So far ...

Pre-PCP Theorem. — Edge Disjoint Paths

PCP Theorem (proof)

post-PCP Theorem — Håstad's 3-bit PCP / 3Lin

Now ---- — Max-Cut / Unique Games Conjecture

Max-Cut

Goemans-Williamson's Algorithm

[Semi-definite Programming]

Unique Games Conjecture

$\Rightarrow$ GW is optimal!

[Majority is Stablest].

Semi-Definite Programming (Quick primer)

LP (linear pgm)

- Real variables $x_1, x_2, \ldots, x_n \in \mathbb{R}$
- Constraints (linear)
$$\left.\begin{array}{c}1\\2\\ \vdots\\ m\end{array}\right\} \quad \sum_{i=1}^{n} a_i x_i \geq b \qquad \text{(polytope)}$$
- Maximize
$$\sum_{i=1}^{n} c_i x_i \qquad (\text{linear } f^n)$$

SDP (semi-definite pgm)

- Vector valued variables $v_1, v_2, \ldots, v_n \in \mathbb{R}^n$
- Constraints (linear on pairwise dot products)
$$\left.\begin{array}{c}1\\2\\ \vdots\\ m\end{array}\right\} \quad \sum_{i,j=1}^{n} a_{ij}\, v_i \cdot v_j \geq b$$
- Maximize
$$\sum_{i,j=1}^{n} c_{ij}\, v_i \cdot v_j$$

(IMP: we cannot restrict the dimension $\mathbb{R}^n$)

Theorem

- LP can be solved in polynomial time.
- SDP can be solved in polynomial time.

(convex prog, ellipsoid method)

SDP is an LP in disguise.

Let
$$X = \begin{array}{cc} & \quad\quad j \\ i & \begin{bmatrix} & \vdots & \\ \cdots & v_i \cdot v_j & \\ & & \end{bmatrix}_{n \times n} \end{array} = \begin{array}{cc} & \quad\quad j \\ i & \begin{bmatrix} & \vdots & \\ \cdots & x_{ij} & \\ & & \end{bmatrix} \end{array}$$

- Think of $\{x_{ij}\}_{i,j=1}^{n}$ as real variables.
- Maximize $\sum_{i,j=1}^{n} c_{ij} x_{ij}$

  Constraints
$$\begin{array}{c} 1 \\ 2 \\ \vdots \\ m \end{array} \left\{ \sum_{i,j=1}^{n} a_{ij} x_{ij} \geq b \right.$$

- and X is a Positive Semi-Definite matrix.

Def A symmetric $n \times n$ matrix X is called positive semidefinite if any of these equivalent conditions holds:

① $a^t X \cdot a \geq 0 \quad \forall a \in \mathbb{R}^n$

$$[\ a^t\ ]\begin{bmatrix} & & \\ & X & \\ & & \end{bmatrix}\begin{bmatrix} \\ a \\ \\ \end{bmatrix} \geq 0$$

② All eigenvalues of X are ≥ 0.

③ $x_{ij} = v_i \cdot v_j$ for some $v_1, v_2, \ldots, v_n \in \mathbb{R}^n$.

Note - X is PSD can be thought of as a set of (infinitely many) linear constraints

$a^t X a \geq 0$, one constraint for every a.

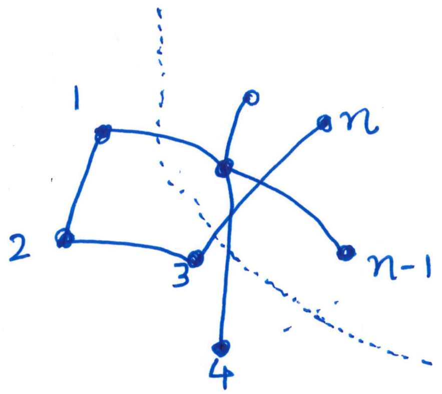
- If X is not PSD, one can find, in polytime, $\tilde{a}$ s.t. $\tilde{a}^t X \tilde{a} < 0$ (eigenvector with -ve eigenvalue).

- $\Rightarrow$ "separation oracle" $\Rightarrow$ ellipsoid algorithm.

▨

MAX-CUT

Given a graph $G(V,E)$, find a cut of max size.



$V = \{1, 2, \dots, n\}$.

Integer program Let $x_i = \begin{cases} +1 & \text{if } i \in \text{LHS} \\ -1 & \text{if } i \in \text{RHS} \end{cases}$

$$OPT(G) = \text{Max} \sum_{(i,j) \in E} \frac{1 - x_i x_j}{2}$$

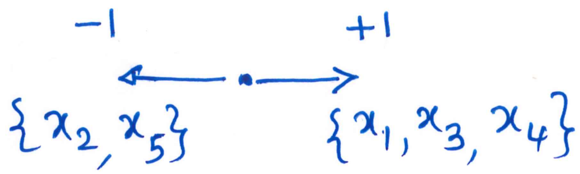
s.t. $\quad x_i \in \{-1, 1\} \quad \forall \ 1 \le i \le n$.

- NP-hard.
- $\frac{1}{2}$-approx trivial.
- [GW]. α_{GW}-approx, $\quad \alpha_{GW} \approx 0.878$

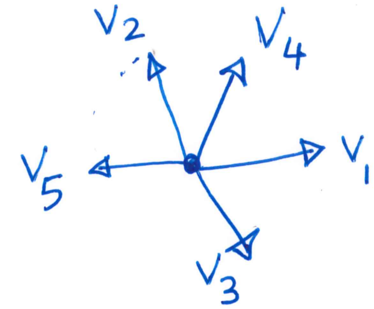
via SDP-relaxation.

$\downarrow$

Allow $x_i \in \{-1, 1\}$ to be unit vector $v_i \in \mathbb{R}^n$



Relax $\rightsquigarrow$



$$OPT(G) = \max \sum_{(i,j)\in E} \frac{1 - x_i x_j}{2} \quad \rightsquigarrow \quad SDP(G) = \max \sum_{(i,j)\in E} \frac{1 - V_i \cdot V_j}{2}$$

s.t. $x_i \in \{-1, 1\} \ \forall i$. s.t. $V_i \cdot V_i = 1 \ \forall i$

$$\|V_i\| = 1$$

Can be solved in polytime.

<u>Note</u>. $OPT(G) \le SDP(G)$. (relax^n)

$\alpha_{GW} \cdot SDP(G) \le OPT(G)$ (to prove).

<u>Rounding method</u>

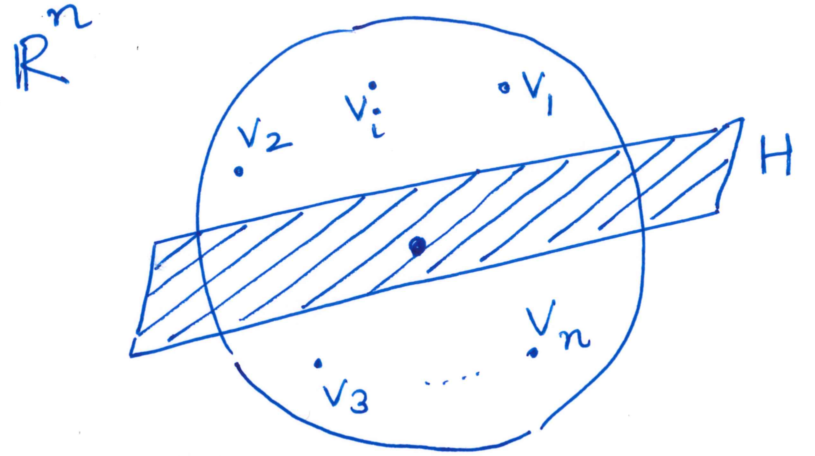
- $\{V_1, V_2, \ldots, V_n\} \longrightarrow \{+1, -1\}$ $V_i \to x_i$
- $SDP(G) \longrightarrow \sum_{(i,j)\in E} \frac{1 - x_i x_j}{2} \ge \alpha_{GW} \cdot SDP(G)$.
- Random hyperplane rounding!

Given

SDP. solution to

$$\text{Max} \sum_{(i,j) \in E} \frac{1 - v_i \cdot v_j}{2}$$

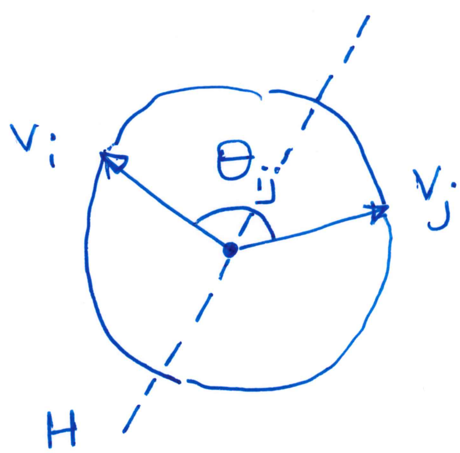
$$\text{s.t.} \quad \|v_i\| = 1 \quad \forall\, i.$$



- Pick a random hyperplane H thru origin.
- $v_i \longrightarrow x_i = \begin{cases} +1 & \text{if } v_i \text{ is on LHS of H} \\ -1 & \text{if } v_i \quad \text{"} \quad \text{RHS} \quad \text{"} \end{cases}$

Claim. $\mathbb{E}\left[\frac{1 - x_i x_j}{2}\right] \geqslant \alpha_{GW} \cdot \frac{1 - v_i \cdot v_j}{2}$ $\qquad \alpha_{GW} \approx 0.878$

Proof



$$v_i \cdot v_j = \cos \theta_{ij}.$$

$$\Pr[\text{H separates } v_i, v_j] = \frac{\theta_{ij}}{\pi}.$$

$$\therefore \mathbb{E}\left[\frac{1-x_i x_j}{2}\right] = \Pr[H \text{ separates } v_i, v_j]$$

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

$$\geq \alpha_{GW} \cdot \frac{1-\cos\theta_{ij}}{2}$$

$$= \alpha_{GW} \cdot \frac{1 - v_i \cdot v_j}{2}$$

where

$$\alpha_{GW} = \min_{0 \leq \theta \leq \pi} \frac{\theta/\pi}{(1-\cos\theta)/2}$$

$$\approx 0.878 \quad \text{at} \quad \theta_c \approx 117^\circ.$$

If CUT denotes the size of the cut output,

$$\mathbb{E}[\text{CUT}] = \mathbb{E}\left[\sum_{(i,j)\in E} \frac{1-x_i x_j}{2}\right]$$

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

$$\geq \alpha_{GW} \cdot \sum_{(i,j)\in E} \frac{1 - v_i \cdot v_j}{2}$$

$$= \alpha_{GW} \cdot \text{SDP}(G). \qquad \blacksquare$$