) = -\frac{1}{k-1}.$$

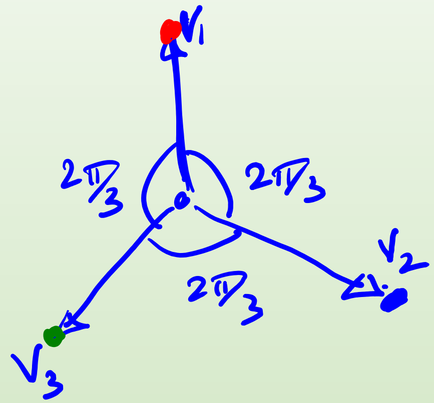
eg: $k=3$. $\gamma(i)^T \gamma(j) = -\frac{1}{2}$.



Claim: If G' has a k-coloring, it has a vector k-coloring.

Want : $r(i) \in \mathbb{R}^n$ s.t. $\forall \{i,j\} \in G'$: $r(i)^T r(j) = \frac{-1}{k-1}$.

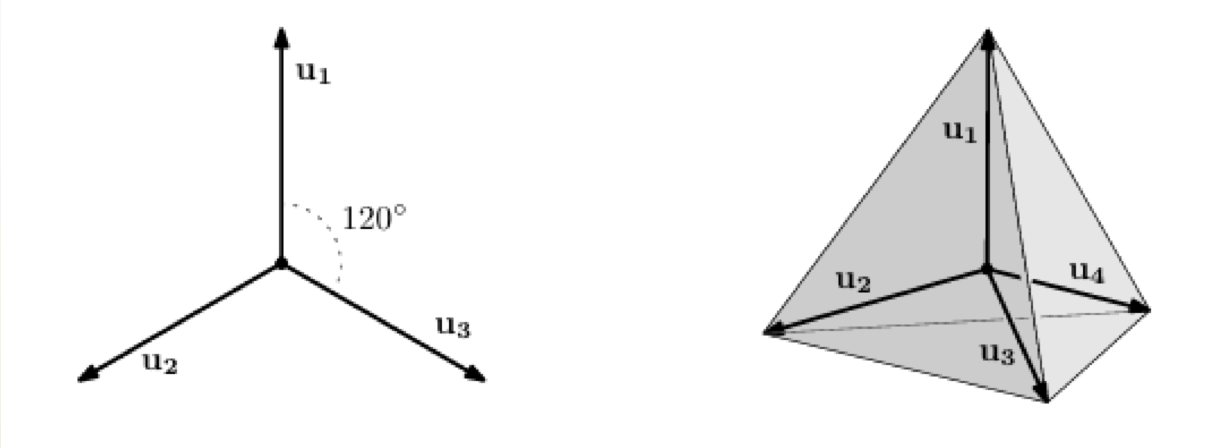
eg: $k=3$.



3-colorable graph

$$r(i) = V_{\chi(i)}.$$

Solutions for $k = 3,4$

- 

In general: $\quad r(i) := \dfrac{e_{x(i)} - \frac{1}{k}\sum_{\ell=1}^{k} e_\ell}{\left\| e_{x(i)} - \frac{1}{k}\sum_{\ell=1}^{k} e_\ell \right\|}$

Wanted to show: $\vartheta(G) \leq \underbrace{\chi(\bar{G})}_{=k}$. then it has a vector k-coloring.

$$\min \quad t$$
$$v_i^T v_j = -1 \quad \forall \{i,j\} \in \bar{E}$$
$$\|v_i\|^2 = t-1$$

$\wr\!\!\downarrow$!!!

$\leq k$.

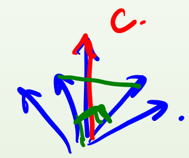
$$\min \quad t$$
$$v_i^T v_j = -\frac{1}{t-1} \quad \forall \{i,j\} \in \bar{E}$$
$$\|v_i\|^2 = 1.$$

$\Rightarrow$ constraints for v_i's to be a vector t-coloring!

$\Rightarrow \text{opt} \leq k$.

$\bar{G}$ has a k-coloring $\Rightarrow$ it has a vector k coloring $\Rightarrow$ there is a feasible solution to the SDP with value k

$$\left(\alpha(G^k)\right)^{1/k} \le \vartheta(G) =$$



$$\min_{c} \max_{i \in V} \frac{1}{(c^T u_i)^2}.$$

Recap

max $f(x)$ $\quad x$: variables

s.t. $g_1(x) \geq 0$ $\quad = (x_1, \cdots, x_n)$

$g_2(x) \geq 0$

- An SDP (Semidefinite Program) is an optimization problem where the n^2 variables are entries x_{ij} of a symmetric psd matrix $X \in \mathrm{SYM}_n$, and the objective and constraints are linear in $x_{ij}'s$

- $\vartheta(G)$ can be expressed as a minimization SDP: i.e, as an optimization problem which can be solved efficiently

- If $\bar{G}$ has a k-coloring, then we can use it to find a feasible solution to the SDP with value k.

- This gives us: $\omega(G) \leq \vartheta(G) \leq \chi(\bar{G})$

Perfect Graphs

Sandwich Thm: $\forall G: \omega(\bar{G}) \leq \vartheta(G) \leq \chi(\bar{G})$

- **Perfect Graphs** are graphs G where $\omega(G') = \chi(G')$ for all vertex-induced subgraphs G' of G

- By sandwich theorem, can compute $\omega(G)$ and $\chi(G)$ for all perfect graphs G efficiently (since by sandwich theorem, both are $= \vartheta(\bar{G})$)

- Examples: Bipartite graphs, Chordal graphs, Interval Graphs

Perfect graph theorems

- **Weak Perfect Graph Theorem** (Lovasz 1972)

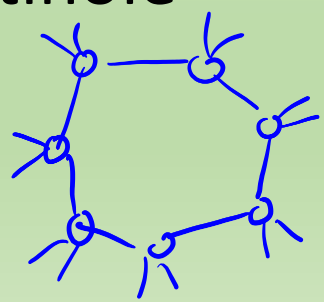
G is a perfect graph if and only if $\bar{G}$ is perfect

$\Rightarrow$ Can compute $\chi(G)$, $\alpha(G)$, $\omega(G)$ from $\vartheta(G)$ and $\vartheta(\bar{G})$.

- **Strong Perfect Graph Theorem** (CRST, 2006, Annals of Math)

A graph is perfect iff it contains no odd hole (odd induced cycle of length ≥ 5) and no odd antihole

- Smallest non-perfect graph is C_5



SDPs and Designing Algorithms

(Part 4)

Easy and hard problems

- “Easy” problems: Problems for which there are efficient (polynomial-time) algorithms
  - Sorting n numbers
  - Matching
  - Finding a minimum Spanning tree
  - Min-cut
  - Max-Flow

- “Hard” problems: NP-hard, optimal solutions likely cannot be found efficiently
  - Minimum Vertex-Cover
  - Max-Cut
  - Minimum Set-Cover
  - Maximum Independent Set

Approximation

- Area of approximation algorithms: design efficient algorithms that *provably* find an *approximately* optimal solution

- Example:
  - Input: Graph G
  - Output: Independent Set S
  - Objective: Maximize |S|

- A β-approximation algorithm ($\beta \leq 1$):
  - If OPT is the optimal value output an answer $|S|$ with the *guarantee* that $|S| \geq \beta \cdot \mathrm{OPT}$

- Definition applies to all maximization problems: if ALG is a solution given by the algorithm, want $ALG \geq \beta \cdot \mathrm{OPT}$

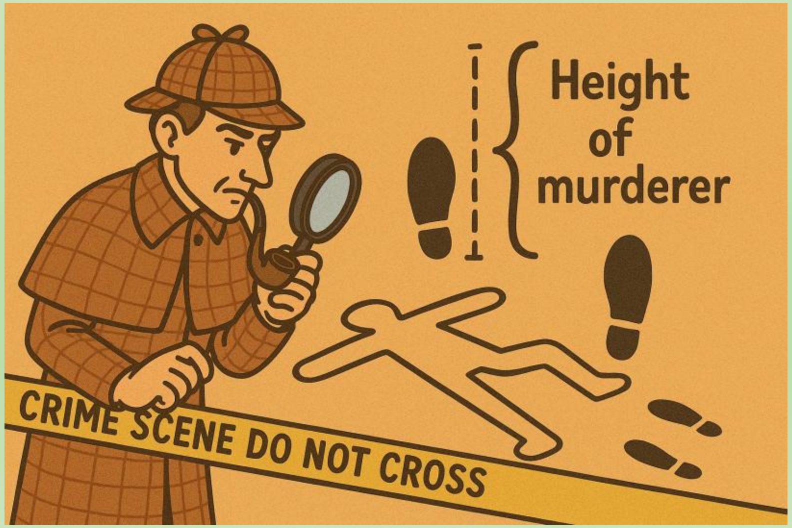
- For minimization problems, want to ensure that we get a solution not too larger than the optimal, i.e. $\mathrm{ALG} \leq \Delta \cdot \mathrm{OPT}$ (for $\Delta \geq 1$).

But how does it work?

- How can we guarantee that $\mathrm{ALG} \geq \beta \cdot \mathrm{OPT}$, when we have no idea what OPT will be?

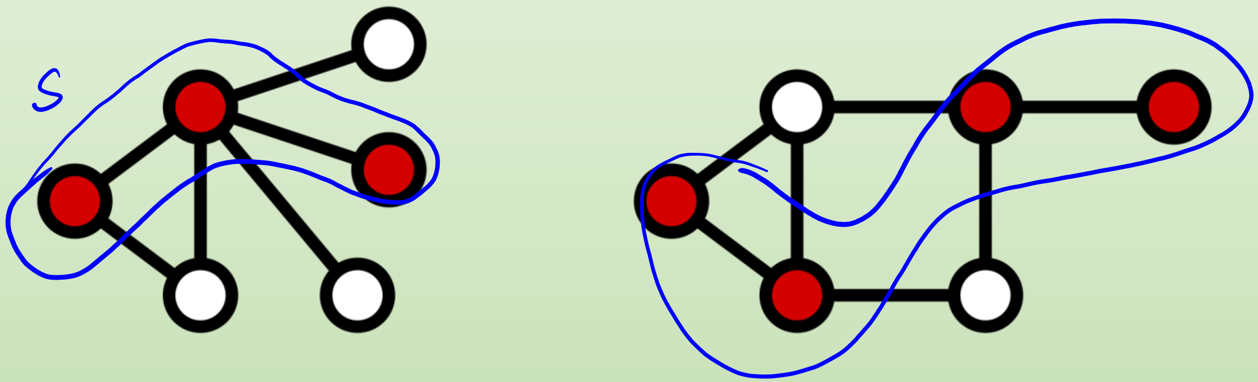
- *Answer:* Use a *proxy* to get an idea of what OPT should be like!

- Very similar to Lovasz's theta function!

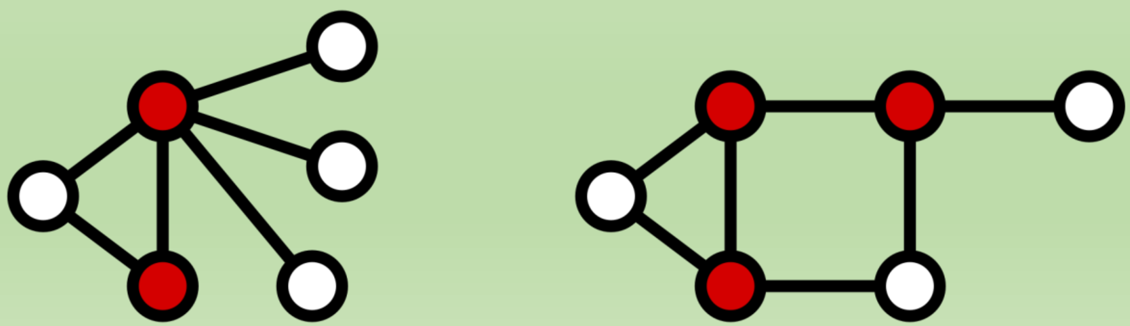


Example: Minimum Vertex-Cover

- A vertex cover is a subset S of vertices that covers (touches) all edges



- Goal: Find a minimum-sized vertex cover in input graph G



Optimization formulation and relaxation

for Vertex Cover

Given $G = (V, E)$, $V = \{1, 2, \ldots n\}$

Let variables be $x_1, x_2, \ldots x_n$.

$$Z = \min \sum_{i \in V} x_i$$

$$\text{s.t.} \quad x_i + x_j \geq 1 \quad \forall \{i,j\} \in E$$

$$x_i \in \{0,1\} \quad \forall i \in V$$

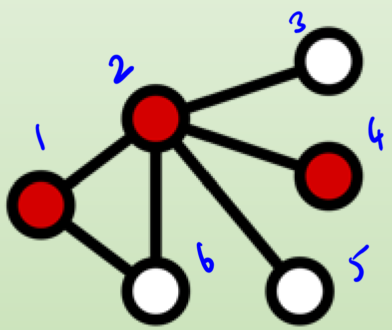
Any feasible soln $x \in \{0,1\}^n$ to this problem is an indicator vector of a vertex cover in G

$\Rightarrow$ optimal solution will be a minimum vertex cover in G

$$\min \quad x_1 + x_2 + x_3 + x_4 + x_5 + x_6 .$$

$$\text{s.t.} \quad \left.\begin{array}{l} x_1 + x_2 \geq 1 \\ x_1 + x_6 \geq 1 \\ x_2 + x_6 \geq 1 \\ x_2 + x_5 \geq 1 \\ x_2 + x_4 \geq 1 \\ x_2 + x_3 \geq 1 . \end{array}\right\}$$

$$x_i \in \{0,1\} \quad \forall i \in \{1, \ldots 6\}$$



Relaxation

$$Z_{LP} := \min \sum_{i \in V} x_i$$

$$\text{s.t.} \quad x_i + x_j \geq 1 \quad \forall \{i, j\} \in E$$

$$\left.\begin{array}{l} x_i \geq 0 \\ x_i \leq 1 \end{array}\right\} \forall i \in V.$$

$\longrightarrow$ LP, efficiently solvable.

Prev: $x_i \in \{0, 1\}$

$\rightarrow$ Output on solving need not be "integral".

x_i's may be fractional!

eg: $x^* = (0.2, \ 0.3, \ 0.47, \ 0.9, \ 0.6)$

Know. $\sum_{i \in V} x_i^* \leq OPT$ (i.e $Z_{LP} \leq OPT$)

Algorithm

$$\min \sum_{i \in V} x_i$$
$$s.t. \;\; x_i + x_j \geq 1 \quad \forall \{i,j\} \in E$$
$$x_i \geq 0 \quad \forall i \in V$$

① Feed the above into a solver, get a solution $x^* = (x_1^*, \ldots x_n^*)$
with value $Z_{LP}^* = \sum_{i \in V} x_i^* \leq OPT \hookrightarrow$ size of min vertex cover in G.

② Let $S = \{ i : x_i^* \geq \frac{1}{2} \}$.

<u>Claim</u>: S is a vertex cover in G.

<u>Proof</u>: Take any $\{i,j\} \in E$. We know that $x_i^* + x_j^* \geq 1$

$\Rightarrow$ atleast one endpoint on every edge is in S.

Why is $|S|$ small?

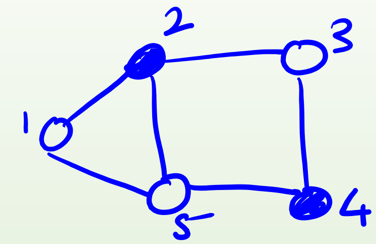
$$|S| = \sum_{i \in S} 1 \leq 2 \sum_{i \in S} x_i^* \quad \left(\text{since } x_i^* \geq \tfrac{1}{2} \quad \forall i \in S\right)$$

$$\leq 2 \left(\sum_{i \in V} x_i^*\right) \quad \left(\text{since } x_i^* \geq 0 \ \forall i\right)$$

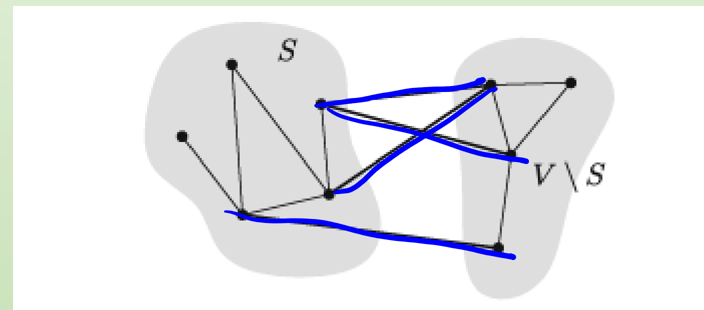
$$= Z_{LP}$$

$$\leq 2 \cdot \underbrace{OPT}_{\text{min vertex cover size.}} \quad \left(\text{since } Z_{LP} \leq OPT\right)$$

The MAX-CUT problem



- Input: $G = (V, E)$ $V = \{1, 2, \cdots n\}$
- Objective: Find a partition (S, S^c) of V that maximizes the number of edges across the cut: $\max\ |E(S, S^C)|$



- Unlike min-cut, NP Hard to find exactly!

- A $\frac{1}{2}$ - approximation was known
- Goemans and Williamson [1994] give an algorithm that guarantees a 0.878 approximation, which introduced the use of SDPs in designing algorithms.

An observation

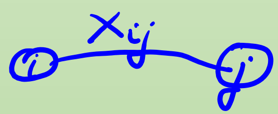
Fact: $\overbrace{OPT}^{\text{Max cut value}} \geq \frac{1}{2}|E|$ $\forall$ graphs G.

Proof: Choose a set $S \subseteq V$ randomly as follows:

For each $i \in V$: choose i to be in S w.p. $\frac{1}{2}$ (independently).

Will analyze: $\mathbb{E}[|E(S, S^c)|]$.

$$X_{ij} = \begin{cases} 1 & \text{if } i,j \text{ lie on opp sides of cut} \\ 0 & \text{if } i,j \text{ lie on same side of cut.} \end{cases}$$



$$\mathbb{E}[|E(S, S^c)|] = \mathbb{E}\Big[\sum_{i,j \in E} X_{ij}\Big]$$

$$= \sum_{\{i,j\} \in E} \mathbb{E}[X_{ij}] = \sum_{i,j \in E} \frac{1}{2}$$

$$= \frac{1}{2}|E|$$

Randomized algorithm:

Choose an S as above.

Output (S, S^c).

$$\mathbb{E}\left[|E(S,S^c)|\right] \geq \frac{1}{2}|E| \geq \frac{1}{2}\underbrace{OPT}_{\text{max-cut}}$$

Optimization formulation

- Linear formulations do not work very well for MAX-CUT

- A *quadratic formulation*

Variables $x_1 \cdots x_n$, x_i indicates if $i \in S$.

$$\max \sum_{\{i,j\} \in E} \left(\frac{1 - x_i x_j}{2} \right)$$

$$x_i^2 = 1 \equiv x_i \in \{-1, 1\} \quad \forall i \in V$$

$x_i = 1$ means $i \in S$

$x_i = -1$ means $i \notin S$

Claim: Optimum of this formulation is exactly MAX-CUT value of G.

Proof: Pick $\{i,j\} \in E$. $\frac{1 - x_i x_j}{2} = 1$ exactly when $x_i \neq x_j$

$= 0$ if $x_i = x_j$

Relaxation of quadratic form to a SDP

$$\max \sum_{\{i,j\} \in E} \left(\frac{1 - x_i x_j}{2} \right) \xrightarrow[\text{scalar variables by vectors}]{\text{replace}} \max \sum_{i,j \in E} \left(\frac{1 - u_i^T u_j}{2} \right)$$

$$\text{s.t.} \quad x_i^2 = 1 \quad \forall i \in V \qquad\qquad \text{s.t.} \quad \|u_i\|^2 = 1, \quad u_i \in \mathbb{R}^n.$$

Let Y = gram matrix of vectors $\{u_i\}_{i=1}^n$;

$$y_{ij} = u_i^T u_j$$

$$Z_{SDP} = \max \sum_{\{i,j\} \in E} \left(\frac{1 - y_{ij}}{2} \right)$$

$$\text{s.t.} \quad y_{ii} = 1, \quad Y \succeq 0$$

$$\boxed{Z_{SDP} \geq \text{MAX-CUT}(G)}$$

Finding a solution

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-\mathbf{u}_i^T\mathbf{u}_j}{2} \\ \text{subject to} & \mathbf{u}_i \in S^{n-1}, \quad i = 1, 2, \ldots, n. \end{array} \Bigg\} \text{SDP\#1}$$

$\hookrightarrow \|u_i\|^2 = 1$

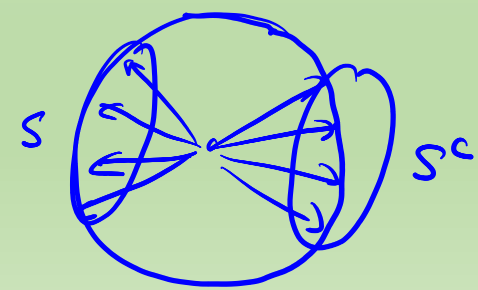
Algo:

① Write out SDP#1, feed it into a solver

Get solution $u_1^*, \ldots u_n^*$ in S^{n-1}

w/ objective function value $= Z_{SDP} \geq \text{MAX-CUT}(G)$.

Intuitively, expect if $\{i,j\} \in E$, u_i, u_j would have large angle between them.

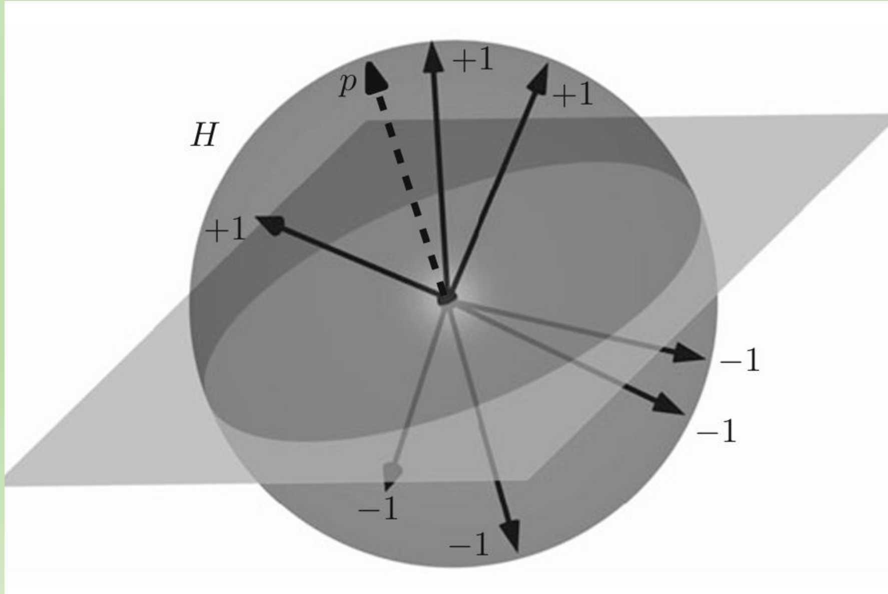


The randomized rounding algorithm

- Choose a $\underbrace{\text{random hyperplane}}_{\text{specify normal vector } p}$ through the origin that cuts S^{n-1}. Vectors $\underbrace{\text{on one side}}_{S=\{i:\, p^T u_i > 0\}}$ give the set S ($\underbrace{\text{and other side is}}_{S^c = \{j:\, p^T u_j \le 0\}}$ S^c).

Remaining : To bound $\mathbb{E}[|E(S,S^c)|]$

[GW'94] : $\mathbb{E}[|E(S,S^c)|] \geq 0.878$ MAX-CUT.

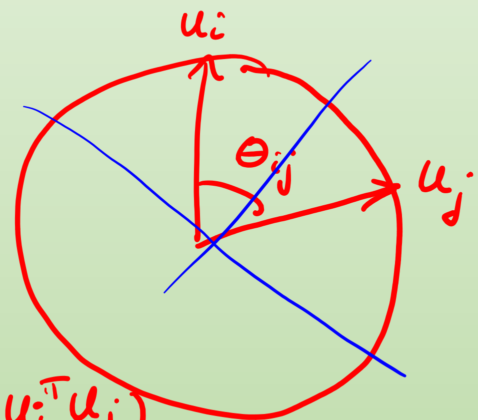


Analysis: Pick any edge $\{i,j\} \in E$

Let $X_{ij} = 1$, if i,j lie on opposite sides of the cut

$$\mathbb{E}\left[\,|E(S,S^c)|\,\right] = \sum_{i,j \in E} \mathbb{E}\left[X_{ij}\right].$$

i $\qquad$ j



$$\Pr(X_{ij}=1) = \frac{\cos^{-1}(u_i^T u_j)}{\pi}$$

$$= \frac{\theta_{ij}}{\pi}.$$

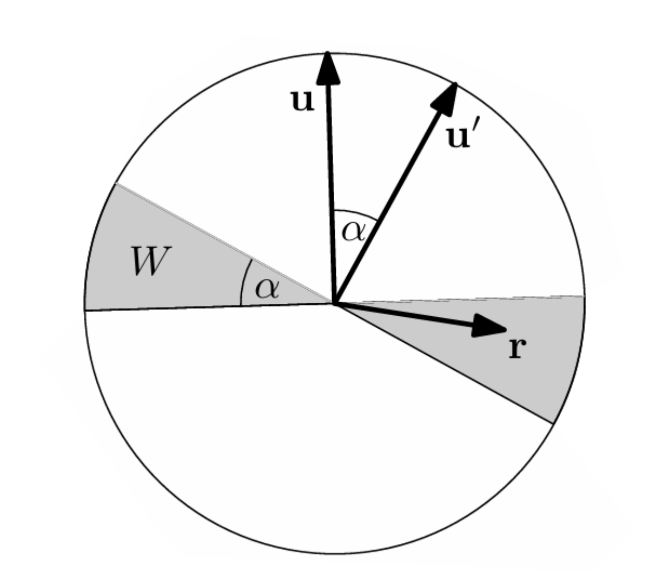
Main Lemma

- Let $u, u' \in S^{n-1}$. Then the probability that u, u' go to different halves is at least:

$$\frac{1}{\pi}\cos^{-1} u^T u'$$

(shown in prev slide)

Probability of separating u, u'



The final bound

- Lemma: $\frac{1}{\pi}\cos^{-1} z \geq 0.87856\ \frac{1-z}{2}$

Set $z = u_i^T u_j$

$\Rightarrow \Pr(x_{ij}=1) \geq 0.87856 \cdot \frac{(1-u_i^T u_j)}{2}$

$\Rightarrow \mathbb{E}[|E(S,S^c)|] = \sum_{i,j \in E} \mathbb{E}[x_{ij}] \geq 0.87856 \times \sum_{i,j \in E} \frac{(1-u_i^T u_j)}{2}$

$= 0.87856 \times Z_{SDP}$

$\geq 0.87856 \times \text{MAX-CUT}(G).$

Closing remarks

- The power of SDPs in approximation is a topic of very active research

References

Most of the material covered can be found in the excellent book:

- B. Gartner and J. Matousek, Approximation Algorithms and Semidefinite Programming, Springer, 2012