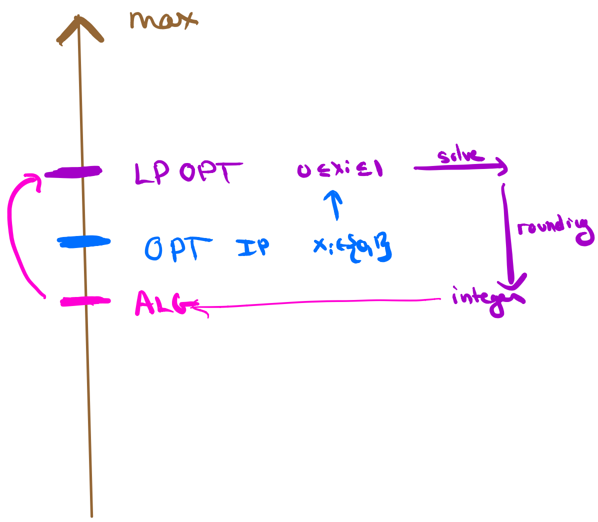
## Today

Randomized rounding of SDPs
- MAX CUT
- 3-coloring




$$\frac{\text{LP OPT}}{\text{ALG}} \le \alpha$$

worst case ratio

[ $\frac{\text{LP OPT}}{\text{OPT}}$ called integrality gap.

## MAXCUT

Input: $G=(V,E)$ $\quad w_{ij} \quad \forall (i,j)\in E$

Goal: partition vertex set so as to max weight of endpts crossing cut.

## IP formulation of MAXCUT

$$x_i = \begin{cases} 0 & \text{on one side of partition} \\ 1 & \text{on other side} \end{cases}$$

$$z_{ij} = \begin{cases} 1 & \text{edge } (i,j) \text{ cut} \\ 0 & \text{o.w.} \end{cases}$$

$$\max \sum_{(i,j)\in E} w_{ij} z_{ij}$$

$z_{ij} \le x_i + x_j \quad \forall (i,j)\in E$ $\qquad x_i \ne x_j \Rightarrow z_{ij} \le 1$

$z_{ij} \le 2-(x_i+x_j) \quad \forall (i,j)\in E$ $\qquad x_i = x_j \Rightarrow z_{ij} = 0$

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

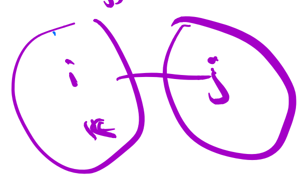
$z_{ij} \in \{0,1\} \quad \forall (i,j)\in E$

$0 \le x_i \le 1$

$0 \le z_{ij} \le 1$

LP OPT $\sum_{(i,j)\in E} w_{ij}$

$$\underbrace{z_{ij}}_{0} \le \underbrace{z_{ik}}_{0} + \underbrace{z_{kj}}_{0}$$




$*$ no polynomial sized LP relaxation of MAXCUT has integrality gap $> \frac{1}{2}$.

Another approach:

First, notation change

$\forall i.$ $\quad x_i \in \{-1,+1\}$

define $\boxed{y_{ij} = x_i x_j}$ $\quad \forall i,j \in V$

$\Rightarrow \quad y_{ij} = \begin{cases} 1 & \text{if } x_i = x_j \\ -1 & x_i \neq x_j \end{cases}$

$$\max \sum_{(i,j)\in E} w_{ij} \underbrace{\mathbb{1}[x_i \neq x_j]}_{\frac{1}{2}(1-y_{ij})}$$

Exactly captures MAX CUT!

$y_{ij} = y_{ji}$

$y_{ii} = 1$

Want $\exists\, x_i \;\; \forall i \in V$

s.t. $\quad y_{ij} = x_i x_j \quad \forall i,j)$

$x_i \in \{-1,1\}$

$$\max \sum_{(i,j)\in E} w_{ij} \tfrac{1}{2}(1-y_{ij})$$

$y_{ij} = y_{ji} \qquad \forall i,j \in V$

$y_{ii} = 1 \qquad \forall i \in V$

$\forall \vec{c} \quad \vec{c}^{\,T} Y c \geq 0 \equiv Y \succeq 0$

psd.

Idea: enforce brown by adding linear inequalities to purple.

Examples: $\left(\sum_{i=1}^n x_i\right)^2 \geq 0$ $\quad \sum_{i=1}^n \sum_{j=1}^n \underbrace{x_i x_j}_{y_{ij}} = \boxed{\sum_{i=1}^n \sum_{j=1}^n y_{ij} \geq 0}$

$(x_1 - x_2)^2 \geq 0 \qquad x_1x_1 - x_1x_2 - x_2x_1 + x_2x_2 \geq 0$

$\qquad 1 \quad - 2y_{12} \qquad\qquad 1$

$\qquad 2 - 2y_{12} \geq 0$

$\forall\, (c_1, c_2, \dots, c_n) = \vec{c}$

$\left(\sum_{i=1}^n c_i x_i\right)^2 \geq 0 \qquad \sum_i \sum_j c_i c_j x_i x_j = \sum_i \sum_j c_i c_j y_{ij} \geq 0$

$(c_1 \; \dots \; c_n) \begin{pmatrix} y_{11} & & y_{1n} \\ & y_{ij} & \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \end{pmatrix} \geq 0 \quad \forall \vec{c}$ $\quad Y$ is psd.

$\begin{pmatrix} y_{n1} & \cdots & y_{nn} \end{pmatrix} \begin{pmatrix} c_n \end{pmatrix}$

$\boxed{\text{SDP rounding}}$

Intro to semi-definite programming

linear programming where vars are entries in psd matrix

$\boxed{\text{Defn}}$ If $A$ is a symmetric $n$ by $n$ matrix then $A$ is a positive semidefinite (psd) matrix $\equiv A \succeq 0$ iff any of the following equivalent conditions hold

1. $\forall \vec{c} \in \mathbb{R}^n, \quad c^T A c \geq 0$
2. $\Rightarrow$ $A$ has nonnegative eigenvalues
3. $A = V^T V$ for some $m \times n$ matrix $V$, $m \leq n$
4. $A = \sum_{i=1}^{n} \lambda_i x_i x_i^T$ for some $\lambda_i \geq 0$ and orthonormal vectors $x_i \in \mathbb{R}^n$

Spectral Thm: real symmetric matrix
$n$ real eigenvalues $\lambda_1 \ldots \lambda_n$
orthnormal eigenvectors $x_1 \quad x_n$
$Ax_i = \lambda_i x_i$

$$A = \underbrace{\begin{bmatrix} | & & | \\ x_1 & \cdots & x_n \\ | & & | \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ 0 & & \ddots \ \lambda_n \end{bmatrix}}_{\Lambda} \underbrace{\begin{bmatrix} — x_1 — \\ — x_2 — \\ \vdots \\ — x_n — \end{bmatrix}}_{U^T}$$

$$A = \underbrace{U \Lambda^{1/2}}_{V^T} \underbrace{\Lambda^{1/2} U^T}_{V} \Leftarrow$$

$\forall c \quad c^T A c = c^T V^T V c = \|Vc\|^2 \geq 0$

$x \qquad x^T A x = \lambda x^T x < 0$
$\lambda < 0$

$$A = V^T V = \begin{pmatrix} — v_1 — \\ \vdots \\ — v_n — \end{pmatrix} \begin{pmatrix} | & & | \\ v_1 & \cdots & v_n \\ | & & | \end{pmatrix}$$

$A_{ij} = v_i \cdot v_j$

## SDP rounding

### Intro to semi-definite programming

linear programming where vars are entries in psd matrix

**Defn** If $A$ is a symmetric $n$ by $n$ matrix
then $A$ is a positive semidefinite (psd) matrix $\equiv A \succeq 0$
if any of the following equivalent conditions hold

1. $\forall \vec{c} \in \mathbb{R}^n$, $c^T A c \geq 0$
2. $A$ has nonnegative eigenvalues
3. $A = V^T V$ for some $m \times n$ matrix $V$, $m \leq n$
4. $A = \sum_{i=1}^{n} \lambda_i x_i x_i^T$ for some $\lambda_i \geq 0$ and orthonormal vectors $x_i \in \mathbb{R}^n$

### Semidefinite program (SDP)

max or min $\sum_{i,j} c_{ij} x_{ij}$

subject to $\sum_{i,j} a_{ijk} x_{ij} = b_k$

$x_{ij} = x_{ji}$ $\forall i,j$

$X = (x_{ij}) \succeq 0$

$\equiv$ Vector program

max or min $\sum_{i,j} c_{ij} (v_i \cdot v_j)$

subject to $\sum_{i,j} a_{ijk} (v_i \cdot v_j) = b_k$

$v_i \in \mathbb{R}^n$ $i = 1, \ldots, n$

given $X \Rightarrow X = V^T V$,
set $v_i$ to be $i^{th}$ col of $V$

### Key fact:

SDPs can be solved to within additive error $\varepsilon$
in time poly(size of input, $\log(\frac{1}{\varepsilon})$)

in our discussions, we ignore additive error $\varepsilon$

Recap:

① 

$$\boxed{\begin{array}{ll} \text{Want } \exists\, x_i & \forall i \in V \\ \text{s.t. } y_{ij} = x_i x_j & \forall i,j \end{array}}$$

$$\boxed{\begin{array}{lll} \max & \sum_{(i,j)\in E} w_{ij} \tfrac{1}{2}(1-y_{ij}) & \\ & y_{ij} = y_{ji} & \forall i,j \in V \\ & y_{ii} = 1 & \forall i \in V \end{array}}$$

Opt solution to brown + purple = Opt of MAXCUT

② Brown $\Rightarrow$

$$(c_1, c_2, \dots, c_n) \underbrace{\begin{pmatrix} \cdot & \cdots & \cdot \\ \vdots & y_{ij} & \\ \cdot & \cdots & \cdot \end{pmatrix}}_{Y} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} \geq 0 \quad \forall \vec{c} \in \mathbb{R}^n$$

These constraints $c^T Y c \geq 0 \ \forall c \in \mathbb{R}^n$

$\equiv$ $Y$ is psd matrix!

③ Yields a semidefinite programming relaxation of MAXCUT

$$\boxed{\begin{array}{lll} \max & \sum_{(i,j)\in E} w_{ij} \tfrac{1}{2}(1-y_{ij}) & \\ & y_{ij} = y_{ji} & \forall i,j \in V \\ & y_{ii} = 1 & \forall i \in V \\ \text{plus} & \boxed{Y = \begin{pmatrix} y_{11} & \cdots & y_{1n} \\ y_{n1} & \cdots & y_{nn} \end{pmatrix} \ \text{psd}} & \end{array}}$$

can be solved eficiently using the ellipsoid alg. up to error $\varepsilon$.

We can solve this, "round" results $\Rightarrow$ int soln

$\Rightarrow$ prove that it gives pretty good approx.

Can equivalently write SDP relaxation as a vector program

$$\max \sum_{(i,j)\in E} w_{ij} \tfrac{1}{2}(1-y_{ij})$$

$y_{ij} = y_{ji} \quad \forall i,j \in V$

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

$Y = \begin{pmatrix} y_{11} & \cdots & y_{1n} \\ y_{n1} & \cdots & y_{nn} \end{pmatrix}$ psd

$$\max \sum_{(i,j)\in E} w_{ij} \tfrac{1}{2}(1 - v_i \cdot v_j)$$

$v_i \cdot v_i = 1 \quad \Leftarrow$

$v_i \in \mathbb{R}^n \quad \forall_{1 \le i \le n}$

$\Uparrow$

Relaxation check:

the $OPT$ for MAXCUT $\le OPT_{SDP}(G)$

suppose $(S, \bar{S})$ achieves OPT

feasible soln to SDP $v_i = \begin{cases} (1,0,0\ldots 0) & i \in S \\ (-1,0,0\ldots 0) & i \notin S \end{cases}$

$v_i \cdot v_j = \cos(\text{angle betw}^{n} \ v_i \,\&\, v_j)$

$$\max \sum_{(i,j)\in E} w_{ij} \tfrac{1}{2}(1 - \vec{v}_i \cdot \vec{v}_j)$$

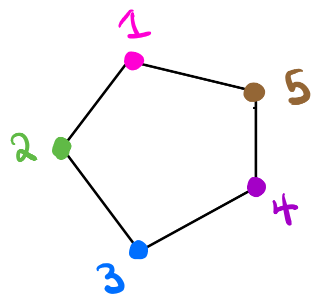
$v_i \cdot v_i = 1$

i.e. $v_i$'s are unit vectors $\in \mathbb{R}^n$

$\rightarrow \tfrac{1}{2}(1 - \cos(\text{angle}(v_i, v_j)))$

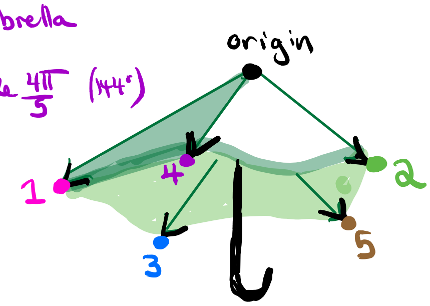
would like $v_i = -v_j$

Example




Lovász Umbrella

all edges at angle $\frac{4\pi}{5}$ (144°)




$$\max \sum_{(i,j)\in E} w_{ij} \underbrace{\tfrac{1}{2}\left(1-\cos\left(\text{angle}\,(v_i,v_j)\right)\right)}_{\approx\, 0.9}$$

all weights equal

OPT = 4

SDP OPT $\approx 4.5$

ratio $\approx \frac{4}{4.5} \approx 0.89$

Lovász umbrella

all edges at angle $\frac{4\pi}{5}$

144°

$$\cos\left(\frac{4\pi}{5}\right) = -\frac{\phi}{2} \approx -.8$$

golden ratio $\frac{1+\sqrt{5}}{2}$

# MAXCUT

Input: $G=(V,E)$ $\quad w_{ij} \quad \forall (i,j)\in E$

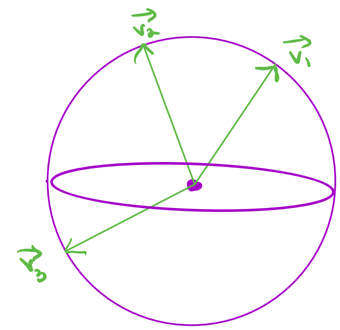
Goal: partition vertex set so as to max weight of endpts crossing cut.

## Vector programming relaxation

$$\max \quad \frac{1}{2}\sum_{(i,j)\in E} w_{ij}\,(1-\vec{v}_i\cdot\vec{v}_j)$$

$$\vec{v}_i\cdot\vec{v}_i = 1 \quad \forall i\in V$$

$$\vec{v}_i \in \mathbb{R}^n$$



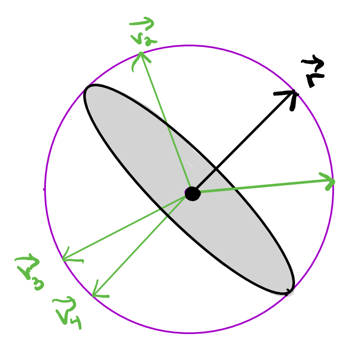

Can solve SDP in poly time.

$\boxed{\text{Claim:}}$ MAXCUT OPT $\leq$ SDP OPT

But how to round? get large contribution to OPT when $v_i \cdot v_j$ very -ve

## Random hyperplane rounding

Solve SDP $\longrightarrow v_1^*, v_2^*, \ldots, v_n^*$
pick random hyperplane thru origin
partition vertices based on which side of hyperplane

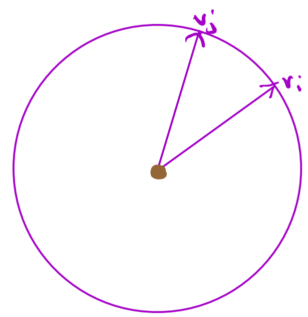



Put $i$ into $S$ if $v_i \cdot \vec{r} \geq 0$
$i$ onto $\bar{S}$ if $v_i \cdot \vec{r} < 0$

$\Rightarrow \boxed{(z_1, \ldots, z_n)}$
$z_i \sim N(0,1)$

$$E(\text{weight of resulting cut } (S,\bar{S})) = \sum_{(i,j)\in E} w_{ij} \Pr((i,j) \text{ gets cut})$$

$(i,j)$

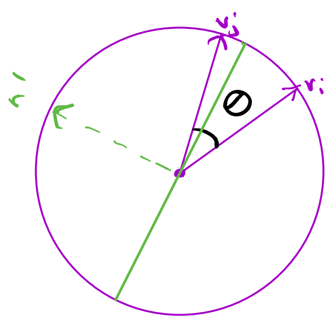



$\vec{r} = \vec{r}' + \vec{r}''$

↑ projection of r to this plane

↑ orthogonal.

$\frac{r'}{\|r'\|}$ is uniformly dist'd around circle




$\Pr((i,j) \text{ gets cut})$

$= \Pr(\text{unif random diam cuts between them})$

$$= \frac{\theta}{\pi} = \frac{\arccos(\vec{v}_i \cdot \vec{v}_j)}{\pi}$$

$$E(\text{weight of resulting cut } (S,\bar{S})) = \sum_{(i,j)\in E} w_{ij} \Pr((i,j) \text{ gets cut})$$

$$= \sum_{(i,j)\in E} w_{ij} \frac{\arccos(\vec{v}_i \cdot \vec{v}_j)}{\pi} \geq \alpha \underbrace{\sum_{(i,j)\in E} w_{ij} \frac{(1 - v_i \cdot v_j)}{2}}_{\text{obj of SDP}}$$

$$\forall x \; -1 \leq x \leq 1 \quad \frac{\frac{\arccos(x)}{\pi}}{\frac{1-x}{2}} \geq \alpha$$

$$\alpha \geq \min_{-1\leq x\leq 1} \frac{\frac{1}{\pi}\arccos(x)}{\frac{1}{2}(1-x)} \geq 0.878\ldots$$

$$E\left(\text{wt of cut produced by alg}\right) \geq 0.878 \; OPT_{SDP} \geq 0.878 \; OPT$$

## Hardness

① If $\exists$ approx alg for MAX CUT with approx ratio $\geq 0.941$, then $P = NP$.

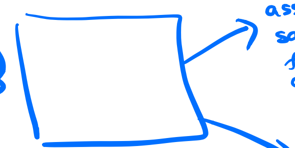
② If the "unique games conjecture" is true, there is no approx alg for MAXCUT with approx ratio better than 0.878....

### Unique Games Conjecture

$\forall \varepsilon > 0$, $\exists$ prime $q$ s.t.

$(1-\varepsilon, \varepsilon)$ Gap version

MAX2LIN(q) instance



if $\exists$ assignment satisfying $\geq 1-\varepsilon$ fraction of eqns output Yes

if no assignment satisfies $> \varepsilon$ fraction output NO

admits no poly-time soln unless $P = NP$

### MAX2LIN(q)

$q$ prime

input: linear equations mod $q$ w/ unknowns $x_1, \ldots, x_n \in \{0, 1, \ldots, q-1\}$

(form $x_i - x_j = c$)

$x_3 - x_{11} \equiv 87 \pmod{97}$

$x_7 - x_{22} \equiv 3 \pmod{97}$

$\vdots$

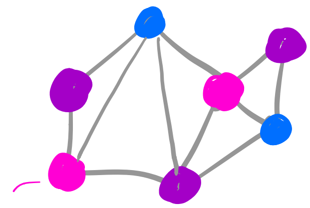
$x_7 - x_9 \equiv 56 \pmod{97}$

Problem: Find assignment of $x_i$'s that satisfies max possible # of eqns

③ Int gap of the [GW] SDP $= 0.878\ldots$

③ Every poly sized LP relaxation of MAXCUT has integrality gap of $\frac{1}{2}$. Janes

# 3-Coloring a 3-colorable graph



Given graph $G=(V,E)$
& promise that it is 3-colorable

What is min $k$ s.t. we can find a $k$-coloring of $G$ in poly time?

**Simple results:**

① A graph with max degree $\Delta$ can be colored with $\leq \Delta+1$ colors

② A 3-colorable graph can be colored with $O(\sqrt{n})$ colors.

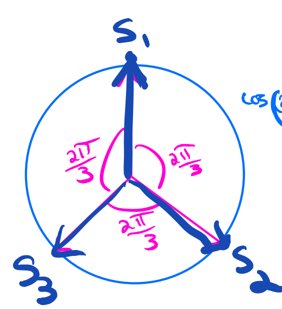
Find a vertex of deg $\geq \sqrt{n}$
Use 3 colors to color it & its neighbors (neighborhood 2-colorable)
Remove it & its neighbors from graph

## An SDP-based alg

$$\begin{aligned} \min \quad & \lambda & \\ \text{s.t.} \quad & \vec{v}_i \cdot \vec{v}_j \leq \lambda & \forall (i,j) \in E \\ & \vec{v}_i \cdot \vec{v}_i = 1 & \forall i \in V \\ & \vec{v}_i \in \mathbb{R}^n & \end{aligned}$$

**Claim:**
if graph is 3-colorable
$\lambda \leq -\frac{1}{2}$


$\Longrightarrow$




$\cos\left(\frac{2\pi}{3}\right) = -\frac{1}{2}$

$$\min \lambda$$

s.t. $\vec{v}_i \cdot \vec{v}_j \le \lambda \quad \forall (i,j) \in E$

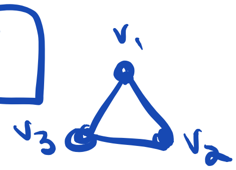
$\vec{v}_i \cdot \vec{v}_i = 1 \quad \forall i$

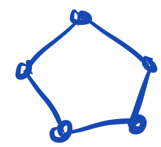
$\vec{v}_i \in \mathbb{R}^n \quad \forall i$

(*)

$\boxed{\text{vector } \vec{v}_i \in \mathbb{R}^n \text{ for each vertex } i \in V}$

**Claim:** if graph is 3-colorable $\lambda \le -\frac{1}{2}$



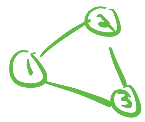
requires 3 colors.

Aside: If G has a triangle, then optimal soln to SDP has $\lambda^* \ge -\frac{1}{2}$

cliques on k vertices = chromatic # k

perfect graphs.

Proof: Suppose




$$0 \le (\vec{v}_1+\vec{v}_2+\vec{v}_3, \vec{v}_1+\vec{v}_2+\vec{v}_3) = \vec{v}_1\cdot\vec{v}_1 + \vec{v}_2\cdot\vec{v}_2 + \vec{v}_3\cdot\vec{v}_3 + \vec{v}_1\cdot\vec{v}_2 + \vec{v}_1\cdot\vec{v}_3 + \vec{v}_2\cdot\vec{v}_1 + \vec{v}_2\cdot\vec{v}_3 + \vec{v}_3\cdot\vec{v}_1 + \vec{v}_3\cdot\vec{v}_2$$

## Algorithm

1. Solve SDP (*) $\Rightarrow v_i^*$ $i=1,\dots,n$
2. Choose $t$ random hyperplanes thru origin
3. Color vertices in each region w/ diff color
4. remove any edges properly colored
5. Repeat steps 2-4 until have proper coloring