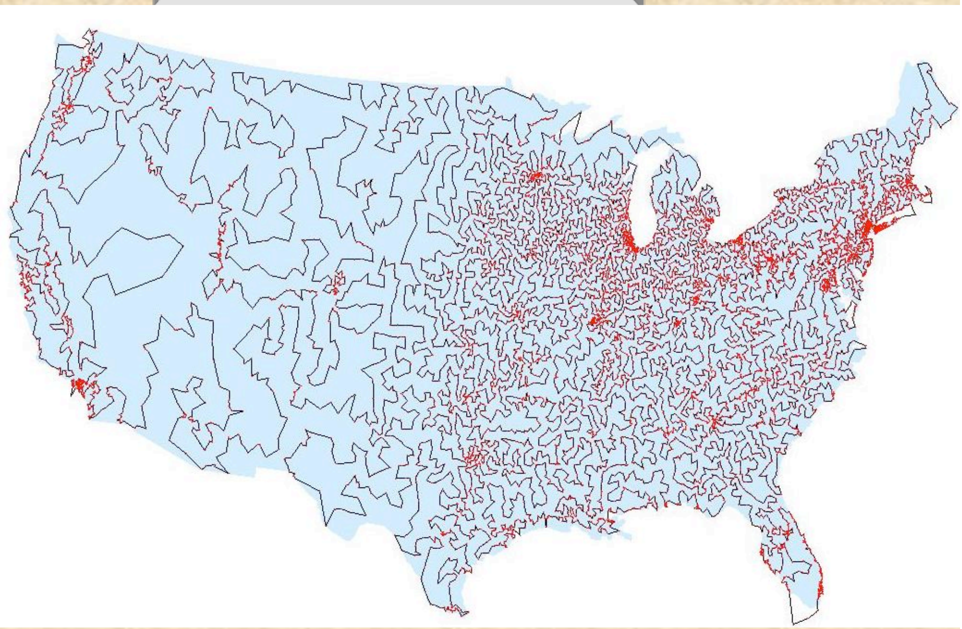
LP/SDP Algorithms

Claire Mathieu
Brown University

Many optimization problems can be written as integers programs

- Traveling salesman tour, vehicle routing
- Steiner tree, connecting
- Clustering, cuts
- Coloring
- Short paths
- Max satisfiability
- …



Integer programming: NP-hard

Linear programming: in P

Maximize (6x+5y)

such that:

2x-5y≤6

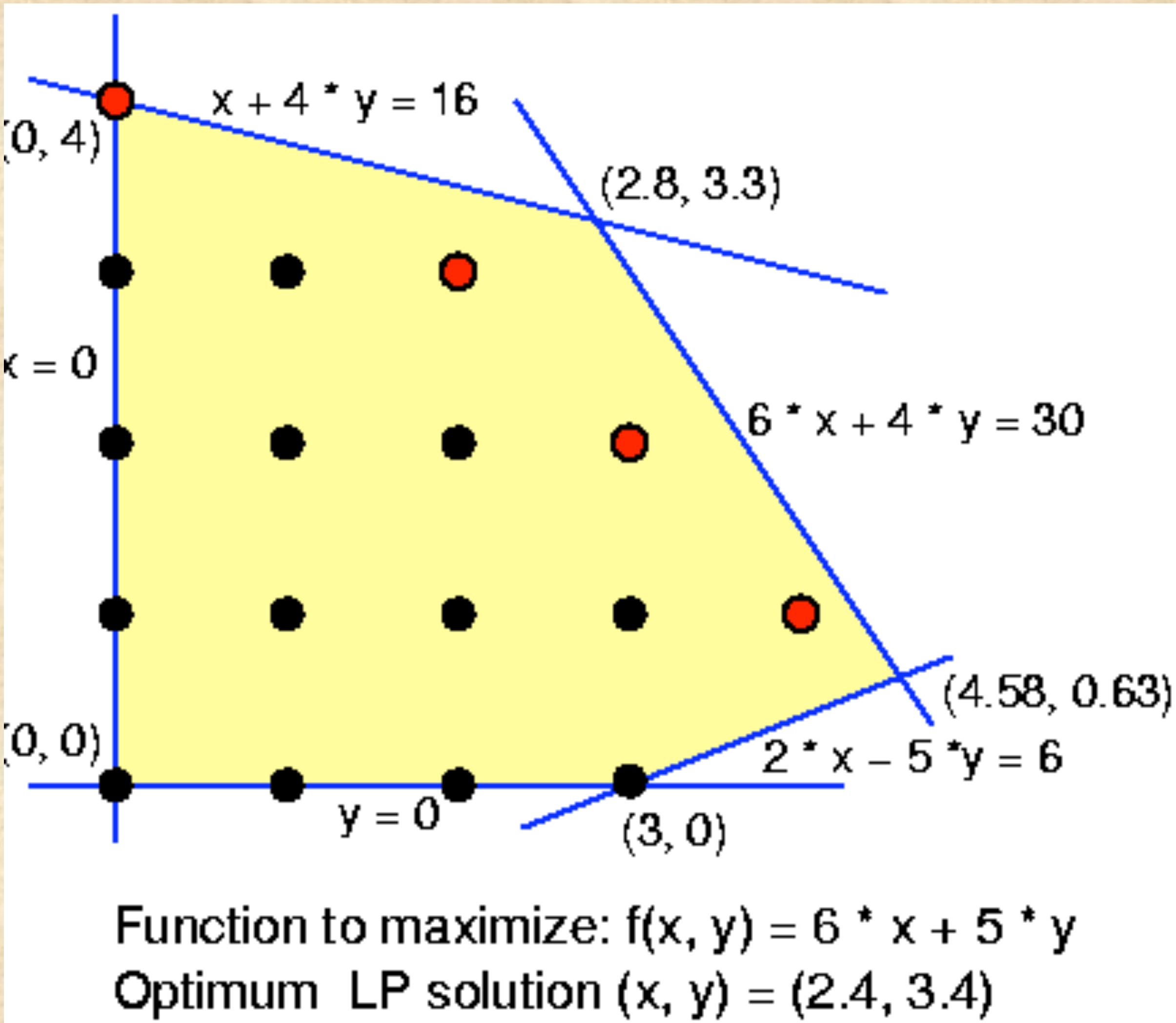
6x+4y≤30

x+4y≤16

x,y≥0

x,y integers (IP)

x,y real (LP)



LP Algo

- Simplex

Steepest descent

Worst case exp

Fast if smooth

- Ellipsoid method

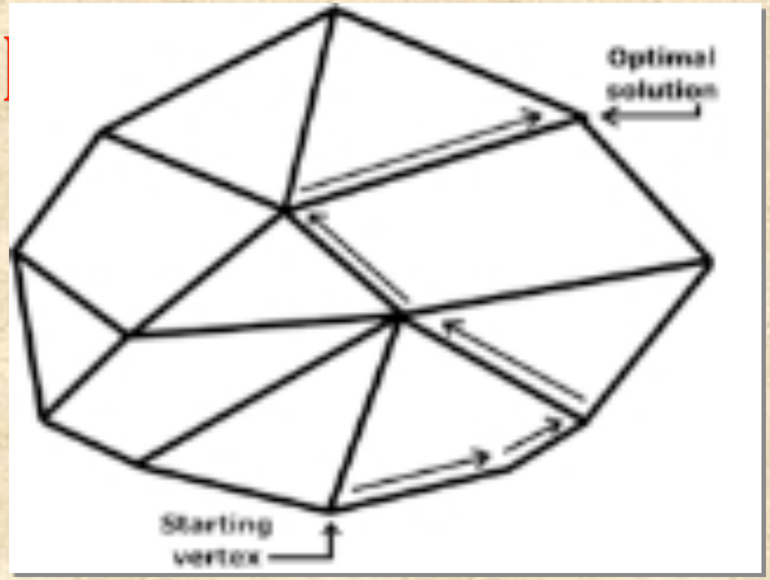
Polynomial time

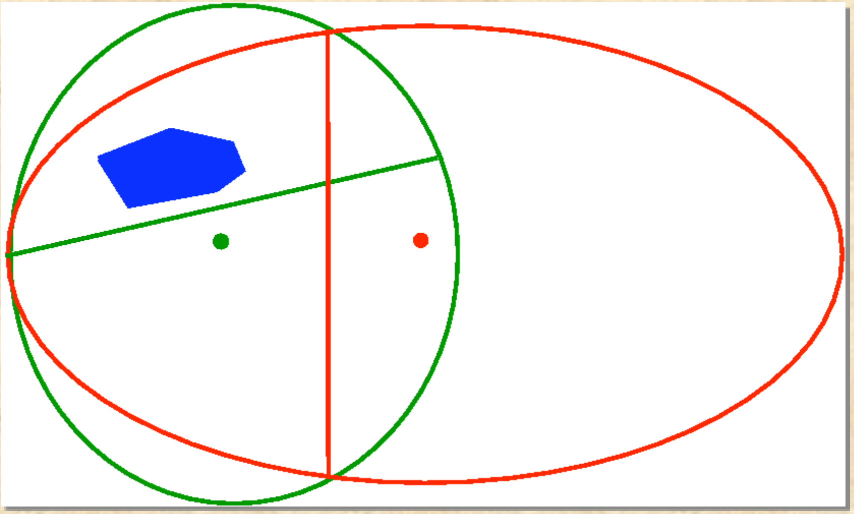
Oracle-Based:

Q: "x feasible?"

A: "yes" or "No since $3x_1+2y_2+y_3<10$"

- Approx in n polylog packing/covering





LP Plan

- Linear programming
- Using LP primal for algorithm: multiway cut
- LP duality
- Using LP dual for analysis: vehicle routing
- Using LP dual for analysis: Correlation clustering
- Using LP Primal-dual for online algorithm: ski rental

What's LP good for?

Approximation algorithm

1. Solve LP relaxation instead of IP

With luck, it's integral

(ex: bipartite matching, totally unimodular matrices)

If not,

2. "Round" solution to a feasible integer solution

Analysis

3. Relaxation implies that LP profit ≥ OPT integer profit

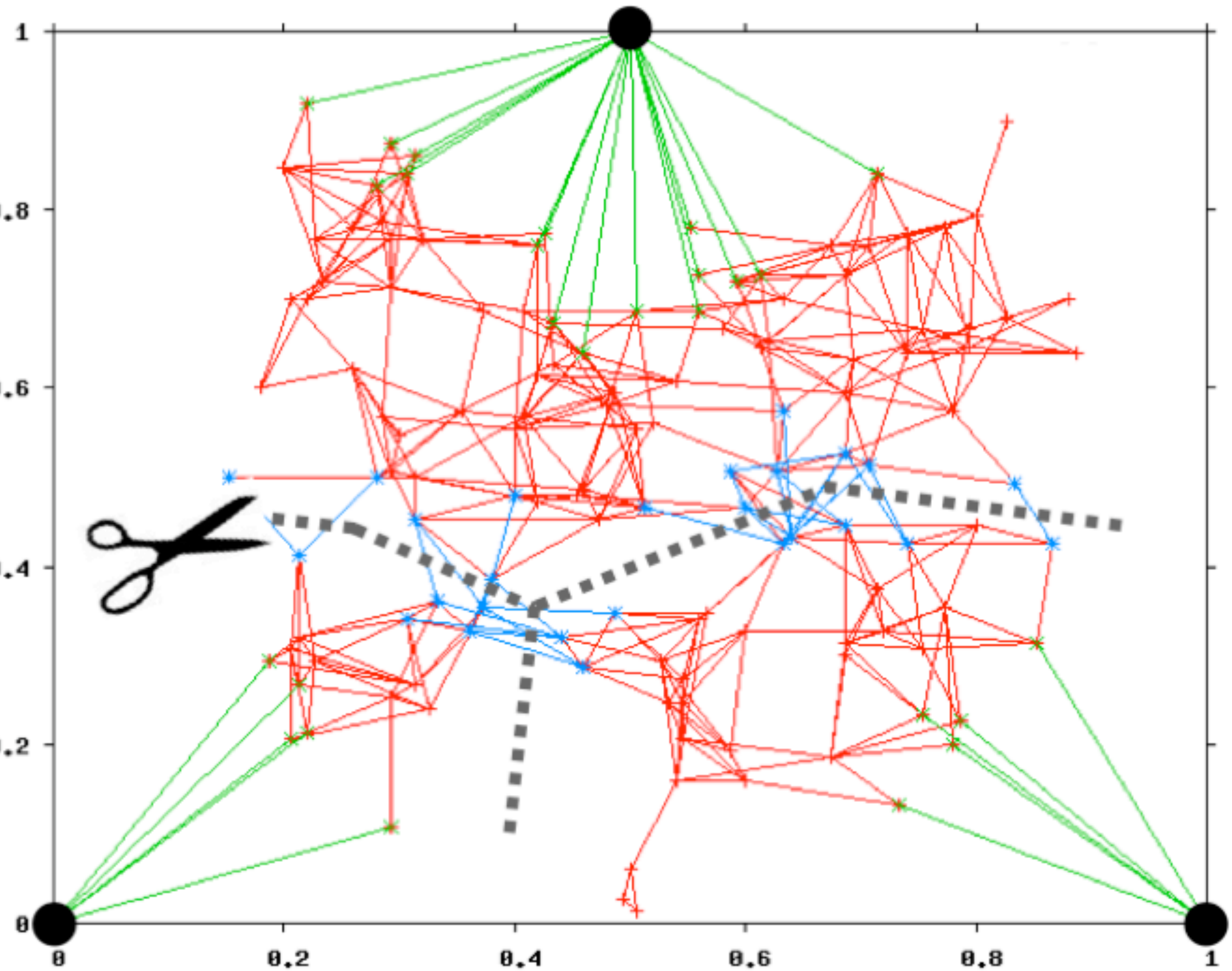
4. Show that profit is not much less than LP profit

Three-Way Cut

- Input: Graph, and three "terminal" vertices
- Output: minimum set of edges disconnecting terminals from one another

Remark: if 3 replaced by 2, then?

3-way cut



IP for three way cut

Three colors x,y,z

For each 3-coloring of the vertices,

count the number of bichromatic edges

and minimize that

Minimize $\Sigma_{\text{edges } e}\ d_e$

subject to:

For vertex u: $x_u+y_u+z_u=1$ (3-coloring)

$x_{t1}=1$, $y_{t2}=1$, $z_{t3}=1$ (one color per terminal)

For edge e=uv: $d_{uv}\geq(1/2)(|x_u-x_v|+|y_u-y_v|+|z_u-z_v|)$

$x_u,y_u,z_u,d_{uv}\geq 0$

x_u,y_u,z_u,d_{uv} integers

LP relaxation

Minimize $\Sigma_{\text{edges } e}\ d_e$

subject to:

For vertex u: $x_u+y_u+z_u=1$ (3-coloring)

$x_{t1}=1$, $y_{t2}=1$, $z_{t3}=1$ (one color per terminal)

For edge e=uv: $d_{uv}\geq(1/2)(|x_u-x_v|+|y_u-y_v|+|z_u-z_v|)$

$x_u,y_u,z_u,d_{uv}\geq 0$

Associate to u point (x_u,y_u,z_u) in triangle $\{x+y+z=1,x,y,z\geq 0\}$

Terminals at corners

➡ Embedding of G

LP goal: min l_1 length of edges in embedding

The rounding problem

1. Solve LP relaxation gives optimal l_1 embedding
2. "Round" solution to 3-way cut:

   how?

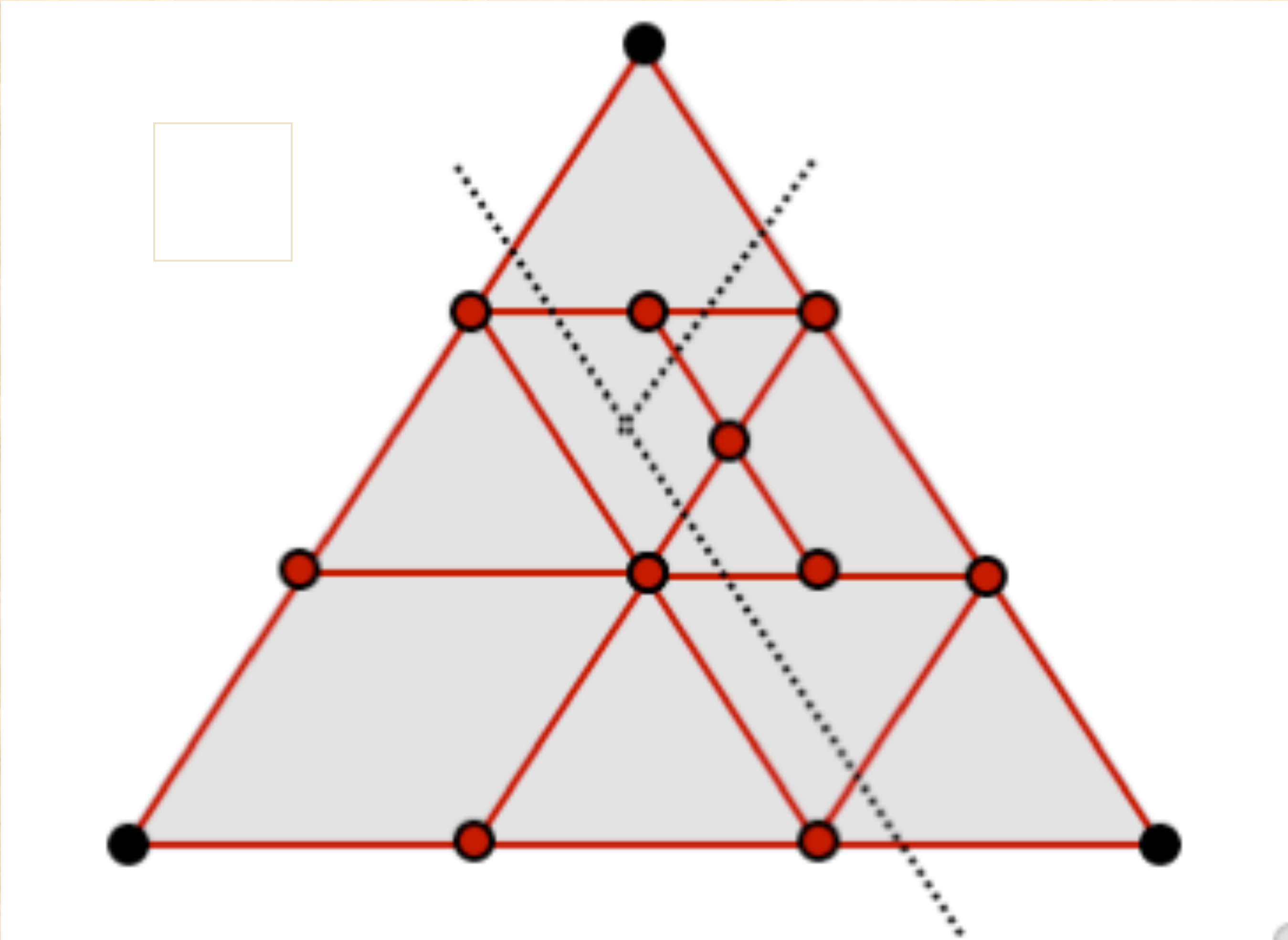
… so that it can be analyzed…

3. Relaxation implies that

   l_1 length of embedding ≤ OPT

4. Let's round so that

   cost of 3-way cut ≤ (small)*l_1 length of embedding



How to round

- Greedy: round $\max(x_u, y_u, z_u)$ to 1, and the other two coordinates to 0
- Independent: round to

100 w.prob. x_u,

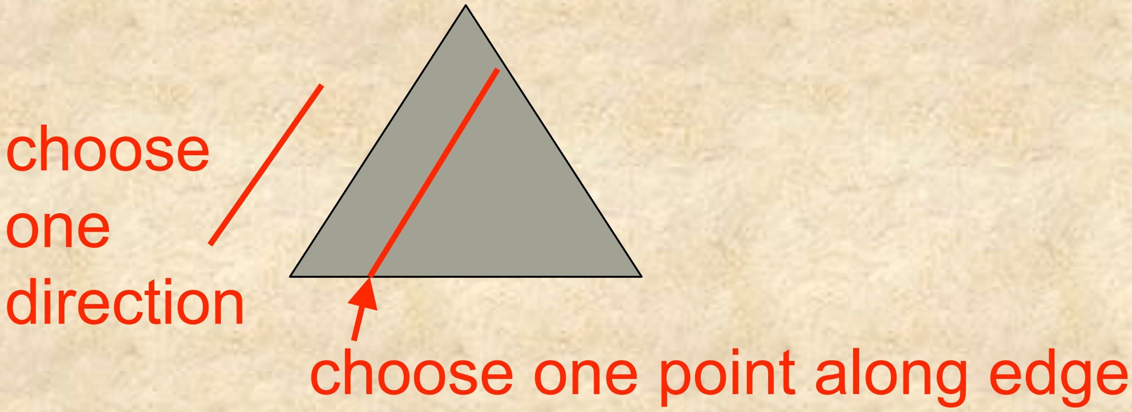
010 w.prob. y_u,

001 w.prob. z_u

- Geometric: better

Geometric rounding

- Pick random line parallel to triangle side: separates one terminal
- Pick random line parallel to triangle side: use it to separate the remaining two terminals



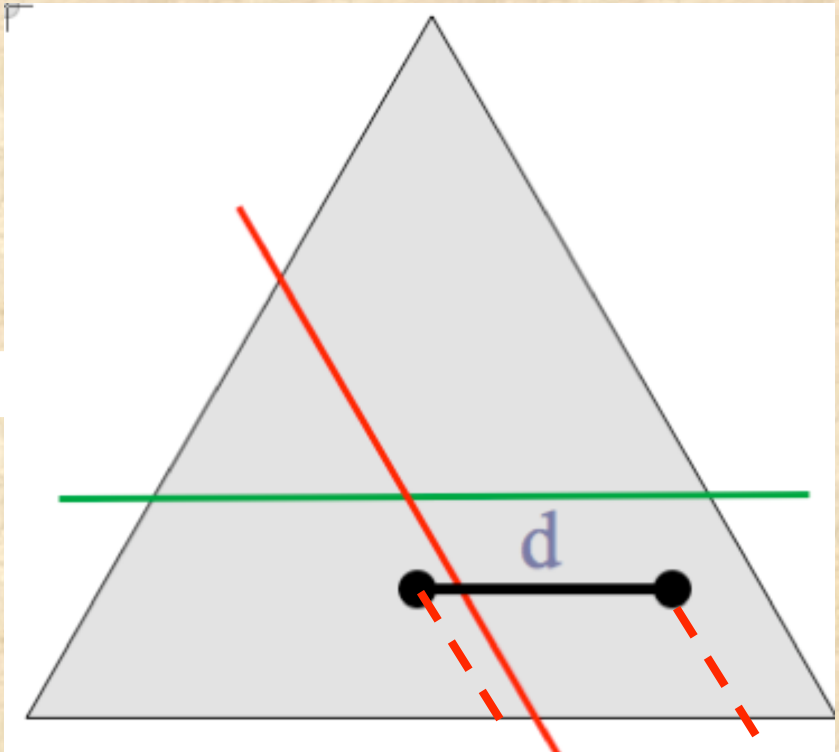
Analysis (1/2)

Prob(e crosses cut)≤

Prob(e crosses red or green line)≤

2 prob(e crosses red)≤

(4/3)d



Analysis (2/2)

E(cost(output)) =

E(number of edges cut) =

$\Sigma_e$prob(e cut) ≤

(4/3)$\Sigma_e d_e$ ≤

(4/3)OPT

[Geometric reasoning gives better cut: (12/11)]

OPEN: finding “right” geometric cut for k-way cut

LP Algorithms

- Vertex cover and set cover
- Scheduling
- Routing
- 3-sat
- …

Algs require:

Good LP relaxation

Good rounding

LP Duality

Minimize $7x+y+5z$

subject to:

$x-y+3z \geq 10$

$5x+2y-z \geq 6$

$x,y,z \geq 0$

Upper bound:

exhibit feasible solution…

Upper bound

(x,y,z)=(2,1,3) feasible

…so: OPT≤30

Lower bound

For example,

can we have OPT ≤ 16?

Minimize 7x+y+5z

subject to:

x-y+3z ≥ 10

5x+2y-z ≥ 6

x,y,z ≥ 0

Upper bound

$(x,y,z)=(2,1,3)$ feasible

…so: $OPT \leq 30$

Minimize $7x+y+5z$

subject to:

$x-y+3z \geq 10$ ← times a

$5x+2y-z \geq 6$ ← times b

$x,y,z \geq 0$

Lower bound

$(x-y+3z)+(5x+2y-z) \geq 10+6$

$6x+y+2z \geq 16$

$7x+y+5x$ has larger coefficients

… so: $OPT \geq 16$

Best lower bound

Maximize $10a+6b$

$7 \geq a+5b$

$1 \geq -a+2b$

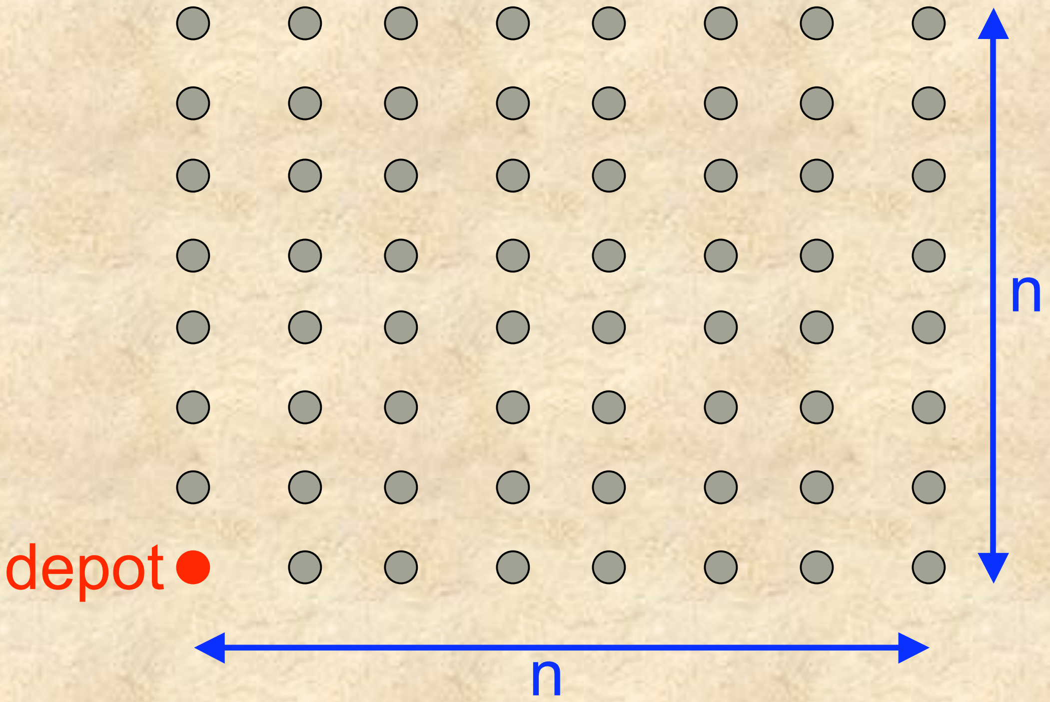
$5 \geq 3a-b$

$a,b \geq 0$

LP duality theorem:

Both LPs have same value

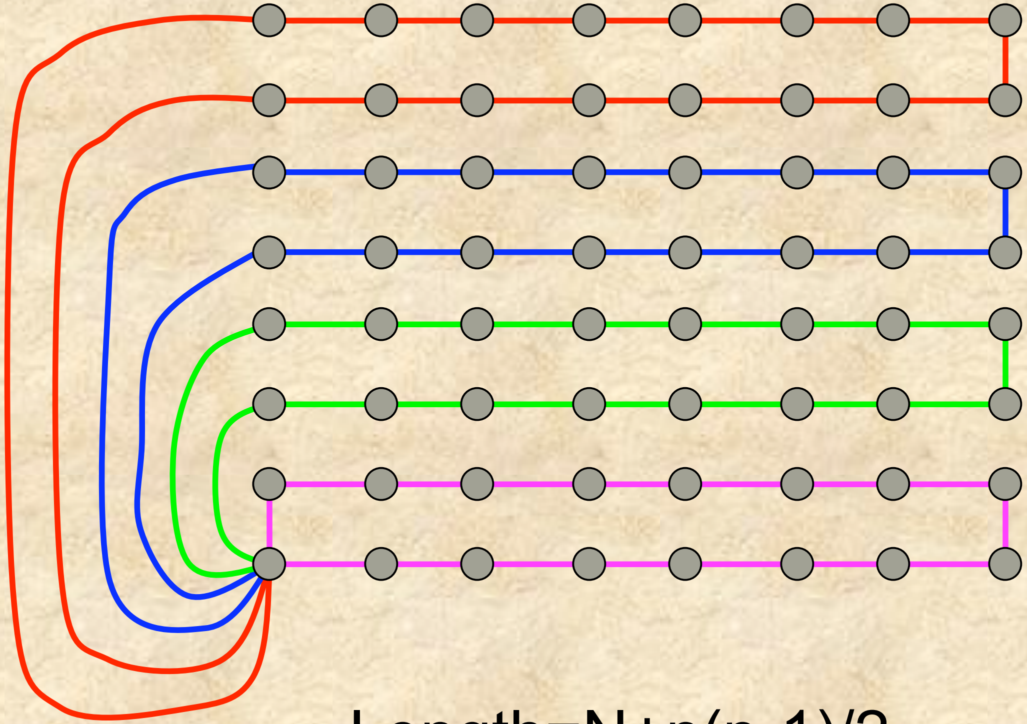
What's LP duality good for?



N=n^2 customers

Vehicle of capacity 2n
Minimize l_1 length of tours

Is this optimal?



Length=N+n(n-1)/2
About (3/2)N

IP for vehicle routing

Variable x_t for each possible tour t visiting ≤ 2n customers
Length w_t of tour t

Min $\Sigma_t\ w_t x_t$
subject to
For customer c: $\Sigma_{t \text{ visiting } c}\ x_t \geq 1$
$x_t \geq 0$
x_t integer

LP primal-dual

Min $\Sigma_t \, w_t x_t$

subject to

For customer c:

$\Sigma_{t \text{ visiting } c} \, x_t \geq 1$

$x_t \geq 0$

Max $\Sigma_c \, y_c$

subject to

For tour t:

$\Sigma_{c \text{ visited by } t} \, y_c \leq w_t$

$y_c \geq 0$

Know solution of value (3/2)N

Exhibit dual feasible solution of value (3/2)N

Feasible dual

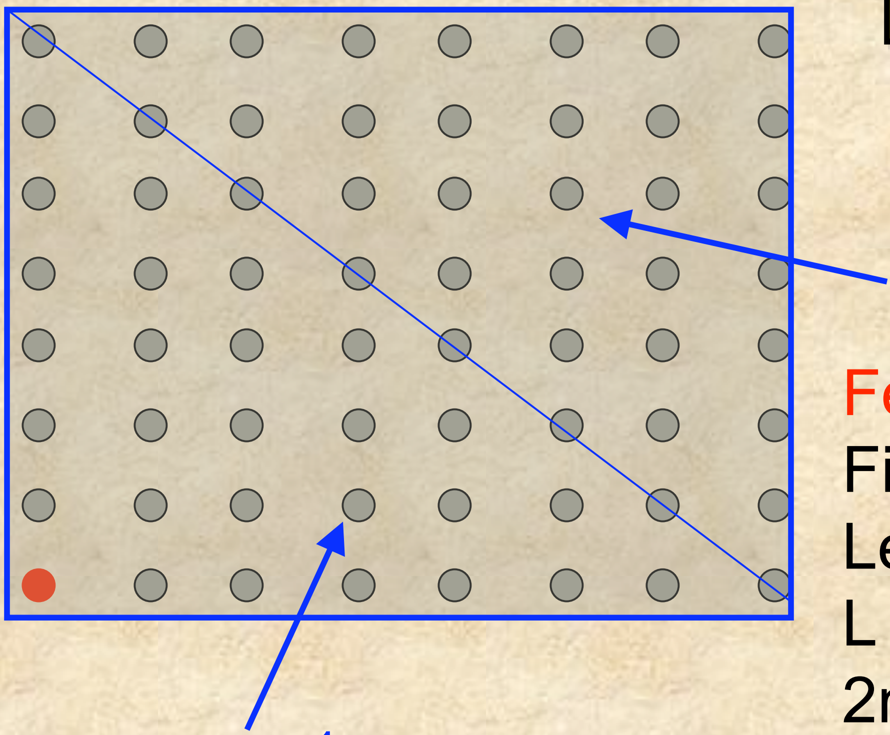
Max $\Sigma_c \, y_c$

subject to

For tour t:

$\Sigma_{c \text{ visited by } t} \, y_c \leq w_t$

$y_c \geq 0$



Feasible?

Fix tour t

Let L=length of t in NorthEast

L customers have y_c=2

2n-L have y_c=1 ✔

OPEN: replace grid by N random uniform points

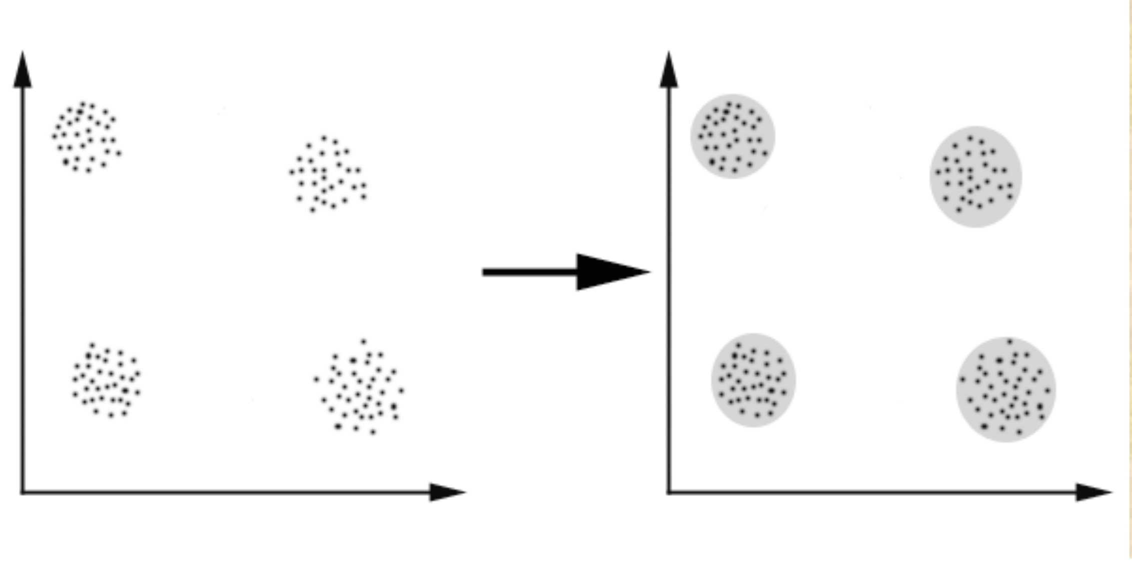
Other uses of LP duality

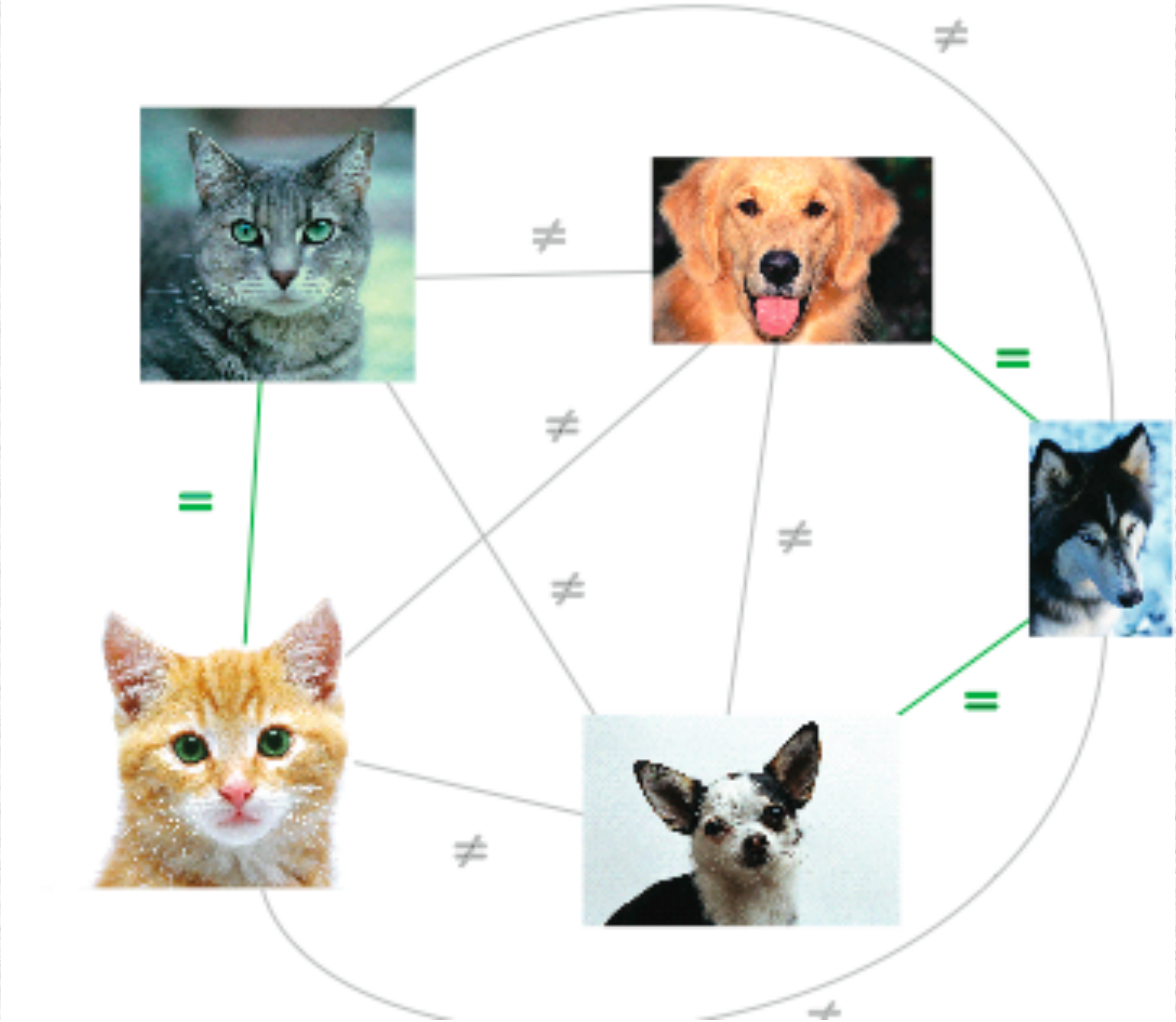
- LP primal used for algorithm
- LP dual used for analysis

Correlation clustering

Clustering

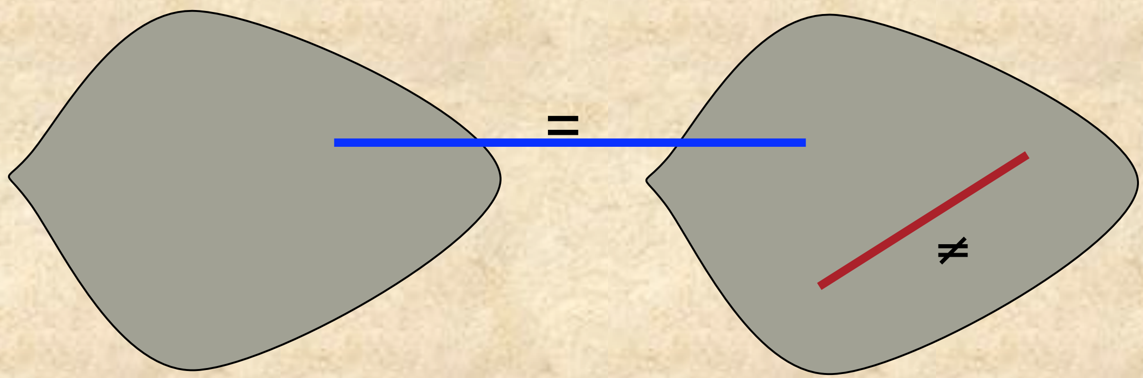
- Organize data in clusters
- Ubiquitous
- Many definitions
- Model is application-dependent





Algorithmic Problem

- Input: complete graph, each edge is labeled “similar” or “dissimilar”
- Output: partition into clusters. Objects inside clusters are similar to one another
- Objective: minimize input/output discrepancies

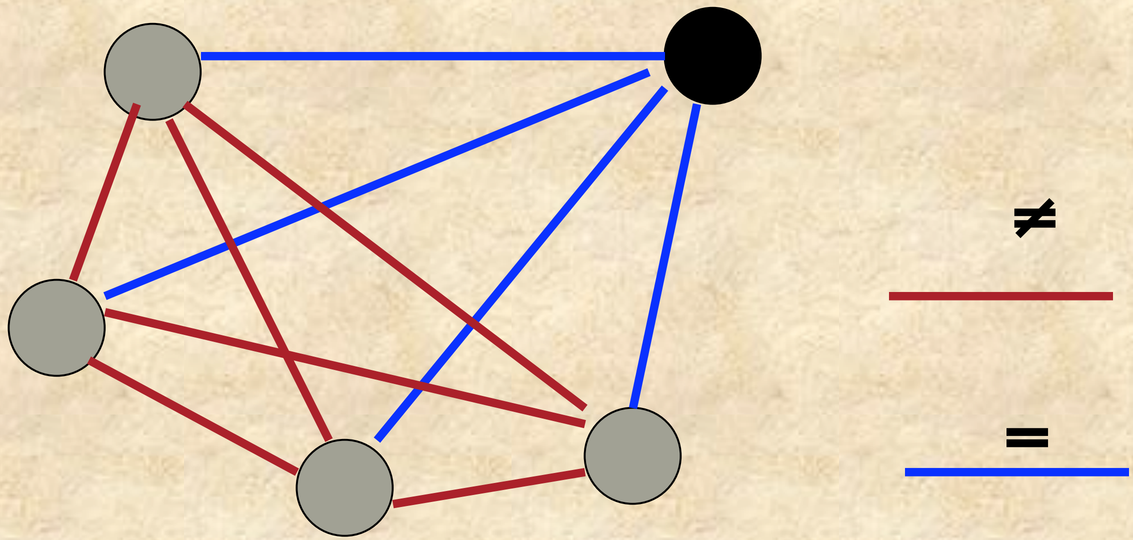


Two types of inconsistencies

Greedy Algorithm

- Pick a vertex u arbitrarily
- Create a cluster C containing all the vertices similar to u, along with u
- Remove C, and repeat

Greedy can be bad



Random Greedy

Pick vertex u uniformly at random

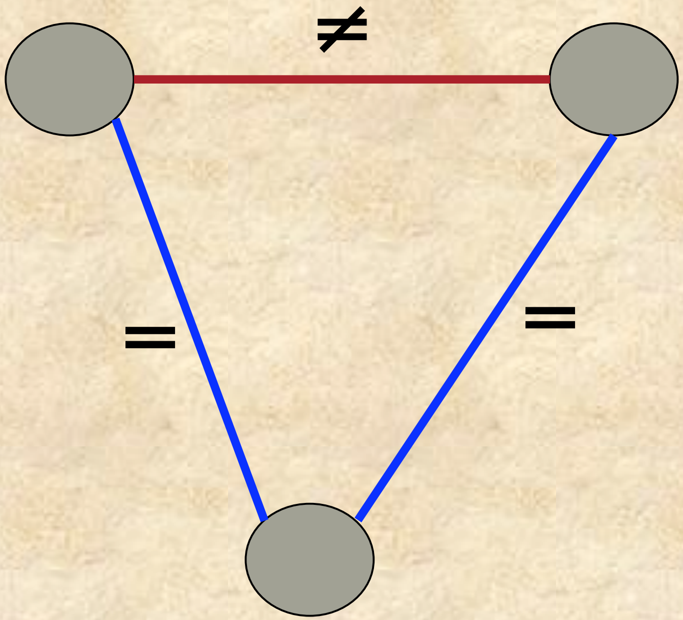
Theorem:

Random Greedy is a 3-approximation

Analysis: Bounding OPT

OPT≥

number of disjoint bad triangles



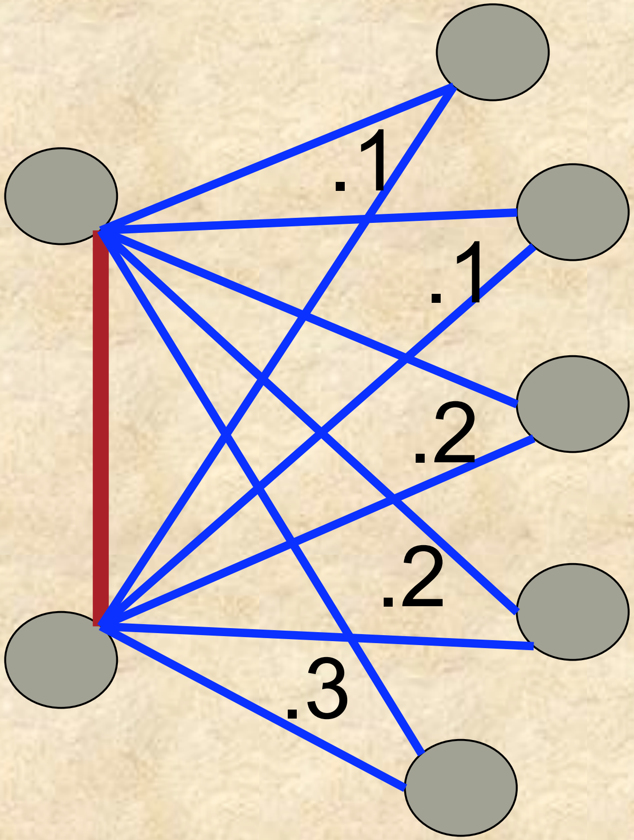
Bounding OPT: bad triangles packing

- Give each bad triangle t a weight a_t
- Such that each edge carries total weight at most 1

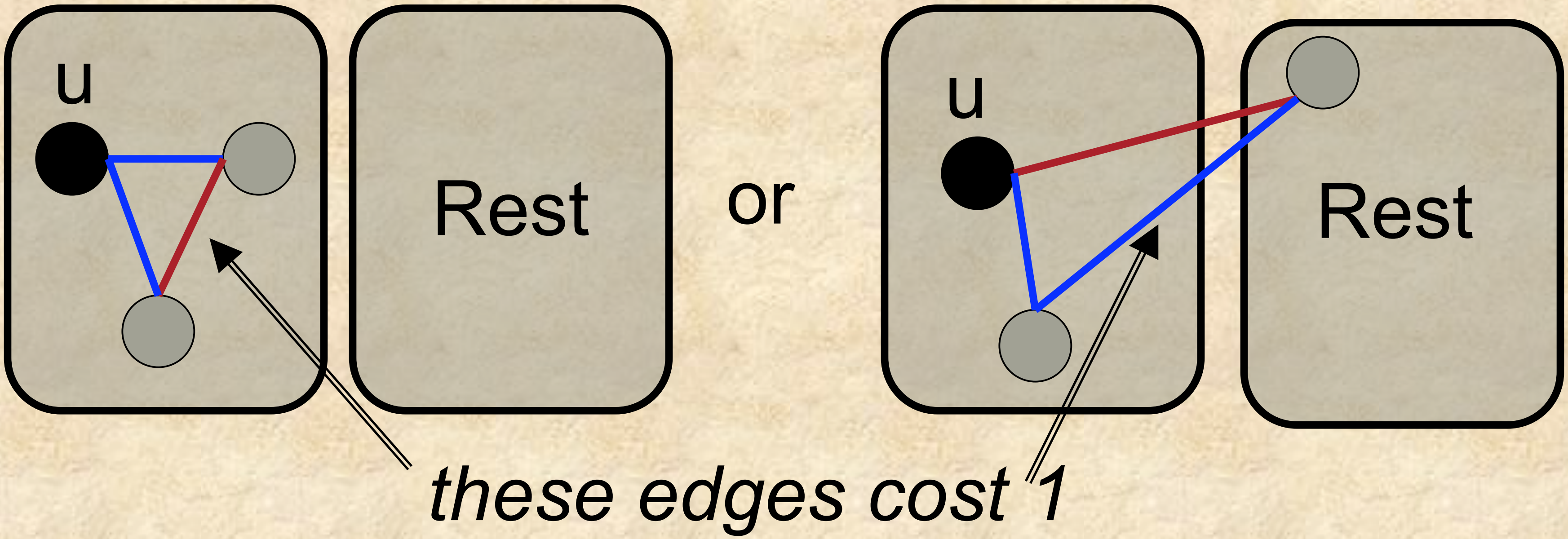
$\Sigma_{t \text{ containing } e} a_t \leq 1$

- Then $\Sigma_t a_t \leq OPT$

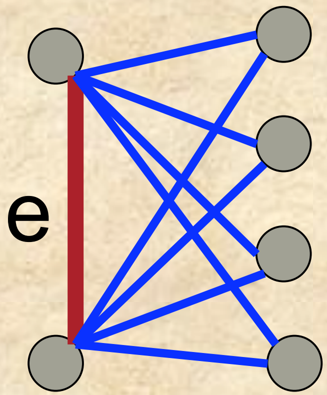
$.1+.1+.2+.2+.3 \leq 1$



Analysis: Bounding Greedy



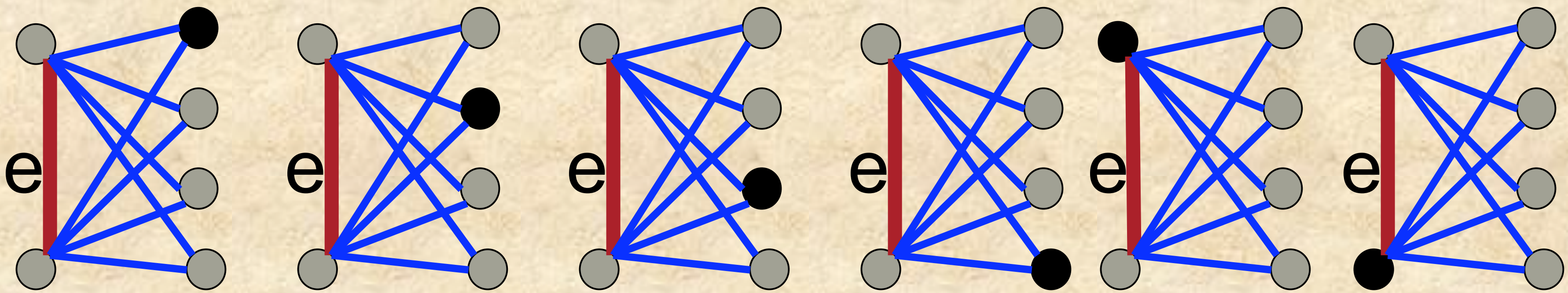
- Let Z_t=whether Greedy destroys bad triangle t by picking one of its three vertices
- Then Cost(Greedy)=$\Sigma_t Z_t$



k bad triangles containing e

Sum Z_t for those triangles

Greedy picks a random vertex



$\Sigma_t Z_t = 1$ 1   1   1   k   k

Weight carried by e: $E(\Sigma_t Z_t) \leq (1+\ldots+1+k+k)/(k+2) \leq 3$

So $a_t = EZ_t / 3$ is a packing of bad triangles

$E(\text{Cost}(\text{Greedy})) = \Sigma_t EZ_t \leq 3$ OPT

Hidden: linear programming duality

Analysis: IP

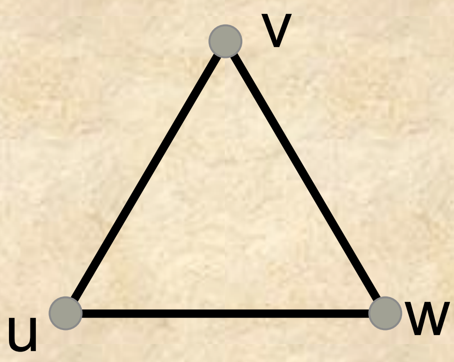
x_{uv}= 1 if u and v are in same cluster

Min $\Sigma_{uv\ dissimilar}\, x_{uv} + \Sigma_{uv\ similar}\,(1-x_{uv})$

Subject to

for all u,v,w : $x_{uv}+x_{vw}+(1-x_{uw})\leq 2$ (uvw consistent)

x_{uv} is 0 or 1



u,v in same cluster
v,w in same cluster
u,w in different clusters
is inconsistent

Using both Primal and Dual

- Dual is implicit in rounding analysis
- Primal is implicit in Alg design

→ Why not do both together?

Primal-dual algorithms

Steiner tree and Steiner forest

Facility location and k-median

…

Online Ski Rental

- Buying skis: B € once.
- Renting skis: 1 € per day.

Online:

Number of ski days not known in advance.

One Algorithm:

Rent, rent, rent, buy.

Goal:

Minimize total cost.

Online IP

$$x = \begin{cases} 1 \text{ - Buy} \\ 0 \text{ - Don't Buy} \end{cases} \qquad z_i = \begin{cases} 1 \text{ - Rent on day i} \\ 0 \text{ - Don't rent on day i} \end{cases}$$

$$\min Bx + \sum_{i=1}^{k} z_i$$

Subject to:

For each day i: $x + z_i \geq 1$

$$x, z_i \in \{0,1\}$$

Online IP: Constraints and variables arrive one by one

LP Relaxation & Dual

P: Primal

$$\min Bx + \sum_{i=1}^{k} z_i$$

For each day i: $x + z_i \geq 1$

$$x, z_i \geq 0$$

D: Dual

$$\max \sum_{i=1}^{k} y_i$$

For each day i: $y_i \leq 1$

$$\sum_{i=1}^{k} y_i \leq B$$

$y_i \geq 0$

Online LP:

- Constraints & variables arrive one by one.
- **Requirement:** Satisfy constraints upon arrival.
- Fractional interpretation: x=.5 means buy one ski, rent the other one

Algorithm for online LP

P: Primal Covering

$$\min Bx + \sum_{i=1}^{k} z_i$$

For each day i: $x + z_i \geq 1$

$$x, z_i \geq 0$$

D: Dual Packing

$$\max \sum_{i=1}^{k} y_i$$

For each day i: $y_i \leq 1$

$$\sum_{i=1}^{k} y_i \leq B$$

$y_i \geq 0$

Initially $x \leftarrow 0$

Each day (new variable z_i, new constraint y_i):

if $x<1$: (skis not yet fully bought)

- $z_i \leftarrow 1-x$ (rent necessary fraction)
- $x \leftarrow x + \Delta x$ (buy a little more)
- $y_i \leftarrow 1$ (update dual, too!)

A 3-point plan

1-2-3

P: Primal

$$\min Bx + \sum_{i=1}^{k} z_i$$

On day i: $x + z_i \geq 1$

$x, z_i \geq 0$

D: Dual

$$\max \sum_{i=1}^{k} y_i$$

On day i: $y_i \leq 1$

$$\sum_{i=1}^{k} y_i \leq B$$

$y_i \geq 0$

1. Primal is feasible.
2. In each iteration, $\Delta P \leq (1 + c)\Delta D$.
3. Dual is feasible.

Then: Output cost $= \Sigma\, \Delta P$

$\leq (1+c)$ Dual — by 2.

$\leq (1+c)$ Opt LP value — by LP duality theorem

$\leq (1+c)$ IP — by LP relaxation

Online LP Algorithm:

On day i:
  if x<1:
    $z_i \leftarrow 1-x$
    $x \leftarrow x + \Delta x$
    $y_i \leftarrow 1$

1. Why is Primal feasible?

On day i: $x + z_i \geq 1$

$$x, z_i \geq 0$$

Online LP Algorithm:

On day i:

    if x<1:

        $z_i \leftarrow 1-x$

        $x \leftarrow x+\Delta x$

        $y_i \leftarrow 1$

2. Why is $\Delta P \le (1+c)\Delta D$?

P: Primal

$$\min Bx + \sum_{i=1}^{k} z_i$$

D: Dual

$$\max \sum_{i=1}^{k} y_i$$

… it depends on Δx.

$\Delta x = x/B + c/B$ works.

Online LP Algorithm:

On day i:

if $x<1$:

$z_i \leftarrow 1-x$

$x \leftarrow x(1+1/B) + c/B$

$y_i \leftarrow 1$

3. Why is Dual feasible?

D: Dual

On day i: $y_i \le 1$

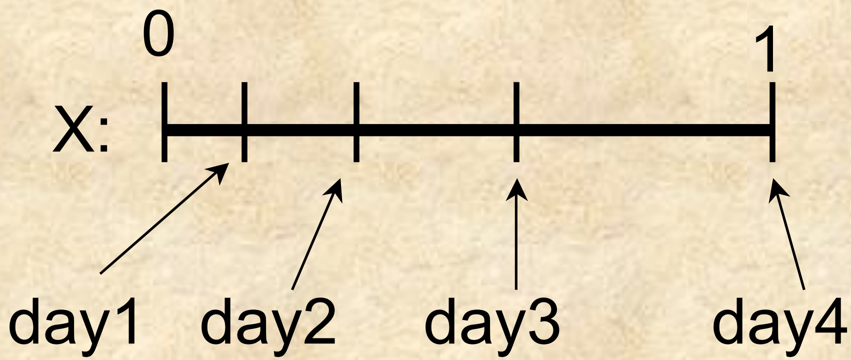
$$\sum_{i=1}^{k} y_i \le B$$

$y_i \ge 0$

… it depends on c.

$c=1/(e-1)$ works ➡ Algorithm $e/(e-1)$ competitive

ONLINE IP ALGORITHM



- Choose (offline) *d* uniformly in [0,1]
- Solve online LP
- Set (online) x=1 on day of "bin" *d* falls in
- Set (online) z_i=1 until then, z_i=0 after

Analysis:

- Prob. That x=1: LP value of x
- Prob. of rental on day i: LP value of z_i

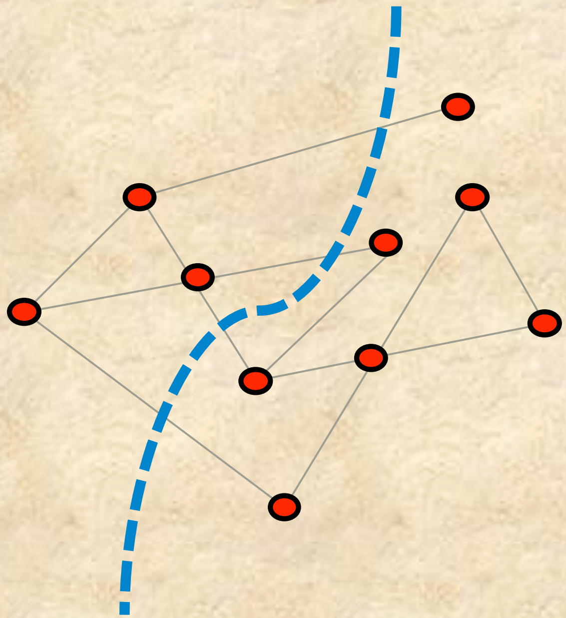
⇒ Competitive ratio = that of online LP Alg

⇒ e/(e-1) competitive algorithm for ski rental

Online primal-dual

- Online set cover
- Virtual circuit routing
- Ad auctions
- Weighted caching
- ...

MaxCut basics



Input: graph

Goal: cut maximum number of edges

Fact: NP-hard

Fact:

Greedy cuts half of the edges: (1/2) approximation.

Question: how to do better?

IP model?

x_i=0 on one side, 1 on the other side of cut

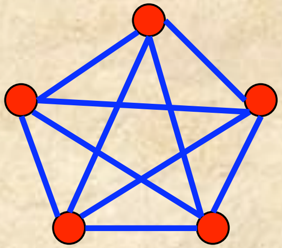
$$\text{Max } \Sigma_e \, d_e$$
$$x_i = 0 \text{ or} 1$$
$$d_{ij} \leq |x_i - x_j|$$

LP Attempts

$\text{Max } \Sigma_e d_e$
$d_{ij} = 0 \text{ or } 1$
$d_{ij} + d_{jk} + d_{ki} \leq 2$
$d_{ij} \leq d_{jk} + d_{ki}$

Integrality gap = 2

$\text{Max } \Sigma_e d_e$
$0 \leq d_{ij} \leq 1$
$d_{ij} + d_{jk} + d_{ki} \leq 2$
$d_{ij} \leq d_{jk} + d_{ki}$



$\Sigma d_{ij} \leq 6$

Add pentagonal constraints: does not help

Add odd cycle constraints: does not help

Add bounded support constraints: does not help

Bounded degree expander with large girth

$\text{Max } \Sigma_e \, d_e$
$x_i = 0 \text{ or } 1$
$d_{ij} \leq |x_i - x_j|$

$\text{Max } \Sigma_{ij \text{ in } E} \, (x_i - x_j)^2$
$x_i = 0 \text{ or } 1$

$\text{Max } (1/4) \, \Sigma_{ij \text{ in } E} \, (x_i - x_j)^2$
$x_i = -1 \text{ or } 1$

$\text{Max } (1/2) \, \Sigma_{ij \text{ in } E} \, 1 - x_i x_j$
$x_i = -1 \text{ or } 1$

$\text{Max } (1/2) \, \Sigma_{ij \text{ in } E} \, 1 - v_i v_j$
$|v_i|^2 = 1$

$\text{Max } (1/2) \, \Sigma_{ij \text{ in } E} \, 1 - y_{ij}$
$y_{ii} = 1$
Y positive semidefinite

M positive semi-definite

M real symmetric.

Three equivalent conditions:

- $M = V^T V$
- All eigenvalues of M are ≥ 0
- For every vector a:

$$a^T M a \geq 0$$

MaxCut Algorithm

1. Solve SDP relaxation

How?

$$\text{Max } (1/2)\, \Sigma_{ij \text{ in } E}\ 1-y_{ij}$$
$$y_{ii}=1$$
$$Y \text{ positive semidefinite}$$

2. Round result to get cut

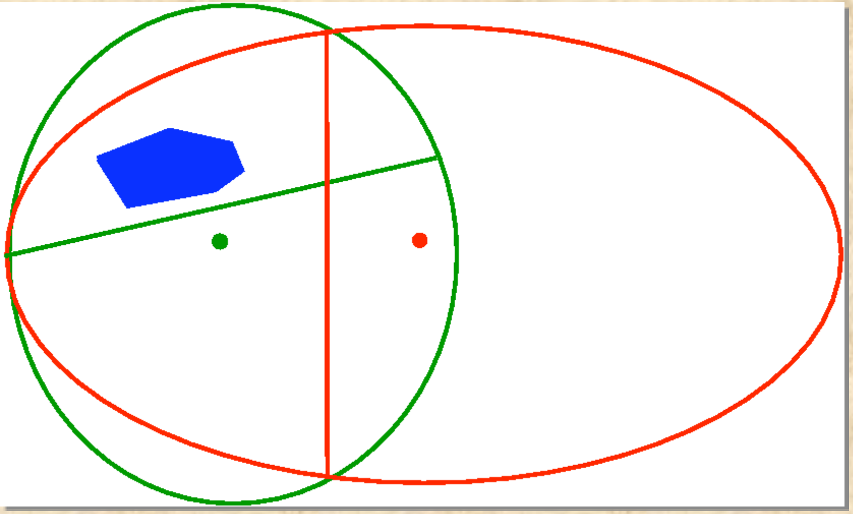
Max (1/2) $\Sigma_{ij \text{ in } E}$ 1-y_{ij}

y_{ii}=1

Y symmetric

For every vector a: $a^TYa \geq 0$

Linear in (y_{ij})



Ellipsoid method

Polynomial time

Oracle-Based:

Q: “Y feasible?”

A: “yes” or “No since [linear inequality] ”

Max (1/2) $\Sigma_{ij \text{ in } E}$ 1-y_{ij}
$y_{ii}=1$
Y symmetric
For every vector a: $a^T Y a \geq 0$

Oracle-Based:
Q: "Y feasible?"
A: "yes" or "No since [linear inequality] "

Max (1/2) $\Sigma_{ij \text{ in } E}$ 1-y_{ij}
$y_{ii}=1$
Y symmetric
eigenvalues≥0

Linear in (y_{ij})

Oracle: Compute eigenvalues
if $\alpha<0$, compute eigenvector u: $u^T Y u = \alpha|u|^2 < 0$

MaxCut Algorithm

1. Solve SDP relaxation

Max (1/2) $\Sigma_{ij \text{ in } E}$ 1-y_{ij}
y_{ii}=1
Y positive semidefinite

✔

2. Round result to get cut

How?

Max (1/2) $\Sigma_{ij \text{ in } E}$ 1-$v_i v_j$
$|v_i|^2$= 1

Vertices → Unit vectors

↓

Cut

Rounding the SDP

Max (1/2) $\Sigma_{ij \text{ in } E}$ $1 - v_i v_j$

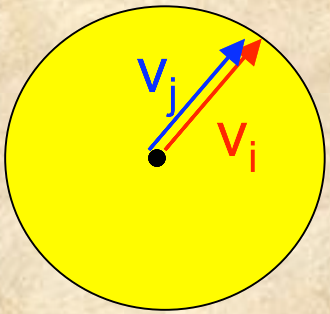
$|v_i|^2 = 1$

Vertices → Unit vectors

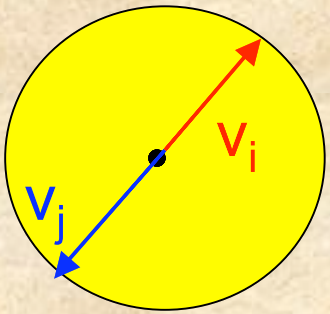
↓

Cut

0 if $v_i = v_j$

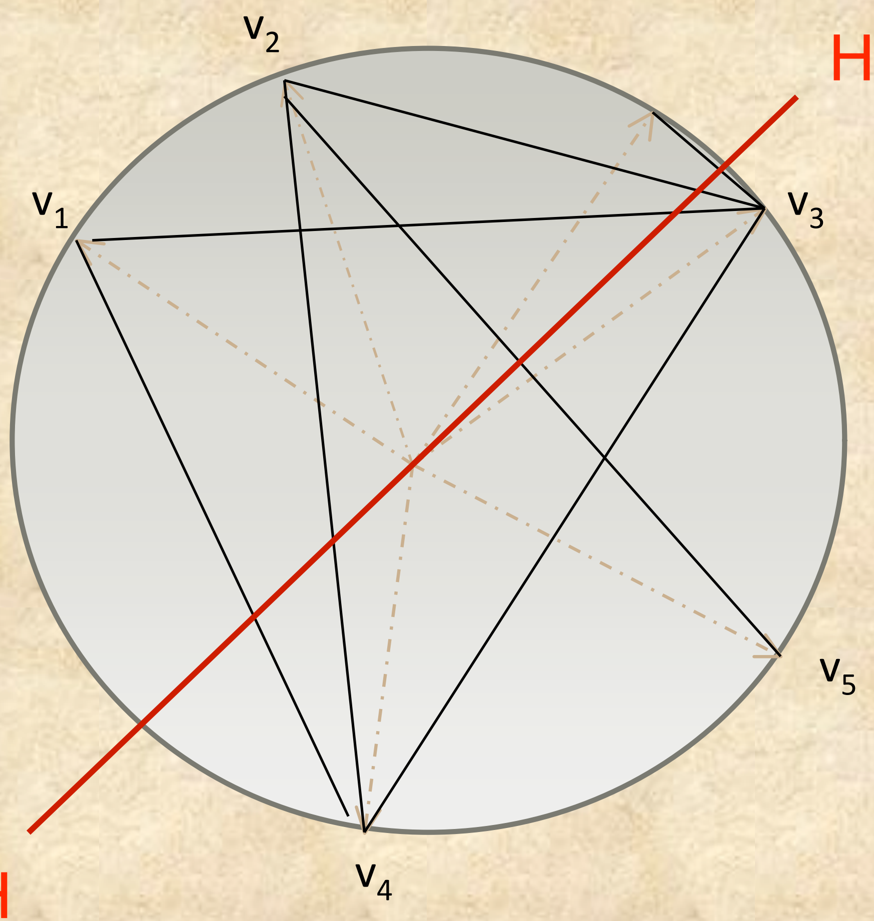


1 if $v_j = -v_i$



If v_i and v_j are close then i and j should end up on same side of graph cut

If v_i and v_j are close then i and j should end up on same side of graph cut



Take a random hyperplane H Through the center of the sphere.

Graph cut:
L={i: v_i is above H}
R={i: v_i is below H}

MaxCut Algorithm

1. Solve SDP relaxation

Max (1/2) $\Sigma_{ij \text{ in } E}$ 1-y_{ij}
y_{ii}=1
Y positive semidefinite

2. Round result to get cut

Max (1/2) $\Sigma_{ij \text{ in } E}$ 1-$v_i v_j$
$|v_i|^2$= 1

Take random hyperplane H through center of sphere.
Output: L={i: v_i is above H}, R={i: v_i is below H}

Analysis

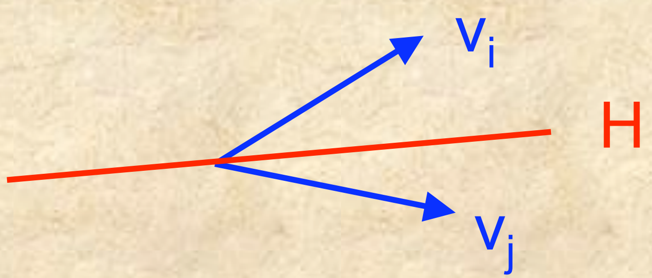
SDP relaxation

Max $(1/2)\ \Sigma_{ij \text{ in } E}\ 1\text{-}v_i v_j$
$|v_i|^2 = 1$

We have:
$OPT \geq (1/2)\ \Sigma_{ij \text{ in } E}\ 1\text{-}v_i v_j$

Rounding
E(cut size)= $\Sigma_{ij \text{ in } E}$ Pr(ij in cut)

Pr(ij in cut)= Pr(H between v_i and v_j)



For random H this equals …

SDP relaxation

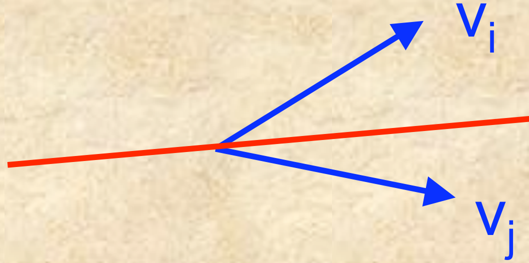
$$\text{Max } (1/2)\, \Sigma_{ij \text{ in } E}\, 1 - v_i v_j$$
$$|v_i|^2 = 1$$

Rounding

E(cut size)=

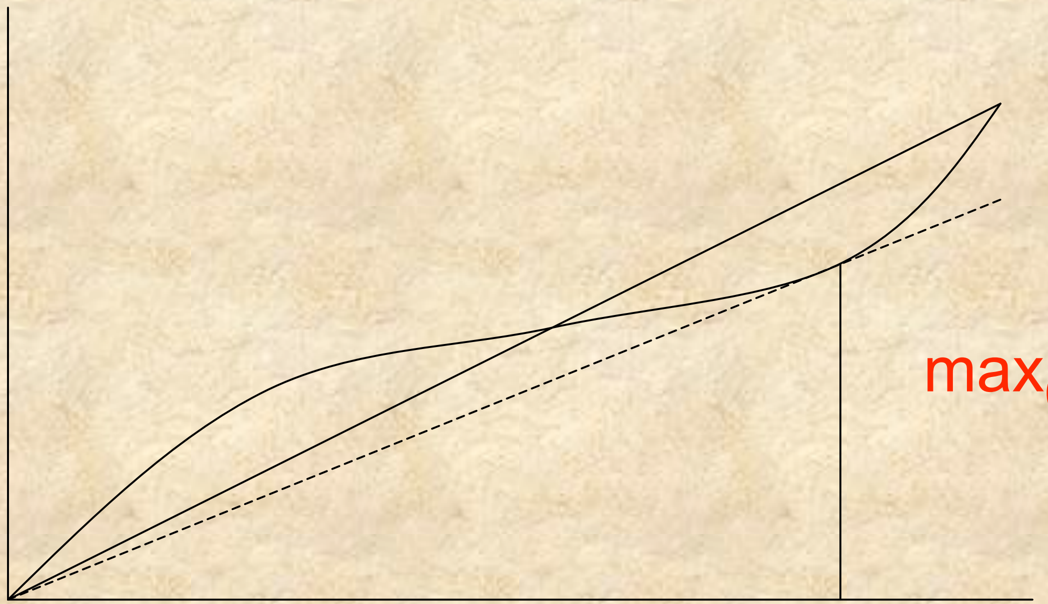
$\Sigma_{ij \text{ in } E}$ Pr(H between v_i and v_j)=

θ_{ij}/π



We have:

OPT$\geq (1/2)\, \Sigma_{ij \text{ in } E}\, 1 - \cos(\theta_{ij})$



$$\max_\theta \frac{\theta/\pi}{1-\cos(\theta)} = 0.878\ldots$$

SDP Algorithms

- MaxCut
- Max-k-Sat
- Coloring
- Scheduling (completion times)
- CSP
- Sparsest Cut
- …

Hardness of MaxCut

Assuming P≠NP and UGC, 0.878 is the best possible approximation ratio for MaxCut

Unique Games Conjecture (UGC)

Input: 2 variables per equation

7x+2y = 11 (mod 23)

5x+3z = 8 (mod 23)

...

....

7z+w = 14(mod 23)

Goal: maximize number of satisfied equations

UGC Conjecture: NP-hard to distinguish between

answer >99% and answer <1%.

Fix ε. For p large, NP-hard to distinguish 1-ε from ε

Uses of UGC

Problem	Best Approximation Algorithm	NP Hardness	Unique Games Hardness
Vertex Cover	2	1.36	2
Max CUT	0.878	0.941	0.878
Max 2- SAT	0.9401	0.9546	0.9401
SPARSEST CUT	$\sqrt{\log n}$	1+ε	Every Constant
Max k-CSP	$\Omega(k/2^k)$	$O(2^{\sqrt{k}}/2^k)$	$O(k/2^k)$

UGC hardness results are intimately connected to the limitations of Semidefinite Programming

- Multiway cut: Calinescu, Karloff, Rabani 1998, Karger, Klein, Stein, Thorup, Young 1999
- Vehicle routing: work in progress
- Correlation clustering: Ailon Charikar Newman 2005
- Online ski rental: by primal-dual, Buchbinder Naor 2009
- Maxcut: Goemans Williamson 1994
- UGC: Khot 2002
- Hardness of MaxCut: Khot Kindler Mossel O'Donnel 2005

Foundational: LP+randomized rounding (Raghavan Thompson 1988), primal-dual (Aggarwal Klein Ravi, Goemans Williamson), SDP (Goemans Williamson)

Updated some slides from Neal Young, LP example from Vazirani's textbook, slides from Seffi Naor, and a couple of slides from Raghavendra.