Maxcut

The Maxcut problem

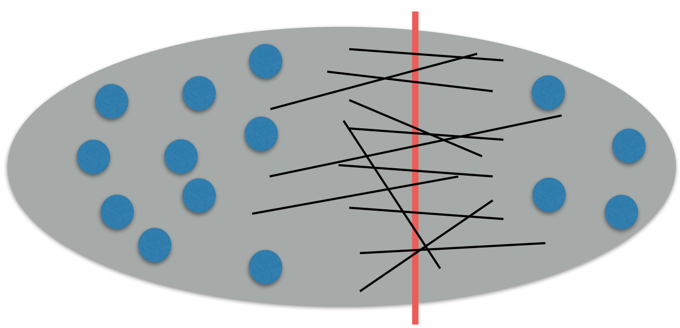
Given:

a graph with non-negative edge weights

Find:

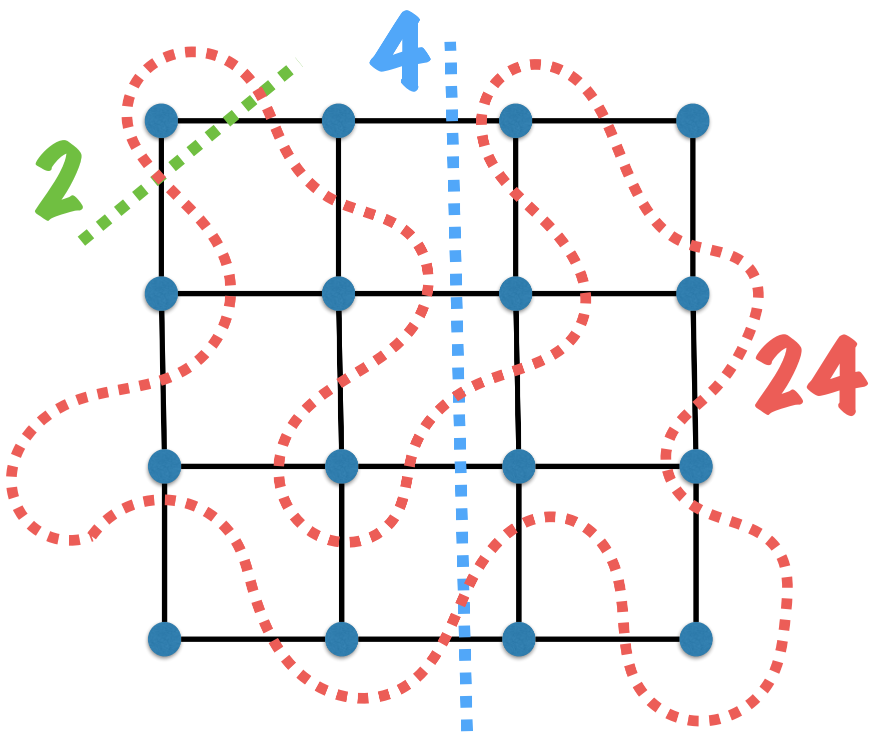
a partition of the vertices into two sets that maximizes:

the weight of edges across the resulting cut



Example

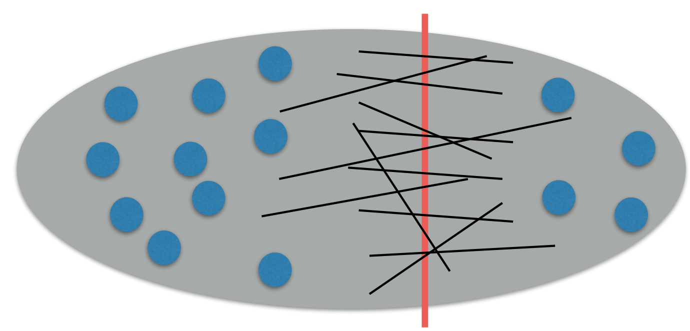
If every edge has weight 1



Maxcut

Given: a graph with non-negative edge weights
Find:a partition of the vertices into two sets
that maximizes:
the weight of edges across the resulting cut

- Does not have to separate a particular pair {s,t}
- Does not have to be balanced



Maxcut **Mincut**

Given: graph, non-negative edge weights

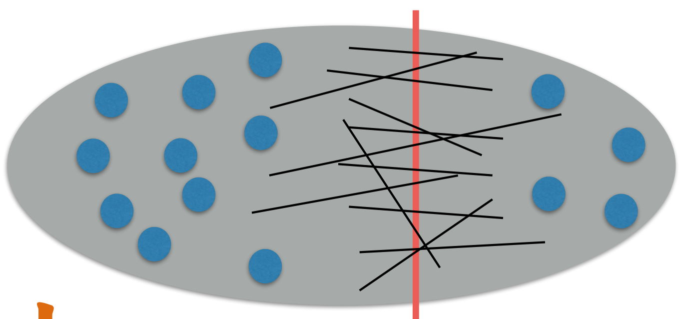
and two vertices s and t

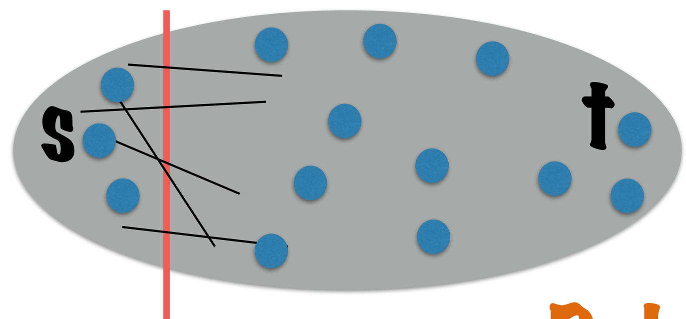
Find: partition of vertices into two sets,

separating s from t

Maximizing Minimizing

weight of edges across the resulting cut





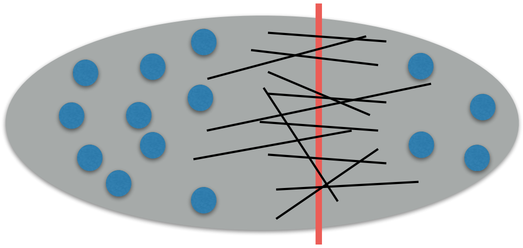
NP-hard Polynomial

Maxcut

Maxcut

Partition graph
to maximize
edge weight across cut

Simple algorithm: Random



S = random subset
each vertex goes into S
independently w.pr. .5

Output cut (S, V-S)

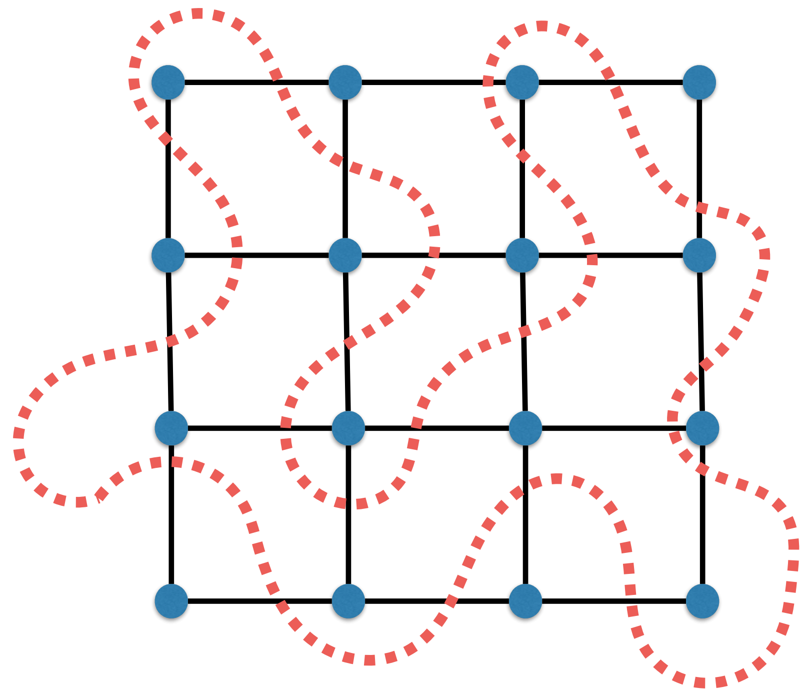
Theorem: it's a 2-approximation

Partition graph
to maximize
edge weight across cut

Analysis: bounding OPT

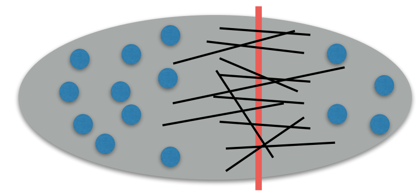
Upper bound on OPT:

$$\mathbf{OPT} \leq \sum_{\{\mathbf{i},\mathbf{j}\} \in \mathbf{E}} \mathbf{w}_{\mathbf{ij}}$$



Partition graph to maximize edge weight across cut

Analysis: bounding output



$$\mathbf{Output} = \sum_{\{\mathbf{i},\mathbf{j}\}\in\mathbf{E}} \mathbf{w_{ij}} \cdot \mathbf{1}(\{\mathbf{i},\mathbf{j}\}\ \mathbf{crosses\ cut})$$

$$\mathbf{E}(\mathbf{Output}) = \sum_{\{\mathbf{i},\mathbf{j}\}\in\mathbf{E}} \mathbf{w_{ij}} \cdot \Pr(\{\mathbf{i},\mathbf{j}\}\ \mathbf{crosses\ cut})$$

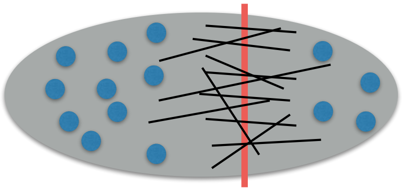
whether {i,j} crosses & probability	i in S .5	i not in S .5
j in S .5	no p=.25	yes p=.25
j not in S .5	yes p=.25	no p=.25

$$\Pr(\{\mathbf{i},\mathbf{j}\}\ \mathbf{crosses\ cut}) = \mathbf{.25} + \mathbf{.25} = \mathbf{.5}$$

$$\mathbf{E}(\mathbf{Output}) = (\mathbf{1}/\mathbf{2}) \sum_{\{\mathbf{i},\mathbf{j}\}\in\mathbf{E}} \mathbf{w_{ij}}$$

Partition graph
to maximize
edge weight across cut

Together



$$\mathbf{OPT} \leq \sum_{\{\mathbf{i},\mathbf{j}\}\in \mathbf{E}} \mathbf{w_{ij}}$$

$$\mathbf{E}(\mathbf{Output}) = (\mathbf{1}/\mathbf{2}) \sum_{\{\mathbf{i},\mathbf{j}\}\in \mathbf{E}} \mathbf{w_{ij}}$$

$$\mathbf{E}(\mathbf{Output}) \geq (\mathbf{1}/\mathbf{2})\mathbf{OPT}$$

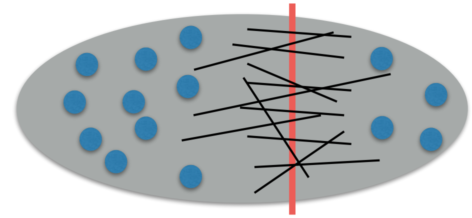
QED

Maxcut

Maxcut

Partition graph
to maximize
edge weight across cut

Can we do better than 2?



Integer programming formulation

One variable per edge:

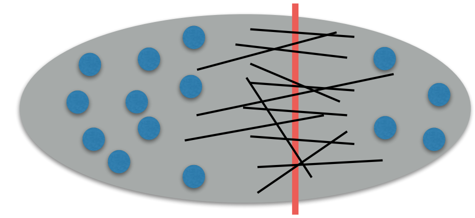
$$\mathbf{x_{ij}} = \begin{cases} 1 & \textbf{if crosses cut} \\ 0 & \textbf{otherwise} \end{cases}$$

Objective: $$\max \sum_{\{\mathbf{i},\mathbf{j}\}\in \mathbf{E}} \mathbf{w_{ij} x_{ij}}$$

How do we represent cuts?

Partition graph
to maximize
edge weight across cut

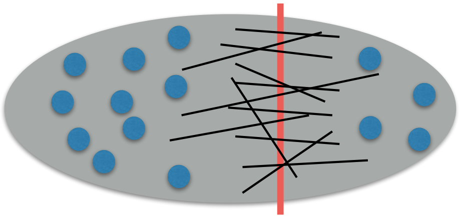
Cuts



Given $(x_e)_{e \in E}$, existence of partition {S,V-S}?

Idea: One variable x_{ij} per vertex pair
instead of per edge

Cut properties ?

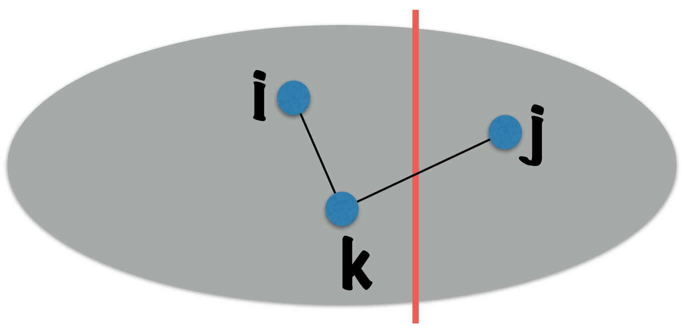


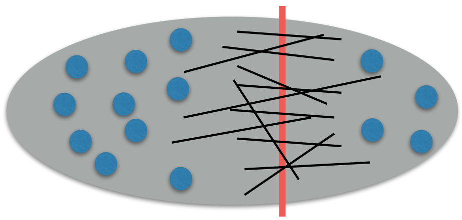
$$\mathbf{implicit\ symmetry}: \ \mathbf{x_{ij} = x_{ji}}$$

Cut properties

$$\mathbf{x_{ij}} = \begin{cases} 1 & \textbf{if crosses cut} \\ 0 & \textbf{otherwise} \end{cases}$$

$$\mathbf{x_{ij} \leq x_{ik} + x_{kj}}$$

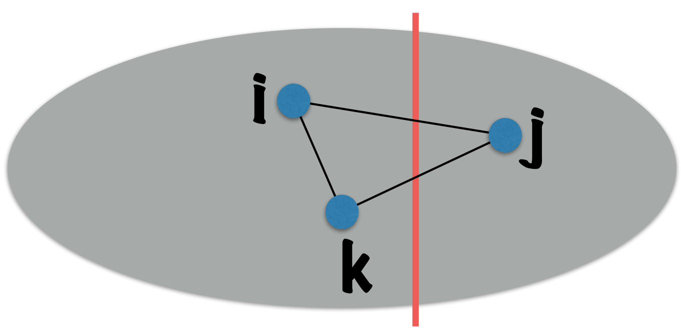


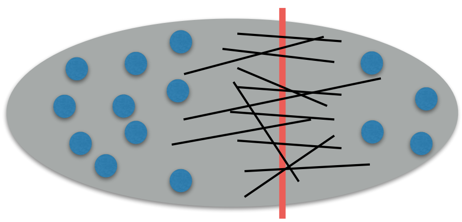


Cut properties

$$\mathbf{x_{ij}} = \begin{cases} 1 & \textbf{if crosses cut} \\ 0 & \textbf{otherwise} \end{cases}$$

$$\mathbf{x_{ij}} + \mathbf{x_{jk}} + \mathbf{x_{ki}} \leq \mathbf{2}$$





Lemma

$$\mathbf{x_{ij} \leq x_{ik} + x_{kj}}$$
$$\mathbf{x_{ij} + x_{jk} + x_{ki} \leq 2}$$
$$\mathbf{x_{ij} \in \{0,1\}}$$

$\Longleftrightarrow$

There exists a cut S
s.t.
$\mathbf{x_{ij} = 1}$
iff pair $\mathbf{i, j}$
crosses cut (S,V-S)

$\Longleftarrow$ is clear

$\Longrightarrow$ needs a proof

$$\mathbf{x_{ij} \le x_{ik} + x_{kj}}$$
$$\mathbf{x_{ij} + x_{jk} + x_{ki} \le 2}$$
$$\mathbf{x_{ij} \in \{0, 1\}}$$

Proof: **let** $\mathbf{S \leftarrow \{1\} \cup \{i : x_{1i} = 0\}}$

$\mathbf{x_{ij}, i \in S, j \notin S :}$ $\quad \mathbf{\underbrace{x_{1j}}_{=1} \le \underbrace{x_{1i}}_{=0} + x_{ij}}$

$\mathbf{x_{ij}, i \in S, j \in S :}$ $\quad \mathbf{x_{ij} \le \underbrace{x_{i1}}_{=0} + \underbrace{x_{1j}}_{=0}}$

$\mathbf{x_{ij}, i \notin S, j \notin S :}$ $\quad \mathbf{x_{ij} + \underbrace{x_{1i}}_{=1} + \underbrace{x_{1j}}_{=1} \le 2}$

QED

LP relaxation for Maxcut

Symmetric variables $\mathbf{x_{ij}}$ for $\mathbf{i}, \mathbf{j} \in \mathbf{V}$

$$\max \sum_{\{\mathbf{i},\mathbf{j}\} \in \mathbf{E}} \mathbf{w_{ij}} \; \mathbf{x_{ij}} \; :$$

$$\mathbf{x_{ij}} \le \mathbf{x_{ik}} + \mathbf{x_{kj}}$$

$$\mathbf{x_{ij}} + \mathbf{x_{jk}} + \mathbf{x_{ki}} \le \mathbf{2}$$

$$\mathbf{0} \le \mathbf{x_{ij}} \le \mathbf{1}$$

Maxcut

Maxcut

LP relaxation for Max-Cut

Symmetric variables $\mathbf{x_{ij}}$ for $\mathbf{i, j \in V}$

$$\max \sum_{\{i,j\} \in E} \mathbf{w_{ij} x_{ij}}$$

$$\mathbf{x_{ij} \le x_{ik} + x_{kj}}, \text{ for all } \mathbf{i, j, k \in V},$$

$$\mathbf{x_{ij} + x_{jk} + x_{ki} \le 2}, \text{ for all } \mathbf{i, j, k \in V}.$$

Q: Can we do better than 2?
A: Not using this LP ...

Theorem: There exists a graph G such that
LP(G)/maxcut(G) > 2- (a little bit).

Idea: Find a graph G such that:

Property 1
LP(G) at least
$\#\{\text{edges}\} \cdot 0.99$

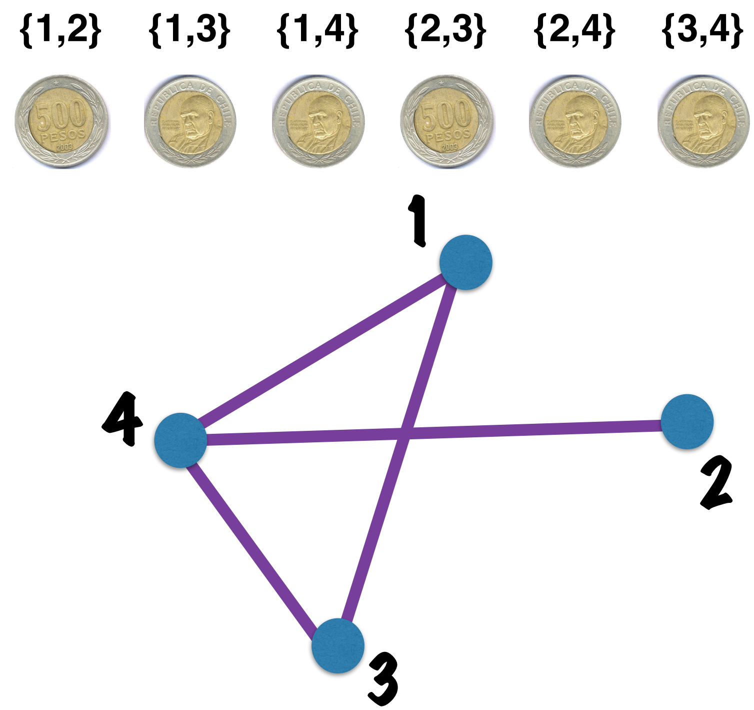
and

Property 2
maxcut(G) at most
$\#\{\text{edges}\} \cdot 0.51$

then $LP(G)/maxcut(G) \geq 0.99/0.51 = 1.9\ldots$

How do we find G? Random graphs

G(n,p): graphs with n nodes,
each edge drawn
independently
with probability p.



For each u,v in V:

head w.p. p $\longrightarrow$ add {u,v} to G,

tail w.p. 1-p $\longrightarrow$ we do not.

Lemma: if all cycles in G have length at least $g = 100 \log(100)$, then

$LP(G) > 0.99 \cdot \#\{\text{edges of } G\}$

Property 1
LP(G) at least
#{edges}·0.99

$G \sim G(n, C/n)$

Markov's Inequality

$\Pr(\#\{\text{cycles of length at most } g-1\} > n^{1/2}) < 0.5$

G': graph after breaking these cycles in G by removing at most $n^{1/2}$ edges.

Hence, $LP(G') > 0.99 \cdot \#\{\text{edges of } G'\}$

G' satisfies Property 1 !

Property 2

maxcut(G') at most

#{edges}· 0.51

$G \sim G(n, C/n)$

maxcut(G')<0.51· #{edges of G'}

with probability at least 0.8

Putting things together ...

Pr(G satisfies Property 1 and Property 2)
> 1-Pr(G does not satisfy Property 1)
-Pr(G does not satisfy Property 2)
> 1 - 0.5 - 0.2
= 0.3

Hence, there exists a graph G' satisfying both properties.

LP(G')/maxcut(G')>0.99/0.51=1.9...

Maxcut

Maxcut

Property 1

LP(G) at least #{edges}·0.99

Random graphs

$G \sim G(n, C/n)$

Pr(#{cycles of length at most g-1}<$n^{1/2}$) < 0.5

Markov's Inequality

X: non-negative random variable
a: positive real number

$$\mathbf{Pr}(\mathbf{X} \geq \mathbf{a}) \leq \frac{\mathbf{E}(\mathbf{X})}{\mathbf{a}}$$

Proof

Step function $$\mathbf{H_a(x)} = \begin{cases} \mathbf{a} & \mathbf{if\ x \geq a,} \\ \mathbf{0} & \mathbf{if\ x < a.} \end{cases}$$

$$\mathbf{X \geq H_a(X),}$$

$$\mathbf{E(X) \geq E(H_a(X)) = a \cdot Pr(X \geq a).}$$

Goal: Pr(#{cycles of length at most g-1}>$n^{1/2}$) < 0.5

Use Markov's inequality!

G~ G(n,C/n) such that n=C^{3g}

X = #{cycles of length at most g-1}, a=$C^{1.5g}$

$$\Pr(X \geq C^{1.5g}) \leq \frac{E(X)}{C^{1.5g}}$$

$$\mathbf{E(X)} \leq \sum_{\ell=3}^{g-1} \frac{n^\ell}{\ell}\left(\frac{C}{n}\right)^\ell = \sum_{\ell=3}^{g-1} \frac{C^\ell}{\ell} \leq C^g,$$

Hence, $\Pr(X \geq C^{1.5g}) \leq \frac{E(X)}{C^{1.5g}} \leq \frac{C^g}{C^{1.5g}} = \frac{1}{C^{g/2}} \leq \frac{1}{n^{1/6}},$

Choosing n>(0.5)^6=64, $\Pr(X < n^{1/2}) > 0.5.$

Maxcut

Maxcut

Quadratic relaxation for Maxcut

$$\max \sum_{\{\mathbf{i},\mathbf{j}\}\in \mathbf{E}} \mathbf{w_{ij}} \left(\mathbf{v_i} \cdot \mathbf{v_j}\right) :$$

$$|\mathbf{v_i}|^{\mathbf{2}} = \mathbf{1}$$

Maxcut

Maxcut

Quadratic relaxation

$$\max \sum_{\{\mathbf{i},\mathbf{j}\}\in \mathbf{E}} \mathbf{w_{ij}} \frac{-\mathbf{v_i}\cdot\mathbf{v_j}+\mathbf{1}}{\mathbf{2}} \ :$$

$$\mathbf{v_i} \cdot \mathbf{v_i} = \mathbf{1}$$

Linear program

Variables: real numbers
Objective: linear
Constraints: linear equalities

$$\mathbf{x_i}$$

$$\sum \mathbf{c_i x_i}$$

$$\textstyle\sum_{\mathbf{j}} \mathbf{a_{kj} x_j = b_k}$$

Vector program

Variables: vectors
Objective: linear in dot products
Constraints: linear equalities in dot products of the vectors

$$\mathbf{v_i}$$

$$\sum \mathbf{c_{ij} v_i \cdot v_j}$$

$$\textstyle\sum_{\mathbf{ij}}^{\mathbf{(k)}} \mathbf{a_{ij}^{(k)} v_i \cdot v_j = b_k}$$

What's the big deal about vector programs?

Variables: vectors $\quad \mathbf{v_i}$

Objective: linear in dot products $\quad \sum \mathbf{c_{ij} v_i \cdot v_j}$

Constraints: linear equalities in dot products of the vectors $\quad \sum_{\mathbf{ij}}^{(\mathbf{k})} \mathbf{a_{ij}^{(k)} v_i \cdot v_j = b_k}$

Let $\mathbf{y_{ij} = v_i \cdot v_j}$

Objective: min or max $\sum \mathbf{c_{ij} y_{ij}}$

s.t. $\quad \forall \mathbf{k} \; : \sum_{\mathbf{ij}}^{(\mathbf{k})} \mathbf{a_{ij}^{(k)} y_{ij} = b_k}$

and there exist vectors $\mathbf{v_i}$

s.t. $\mathbf{y_{ij} = v_i \cdot v_j}$

Positive semi-definite matrices

Consider a matrix $\mathrm{Y} = (\mathrm{y_{ij}})_{1 \leq \mathrm{i,j} \leq \mathrm{n}}$

Assume it is symmetric: $\mathrm{y_{ij}} = \mathrm{y_{ji}}$

then the following are equivalent:

- there exist vectors $\mathrm{v_i}$ s.t. $\mathrm{y_{ij}} = \mathrm{v_i} \cdot \mathrm{v_j}$
- Y is positive semi-definite
- for all vectors a, $\sum_{\mathrm{ij}} \mathrm{a_i y_{ij} a_j} \geq 0$

Semi-definite programming

Consider a matrix $\mathbf{Y} = (\mathbf{y_{ij}})_{\mathbf{1 \le i,j \le n}}$

Objective:

min/max $\sum \mathbf{c_{ij} y_{ij}}$

s.t. $\forall \mathbf{k} \;:\; \sum_{\mathbf{ij}}^{(\mathbf{k})} \mathbf{a_{ij}^{(k)} y_{ij} = b_k}$

$\mathbf{y_{ij} = y_{ji}}$

$\mathbf{Y}$ **positive semi-definite** $\qquad \forall \mathbf{a} : \; \sum_{\mathbf{ij}} \mathbf{a_i y_{ij} a_j \ge 0}$

Convex

Theorem

Objective:

min/max $\sum c_{ij} y_{ij}$

s.t. $\forall k \ : \sum_{ij}^{(k)} a_{ij}^{(k)} y_{ij} = b_k$

$y_{ij} = y_{ji}$

Y **positive semi-definite**

Can be "solved" in polynomial time by ellipsoid algorithm

Quadratic relaxation for Maxcut

$$\max \sum_{\{\mathbf{i},\mathbf{j}\}\in \mathbf{E}} \mathbf{w_{ij}} \frac{-\mathbf{v_i}\cdot\mathbf{v_j}+\mathbf{1}}{\mathbf{2}} \; :$$
$$\mathbf{v_i}\cdot\mathbf{v_i} = \mathbf{1}$$

Can be "solved" in polynomial time by ellipsoid algorithm

Maxcut

Maxcut

Solve the quadratic relaxation for Maxcut

$$\max \sum_{\{\mathbf{i},\mathbf{j}\}\in \mathbf{E}} \mathbf{w_{ij}} \frac{-\mathbf{v_i}\cdot\mathbf{v_j}+\mathbf{1}}{\mathbf{2}} :$$

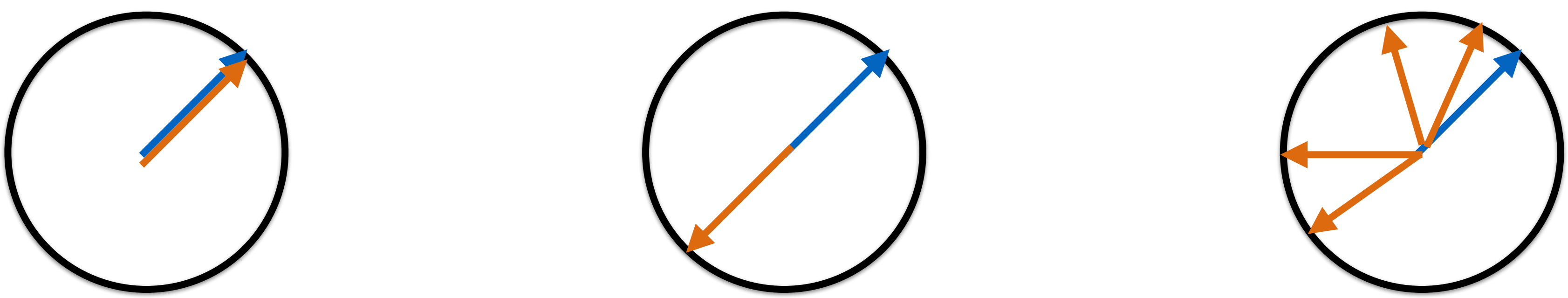
$$\mathbf{v_i}\cdot\mathbf{v_i} = \mathbf{1}$$

How do we round?

$$\mathbf{v_i} \mapsto \mathbf{x_i} \in \{-1, 1\}$$

Value(Output): $\sum_{\{i,j\} \in E} \mathbf{w_{ij}} \frac{-\mathbf{x_i}\mathbf{x_j}+1}{2}$

OPT: $\leq \sum_{\{i,j\} \in E} \mathbf{w_{ij}} \frac{-\mathbf{v_i}\cdot\mathbf{v_j}+1}{2}$



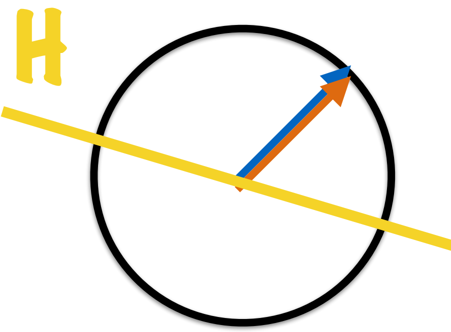
Want: $\mathbf{x_i} = \mathbf{x_j}$ **Want:** $\mathbf{x_i} = -\mathbf{x_j}$ **Want:** ?

Randomized rounding

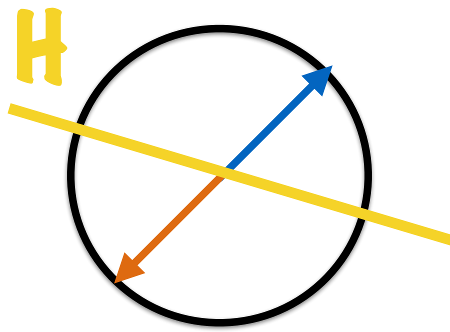
Random line (hyperplane) H

above H: $v_i \mapsto 1$

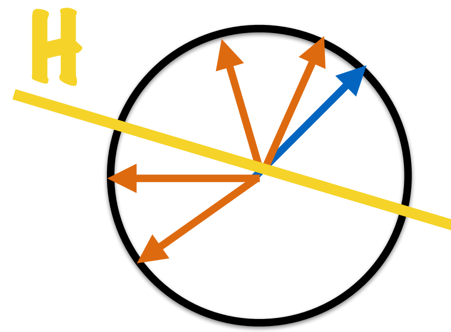
below H: $v_i \mapsto -1$



Want: $x_i = x_j$



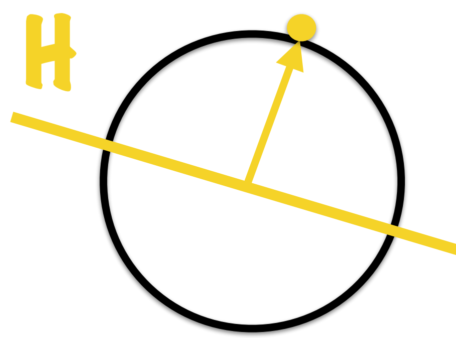
Want: $x_i = -x_j$



Want: ?

How do we pick a random hyperplane?

Pick point uniformly at random on unit sphere
It defines a vector
H: normal hyperplane through the origin.

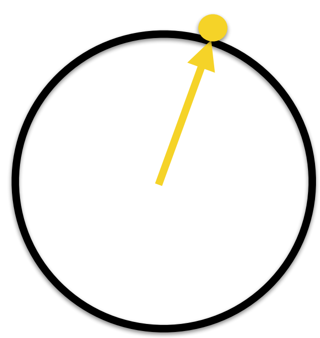


How to pick point on unit sphere?

Coordinate i: $\mathrm{x_i} \leftarrow \mathcal{N}(0, 1)$

Unit vector $\mathrm{r} \leftarrow \mathbf{x}/||\mathbf{x}||$

Or: spherical coordinates
radius = 1
angles = independent uniform



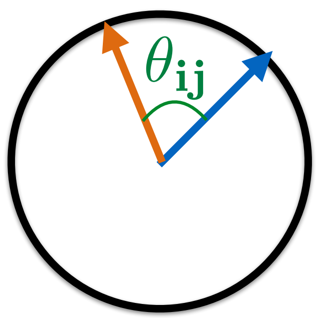
Maxcut

Maxcut

Analysis

OPT: $\leq \sum_{\{\mathbf{i},\mathbf{j}\}\in \mathbf{E}} \mathbf{w_{ij}} \frac{-\mathbf{v_i}\cdot\mathbf{v_j}+\mathbf{1}}{\mathbf{2}}$

$= \sum_{\{\mathbf{i},\mathbf{j}\}\in \mathbf{E}} \mathbf{w_{ij}} \frac{-\cos\theta_{\mathbf{ij}}+\mathbf{1}}{\mathbf{2}}$



Analysis

E(Value(Output)): $\sum_{\{\mathbf{i},\mathbf{j}\}\in \mathbf{E}} \mathbf{w_{ij}}\ \mathbf{E}\left(\frac{-\mathbf{x_i}\mathbf{x_j}+\mathbf{1}}{\mathbf{2}}\right)$

$= \sum_{\{\mathbf{i},\mathbf{j}\}\in \mathbf{E}} \mathbf{w_{ij}}\ \Pr(\mathbf{x_i} \neq \mathbf{x_j})$

Random line (hyperplane) H

above H: $\mathbf{v_i} \mapsto 1$

below H: $\mathbf{v_i} \mapsto -1$

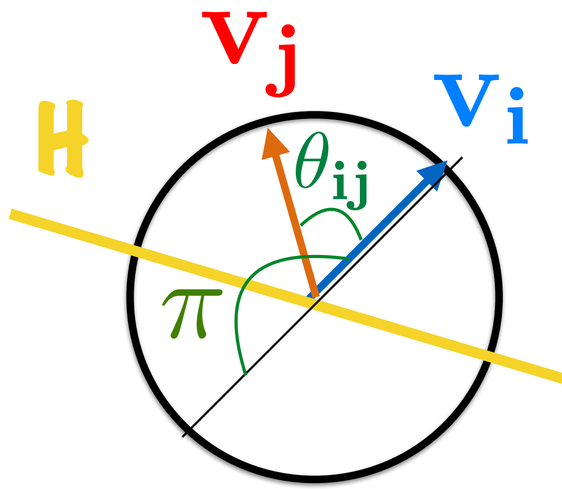
$$\Pr(\mathbf{x_i} \neq \mathbf{x_j}) = \Pr(\mathbf{H} \textbf{ separates } \mathbf{v_i} \textbf{ from } \mathbf{v_j})$$

Analysis

Random line (hyperplane) H

above H: $\mathbf{v_i} \mapsto 1$

below H: $\mathbf{v_i} \mapsto -1$



$$\Pr(\mathbf{H\ separates\ v_i\ from\ v_j}) = \theta_{\mathbf{ij}}/\pi$$

Analysis

E(Value(Output)): $\sum_{\{\mathbf{i},\mathbf{j}\}\in\mathbf{E}} \mathbf{w_{ij}} \frac{\theta_{\mathbf{ij}}}{\pi}$

OPT: $\leq \sum_{\{\mathbf{i},\mathbf{j}\}\in\mathbf{E}} \mathbf{w_{ij}} \frac{-\cos\theta_{\mathbf{ij}}+\mathbf{1}}{\mathbf{2}}$

Lemma: $\forall\theta$: $\frac{\theta}{\pi} \geq .878\ldots \frac{-\cos\theta+\mathbf{1}}{\mathbf{2}}$

E(Value(Output)) $\geq .878\ldots$ **OPT**

Better than .5

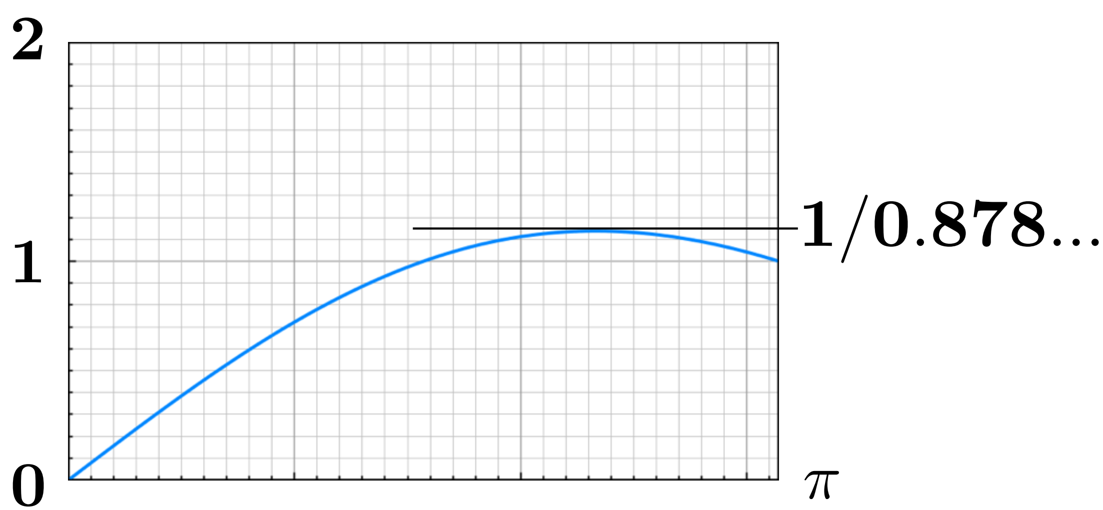
QED

Proof of Lemma

$\forall \theta : \quad \frac{\theta}{\pi} \geq .878\ldots \frac{-\cos\theta + 1}{2}$

$$\mathbf{f} : \theta \mapsto \frac{\pi(1 - \cos\theta)}{2\theta}$$

What is the maximum of f?



QED

Recap

1. Solve SDP relaxation:

$$\max \sum_{\{\mathbf{i},\mathbf{j}\}\in \mathbf{E}} \mathbf{w_{ij}} \frac{-\mathbf{v_i}\cdot\mathbf{v_j}+\mathbf{1}}{\mathbf{2}} \quad : \quad \mathbf{v_i}\cdot\mathbf{v_i} = 1$$

2. Rounding:

Random line (hyperplane) H

above H: $\mathbf{v_i} \mapsto 1$

below H: $\mathbf{v_i} \mapsto -1$

3. Output resulting cut

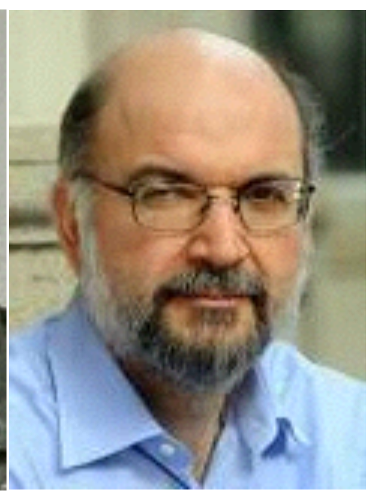
Theorem: E(Value(Output)) $\geq .878\ldots$ OPT

Maxcut

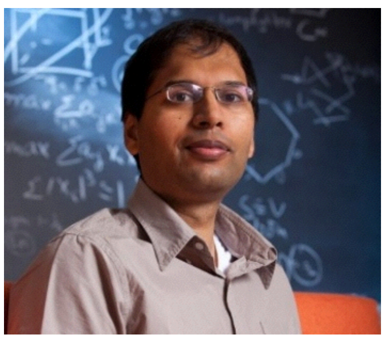
Maxcut

Negative results

KARPfest80

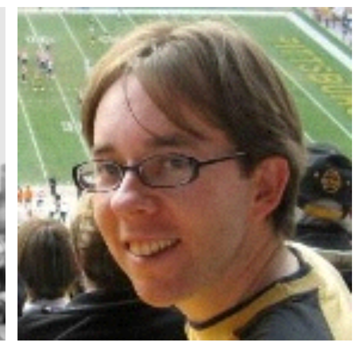


If the Unique Games Conjecture holds then approximating MaxCut better than 0.878... is NP-hard

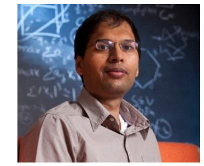

Subash Khot


Guy Kindler


Elchanan Mossel


Ryan O'Donnell

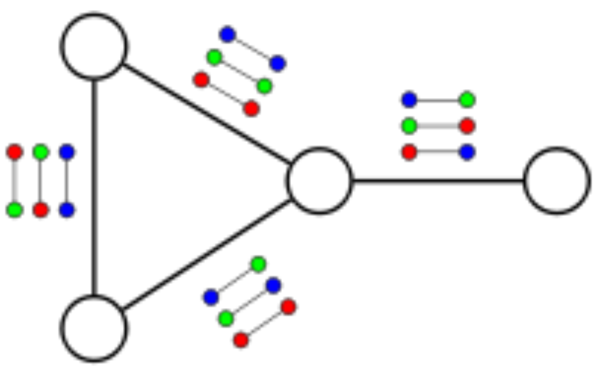
UGC: the following problem is (?) NP-hard

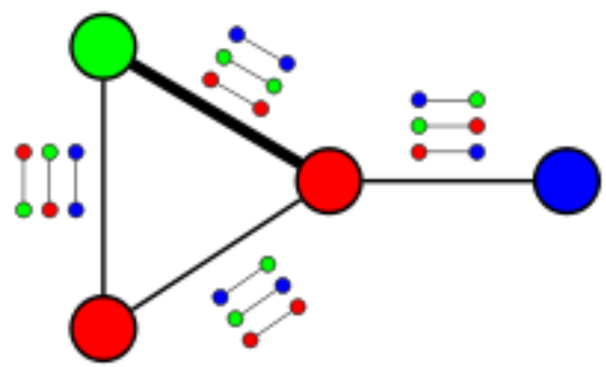


Input:

- graph with for each edge {u,v}, permutation rule "if u is red then must be green; if u is green then v must be blue; if u is blue than v must be yellow..." such that:
  - either there is a coloring satisfying 90% if the edge rules,
  - or no coloring satisfies more than 10% of the edges rules

Output: decide which of the two holds!

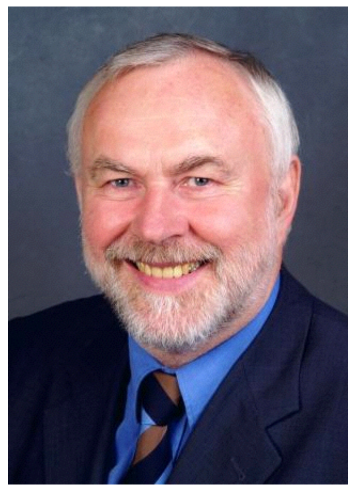




Positive results

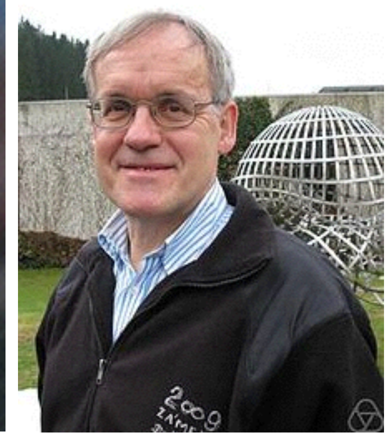
Ellipsoid method ... for SDPs

Leonid
Khachiyan



Martin
Grötschel

László
Lovász



Alexander
Schrijver

SDP relaxation of MaxCut

Charles Delorme

Svatopluk Poljak

Randomized rounding and .878... approx

Michel
Goemans

David
Williamson

Maxcut