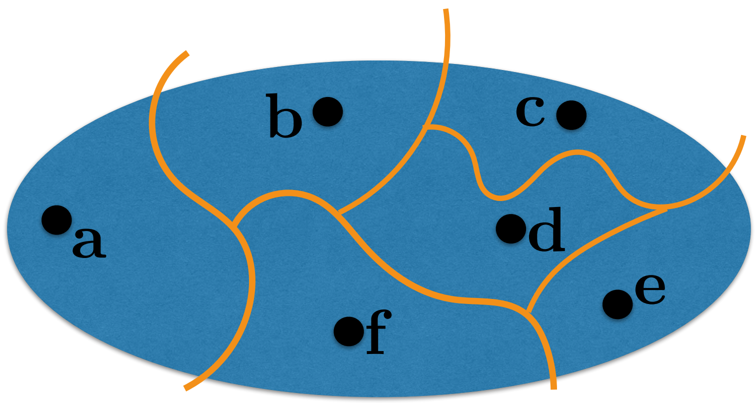
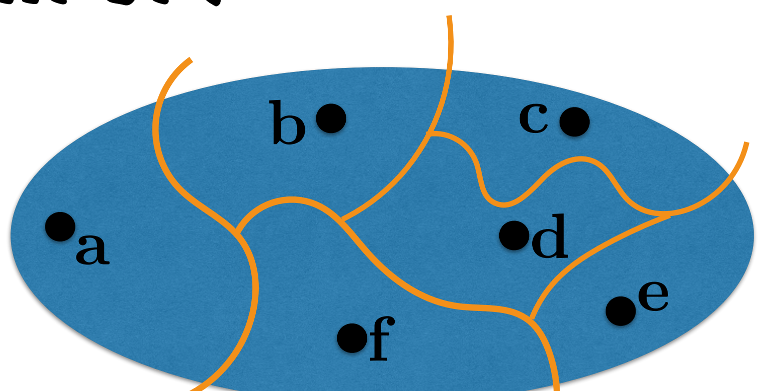
Multiway cut, linear programming and randomized rounding



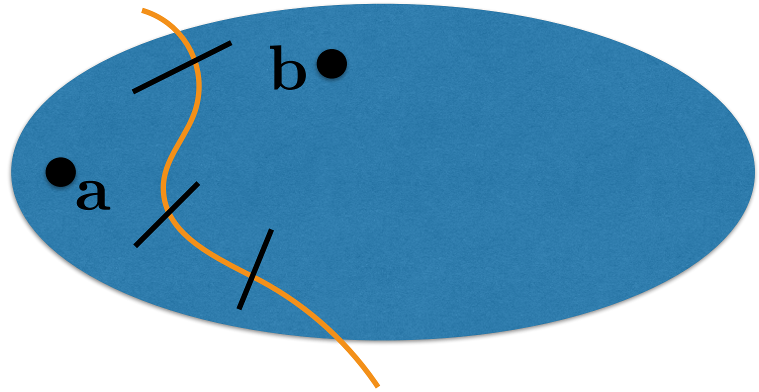
Problem

Let k be a fixed integer.
Given: graph G with edge weights,
k vertices called "terminals"
Find: subset F of edges such that
Removing F disconnects the
terminals from one another
Goal: minimize weight of F



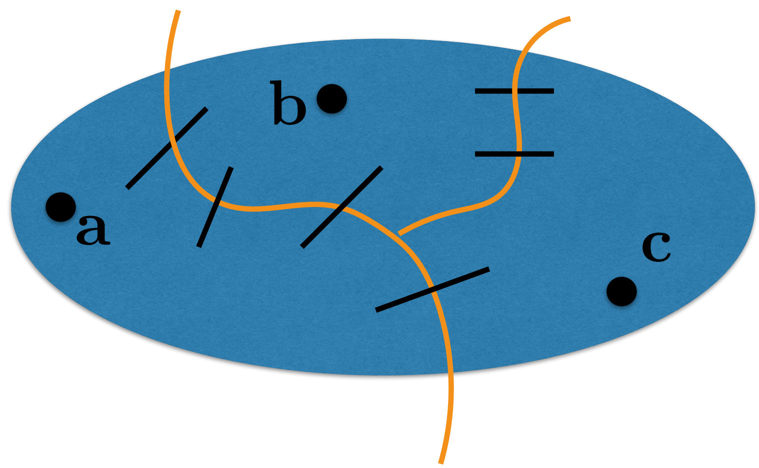
k=2

min cut in P



k=3

NP-hard



Simple algorithm for k=3

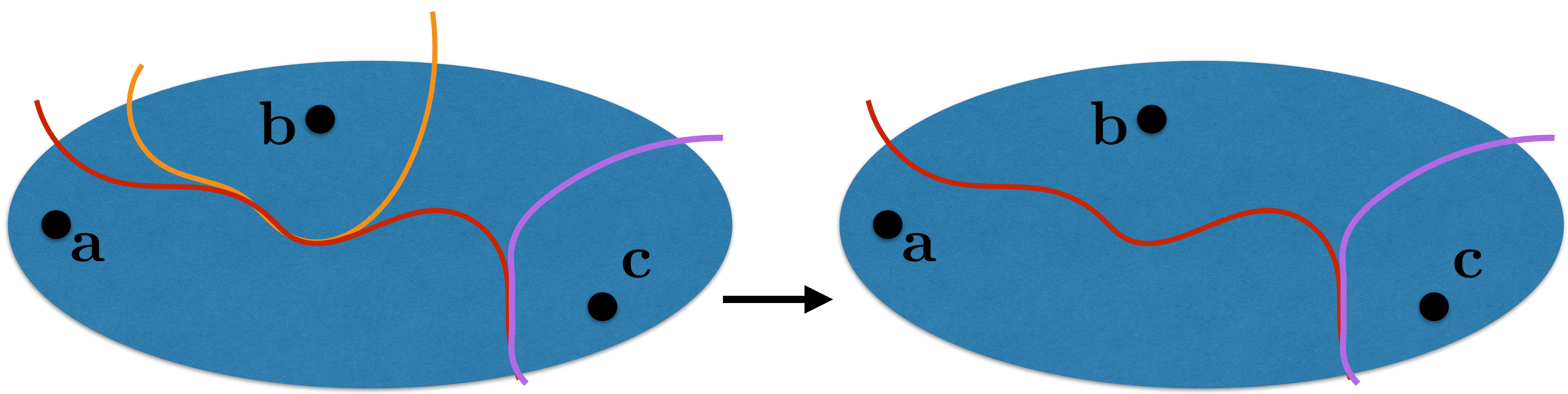
Terminals: a,b,c

Output the two smallest of

{Mincut(a,{b,c}),

Mincut(b,{c,a}),

Mincut(c,{a,b})}



Analysis

- It takes polynomial time...
- It's a correct multiway cut: a,b,c are separated from one another
- But how good is it?

Cost of output

Output is at most

(2/3) (Mincut(a,bc)+Mincut(b,ac)+Mincut(c,ab))

- OPT is at least Mincut(a,bc)
- OPT is at least Mincut(b,ac)
- OPT is at least Mincut(c,ab)

So OPT is at least

(Mincut(a,bc)+Mincut(b,ac)+Mincut(c,ab))/3

Alg is a 2 approximation

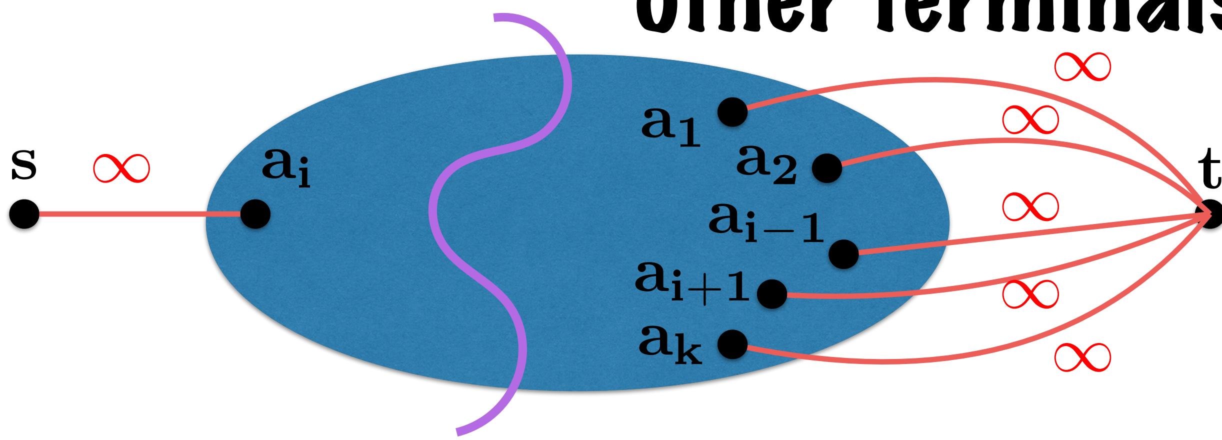
Extension: algorithm for k

Terminals: a1,a2,...,ak

Output the k-1 smallest of

{Mincut(ai,{a1,...,ai-1,ai+1,... ak}): i=1,2,...,k}

To compute the min cut separation of ai from the other terminals:



Mincut(s,t)

Analysis of output

The k-1 smallest cuts cost less than a random choice of k-1, so

$$\mathrm{Cost(Output)} \leq \frac{\mathbf{k-1}}{\mathbf{k}} \sum_{\mathbf{i}} \mathrm{Mincut}(\mathbf{a_i}, \{\mathbf{a_1}, \ldots, \mathbf{a_{i-1}}, \mathbf{a_{i+1}}, \ldots, \mathbf{a_k}\})$$

Analysis of OPT

Let F(i)={edges of OPT that separate ai from some aj}

Consider e in OPT

e separates some ai from some aj

e belongs to F(i) and to F(j)

so

$$\sum_i \mathrm{Cost}(F_i) = 2\mathrm{OPT}$$

F(i)={edges of OPT that separate ai from some aj}

F(i) separates ai from all other terminals so

$$\mathrm{Cost}(F_i) \geq \mathrm{Mincut}(a_i, \{a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_k\})$$

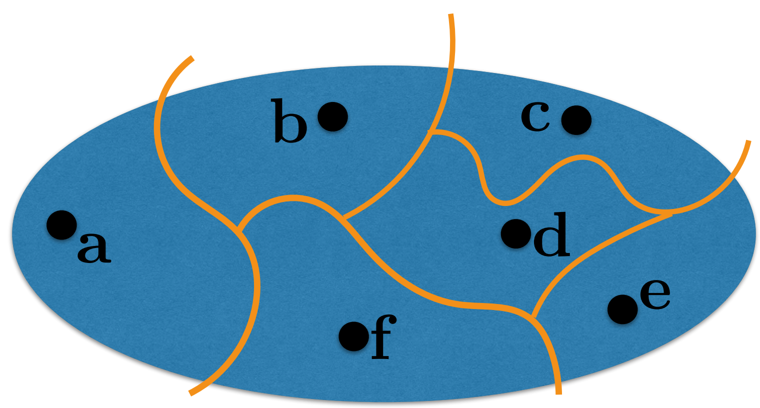
Together

$$
\begin{aligned}
&\mathbf{Cost(Output)} \\
&\le \frac{\mathbf{k-1}}{\mathbf{k}} \sum_{\mathbf{i}} \mathbf{Cost(F_i)} \\
&\le \mathbf{2(1 - \frac{1}{k})OPT}
\end{aligned}
$$

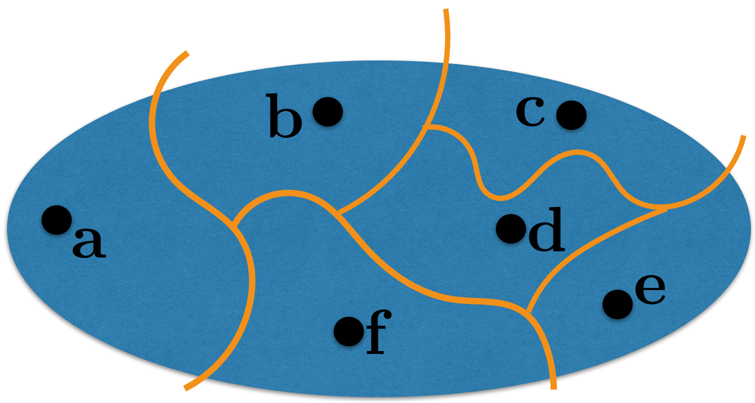
k=2: Optimal

k=3: it's a 4/3 approximation

Multiway cut, linear programming and randomized rounding



Multiway cut, linear programming and randomized rounding

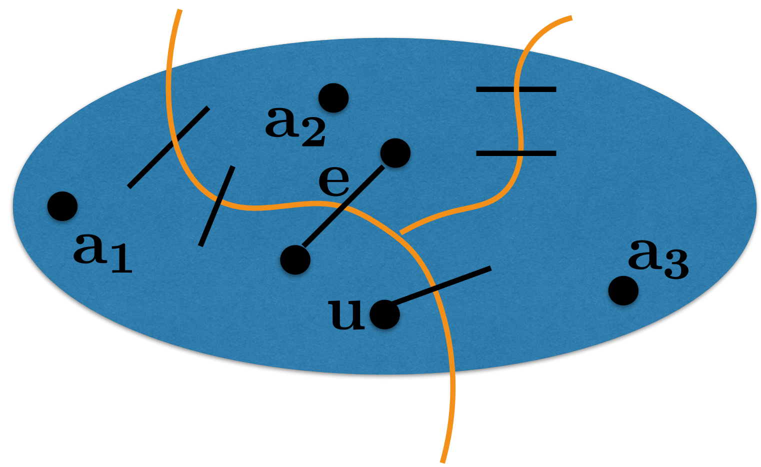


IP model

Variables

$\mathbf{x_{u,i} = 1}$ iff
vertex u belongs to the cluster
of terminal ai.

$\mathbf{z_{e,i} = 1}$ iff
removal of edge e separates ai from some other terminal



IP model

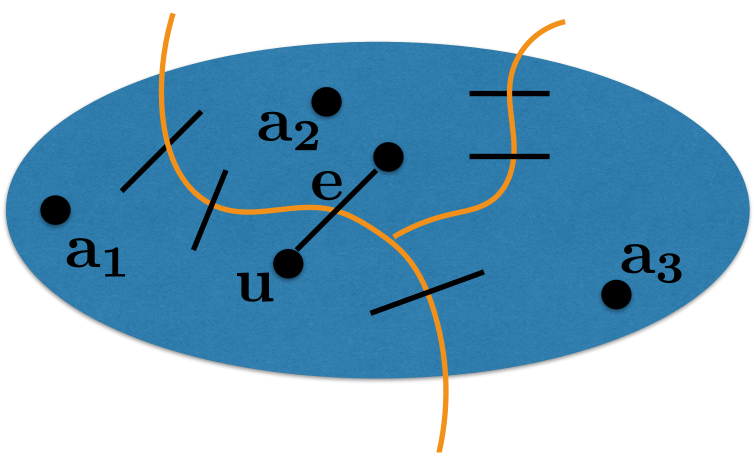
Constraints

$$\mathbf{x_{u,i}, z_{e,i} \in \{0,1\}}$$

$$\mathbf{x_{a_i,i} = 1 \ \ (a_i \text{ belongs to its own cluster})}$$

$$\mathbf{\textstyle\sum_i x_{u,i} = 1 \ \ (u \text{ belongs to some cluster})}$$

$\mathbf{z_{uv,i} = 1}$ **iff** $\mathbf{x_{u,i} \neq x_{v,i}}$ **???**

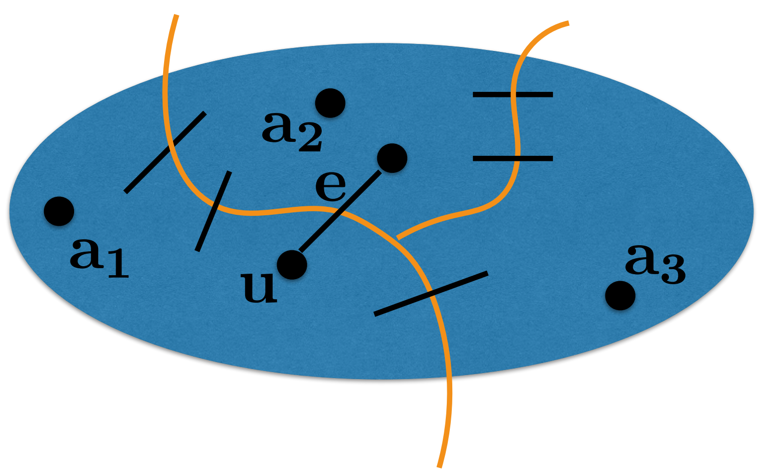


$\mathbf{z_{uv,i} = 1}$ **iff** $\mathbf{x_{u,i} \neq x_{v,i}}$

IP model

Objective

$$\min \quad \mathbf{(1/2)} \textstyle\sum_{\mathbf{e,i}} \mathbf{c_e z_{e,i}}$$



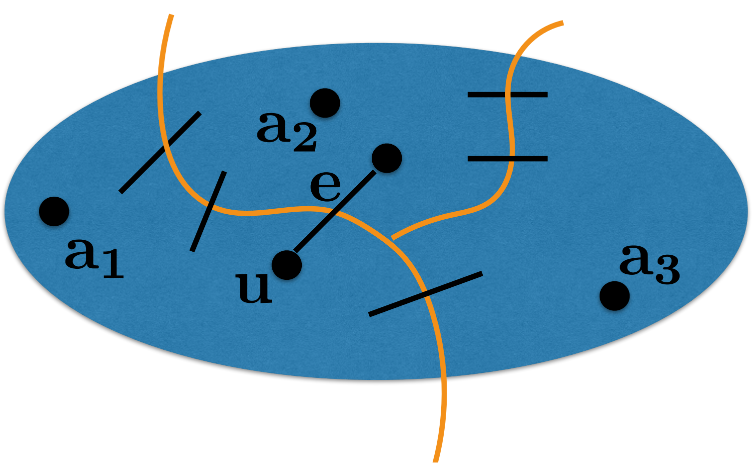
$$\mathbf{z_{uv,i} = 1} \quad \textbf{iff} \quad \mathbf{x_{u,i} \neq x_{v,i}}$$

IP model

How do we express that constraint?

$$\mathbf{z_{e,i} \geq |x_{u,i} - x_{v,i}|}$$

$$\mathbf{z_{e,i} \geq x_{u,i} - x_{v,i}}$$
$$\mathbf{z_{e,i} \geq x_{v,i} - x_{u,i}}$$



IP model

$$\mathbf{x_{u,i}, z_{e,i}} \in \{\mathbf{0}, \mathbf{1}\}$$

$$\mathbf{x_{a_i,i}} = \mathbf{1}$$

$$\sum_{\mathbf{i}} \mathbf{x_{u,i}} = \mathbf{1}$$

$$\mathbf{z_{e,i}} \geq \mathbf{x_{u,i}} - \mathbf{x_{v,i}}$$

$$\mathbf{z_{e,i}} \geq \mathbf{x_{v,i}} - \mathbf{x_{u,i}}$$

$$\min \quad (\mathbf{1}/\mathbf{2}) \sum_{\mathbf{e,i}} \mathbf{c_e z_{e,i}}$$

Linear programming relaxation

$$\mathbf{x_{u,i}, z_{e,i} \in \{0,1\}} \longrightarrow \mathbf{0 \leq x_{u,i}, z_{e,i} \leq 1}$$

$$\mathbf{x_{a_i,i} = 1}$$

$$\mathbf{\textstyle\sum_i x_{u,i} = 1}$$

$$\mathbf{z_{e,i} \geq x_{u,i} - x_{v,i}}$$

$$\mathbf{z_{e,i} \geq x_{v,i} - x_{u,i}}$$

$$\min \quad \mathbf{(1/2) \textstyle\sum_{e,i} c_e z_{e,i}}$$

A geometric interpretation

Variables

$\mathbf{x_{u,i}} = 1$ iff vertex u belongs to the cluster of terminal ai.

$$\text{Vector } \mathbf{x_u} = (\mathbf{x_{u,i}})_\mathbf{i}$$

One dimension for each terminal

$$\sum_\mathbf{i} \mathbf{x_{u,i}} = \sum_\mathbf{i} |\mathbf{x_{u,i}}| = |\mathbf{x_u}|_\mathbf{1}$$

A geometric interpretation

Vector $\mathbf{x_u = (x_{u,i})_i}$

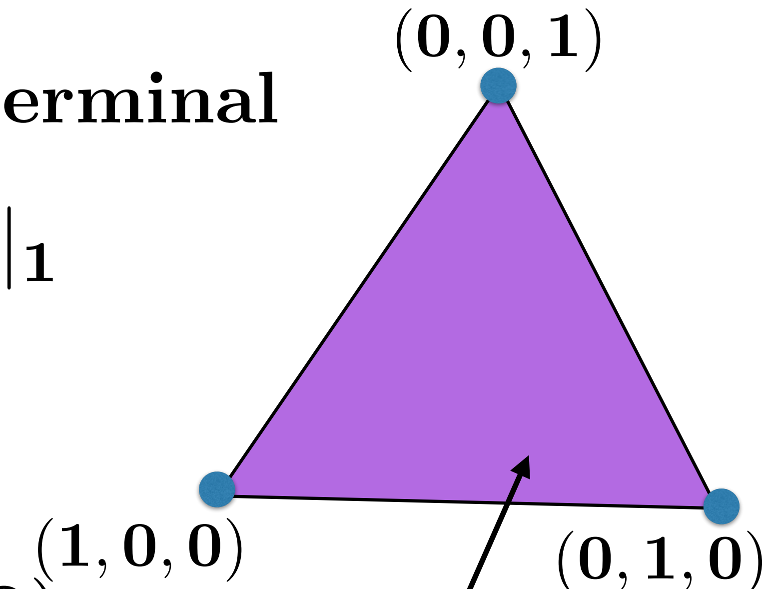
One dimension for each terminal

$$\mathbf{\sum_i x_{u,i} = \sum_i |x_{u,i}| = |x_u|_1}$$

$$\mathbf{|x_u|_1 = 1 \quad \forall u}$$

$$\mathbf{x_u \in [0,1]^k \forall u}$$

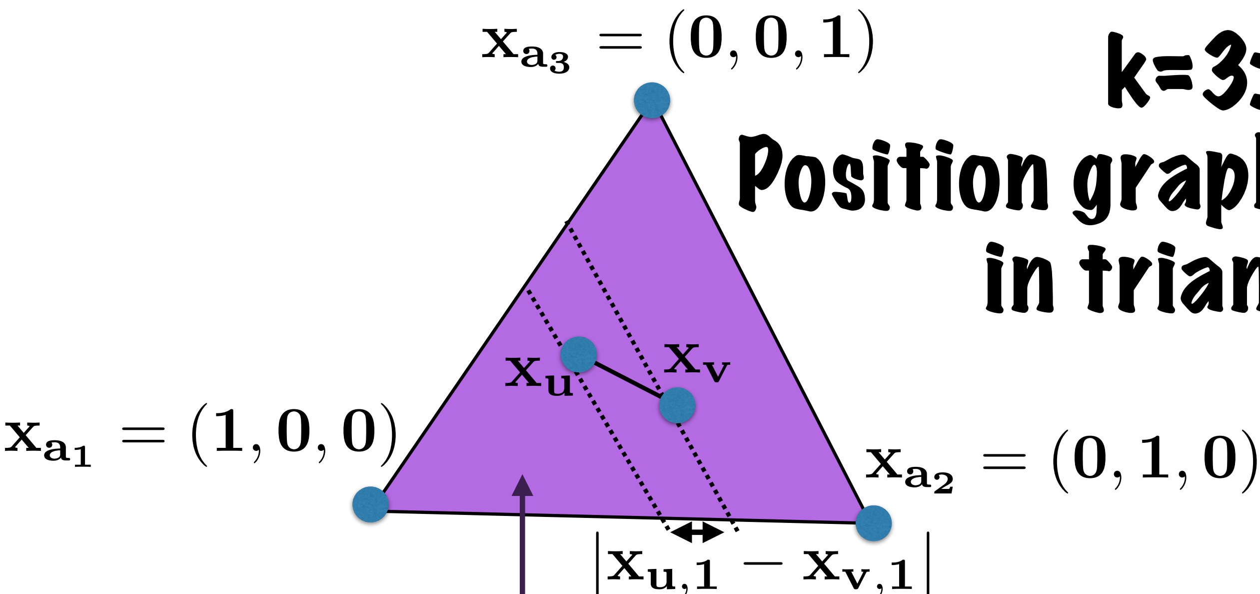
$$\mathbf{x_{a_i} = (0, \ldots, 0, 1, 0, \ldots 0)}$$



$$\begin{cases} \mathbf{x_1 + x_2 + x_3 = 1} \\ \mathbf{x_i \geq 0} \end{cases}$$

$$\min \quad \mathbf{(1/2) \sum_{uv \in E} c_{uv} |x_u - x_v|_1}$$

$$\min \quad (1/2)\sum_{uv\in E} c_{uv}|x_u - x_v|_1$$



$$\begin{cases} x_1 + x_2 + x_3 = 1 \\ x_i \geq 0 \end{cases}$$

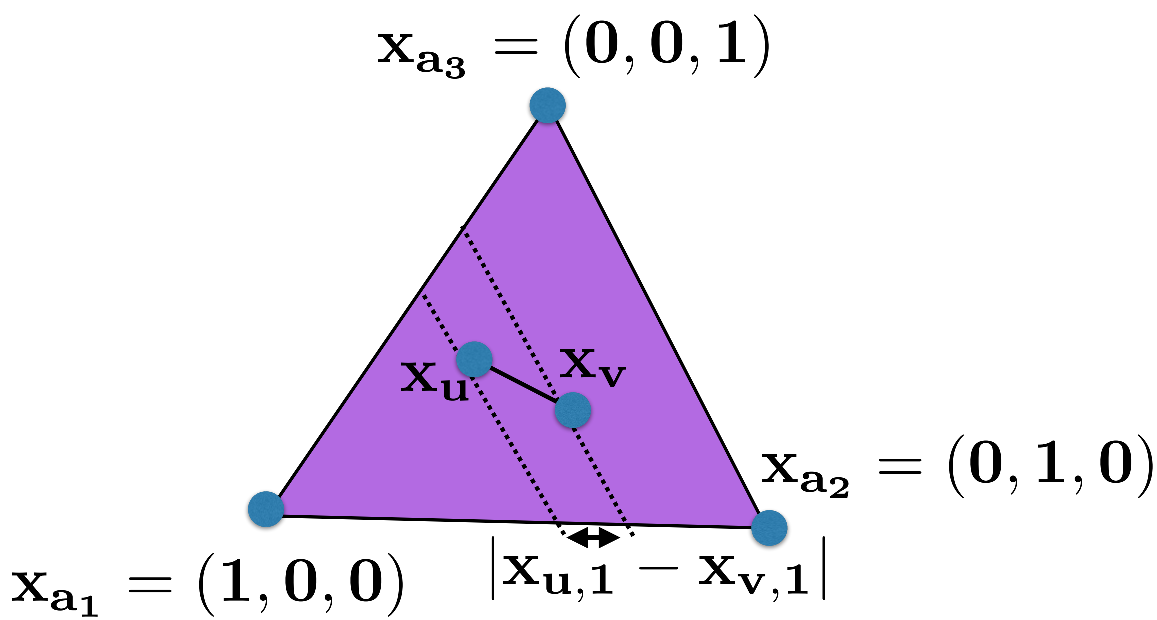
k=3:

Position graph vertices in triangle

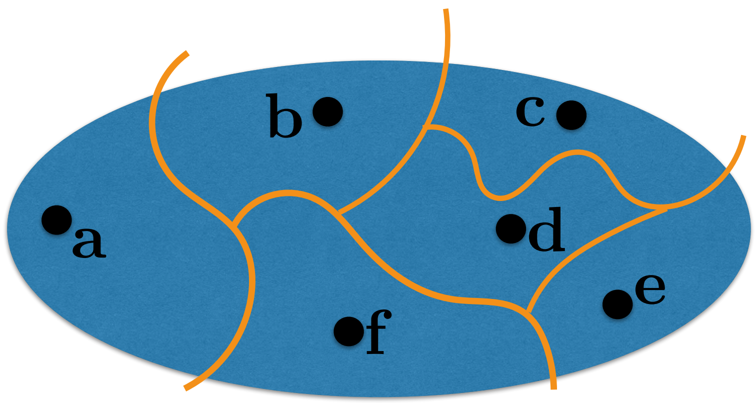
to minimize weighted lengths of projections on sides

A geometric interpretation (k=3)

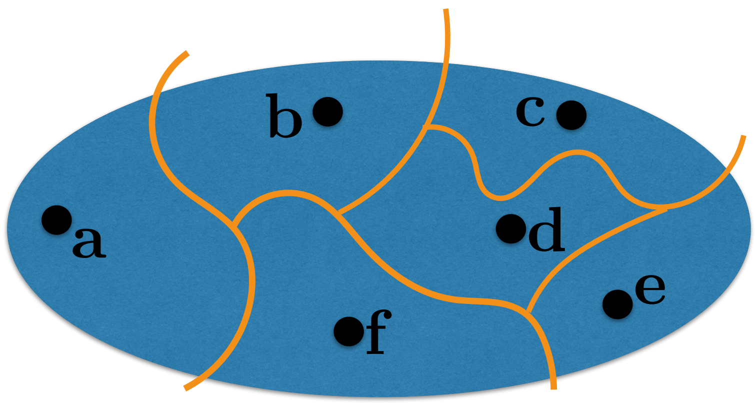
Position graph vertices in triangle to minimize weighted lengths of projections of graph edges on the sides



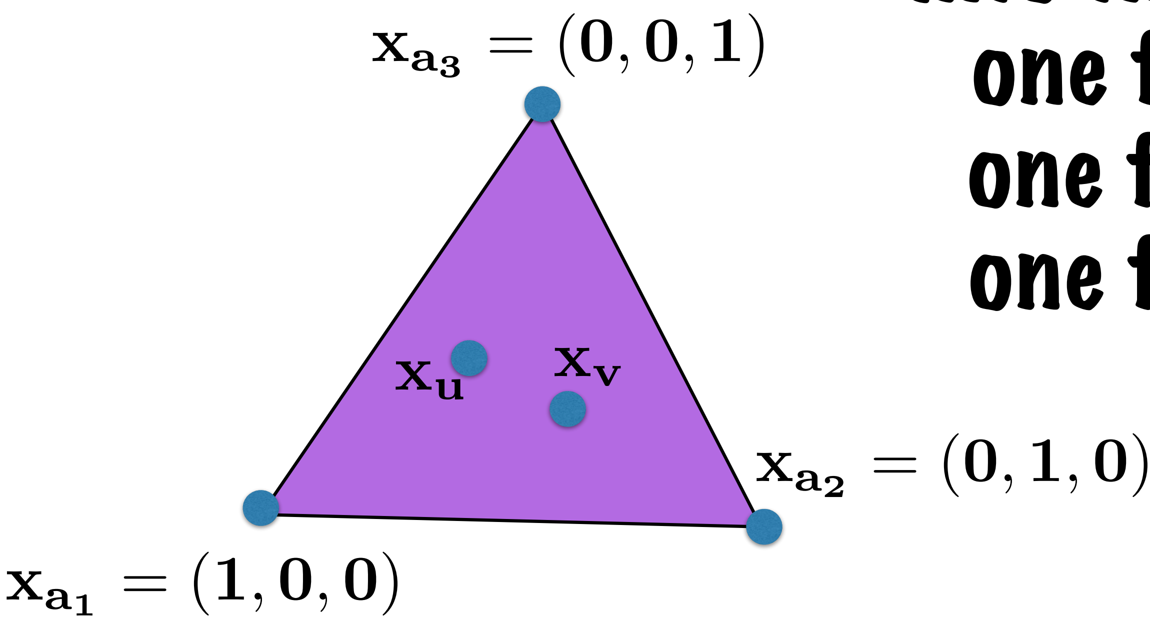
Multiway cut, linear programming and randomized rounding



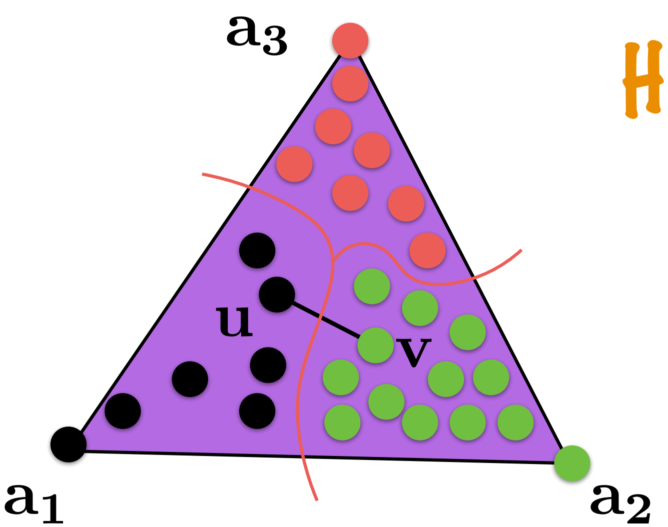
Multiway cut, linear programming and randomized rounding



How do we round?



Partition triangle into three areas:
one for (0,0,1)
one for (1,0,0)
one for (0,1,0)



How do we round?

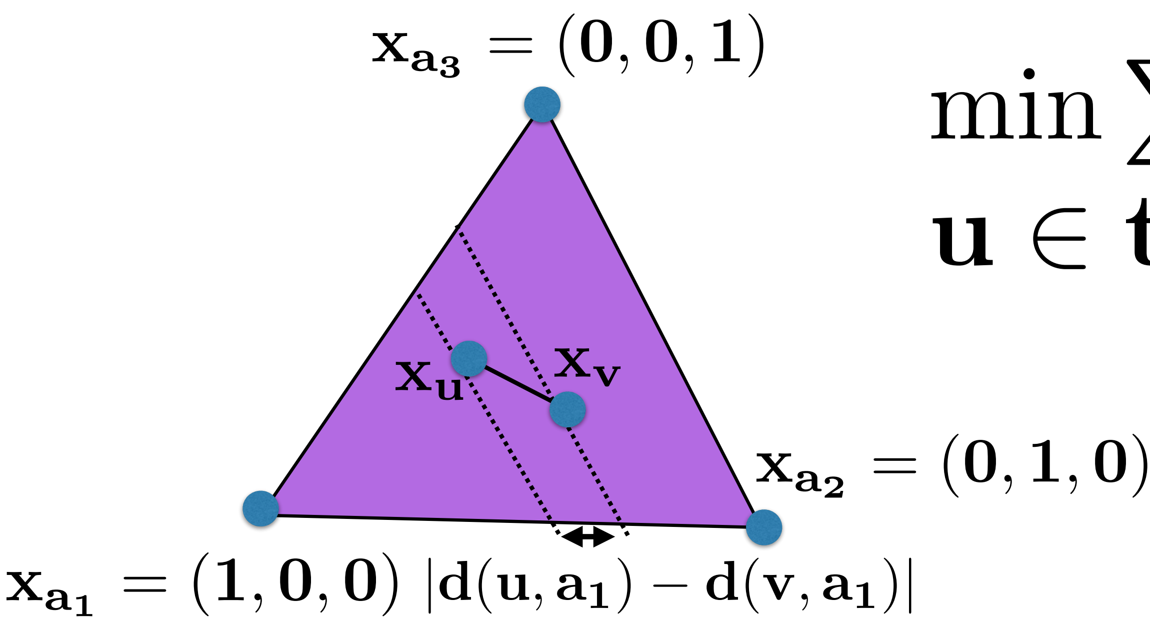
Vertices ● go with a1
Vertices ● go with a2
Vertices ● go with a3
Pay cost of edges across

Good rounding = small cost choice of partition of triangle into three areas

LP relaxation

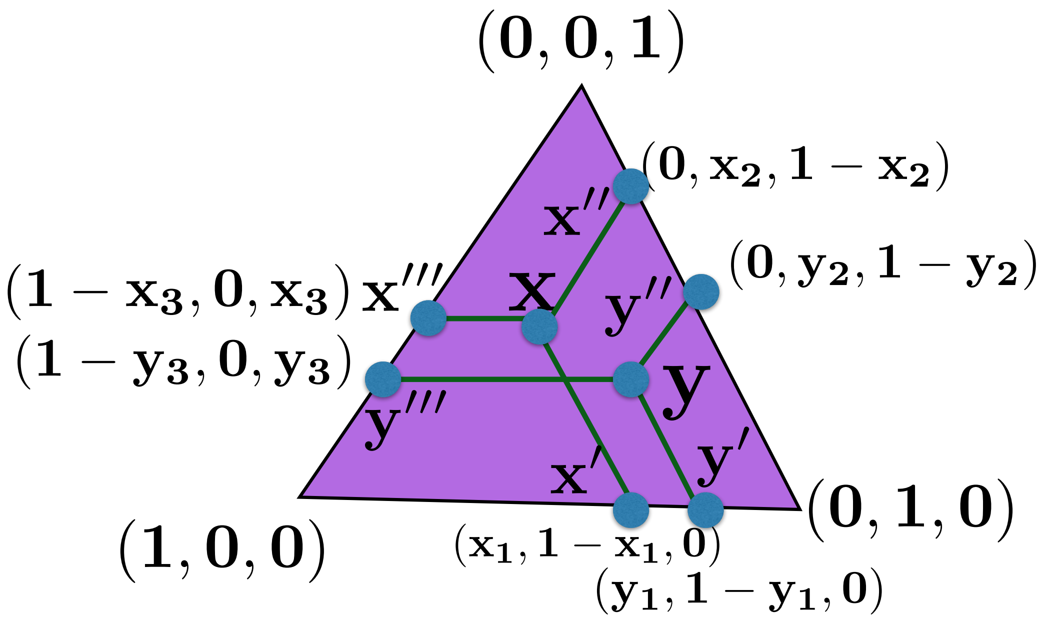
Place vertices in triangle, min lengths of edge projections on sides

$$\mathbf{d(u,v) = \tfrac{1}{2}|x_u - x_v|_1}$$



$$\min \sum_{\mathbf{uv \in E}} \mathbf{c_{uv} d(u,v)} :$$
$$\mathbf{u \in triangle} \quad \forall \mathbf{u}$$

How do we round?



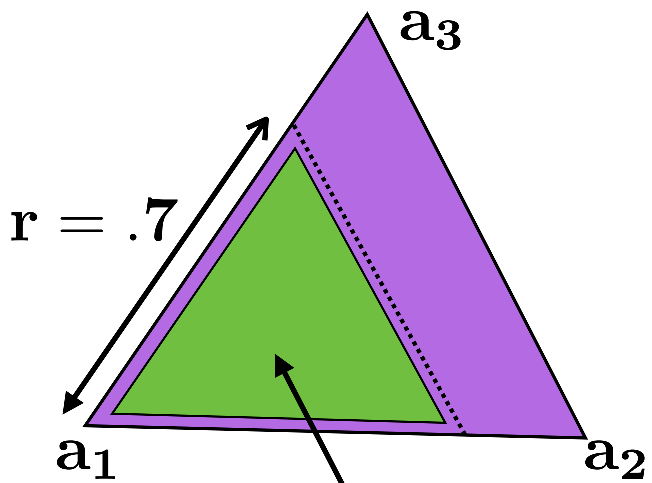
$$\mathbf{d(x', y') = |x_1 - y_1|}$$

$$\begin{aligned}\mathbf{d(u, v)} &= \tfrac{1}{2}(|\mathbf{x_1} - \mathbf{y_1}| + |\mathbf{x_2} - \mathbf{y_2}| + |\mathbf{x_3} - \mathbf{y_3}|) \\ &= \tfrac{1}{2}(\mathbf{d(x', y')} + \mathbf{d(x'', y'')} + \mathbf{d(x''', y''')})\end{aligned}$$

Using balls in l1 metric

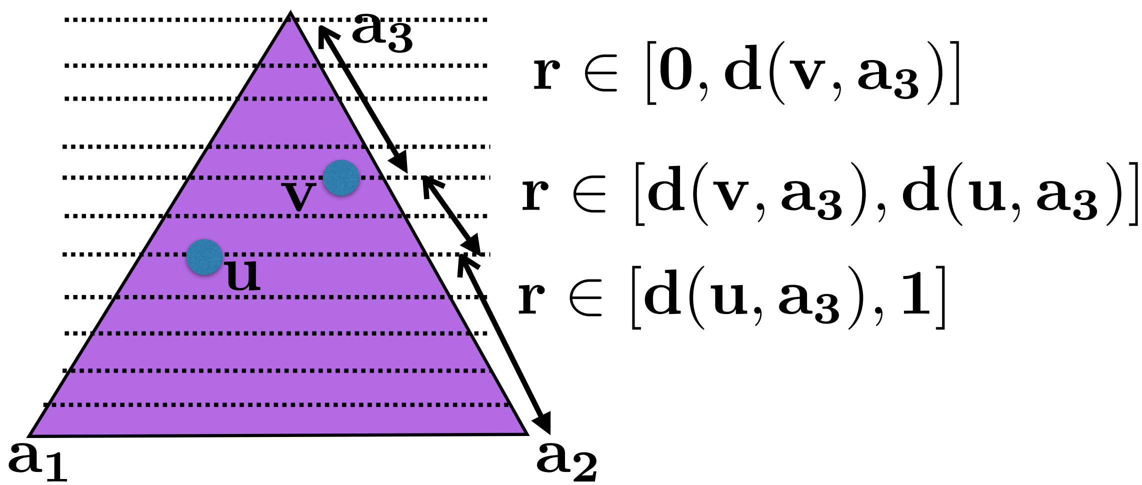
$$\mathbf{d(u,v) = \tfrac{1}{2}|u-v|_1}$$

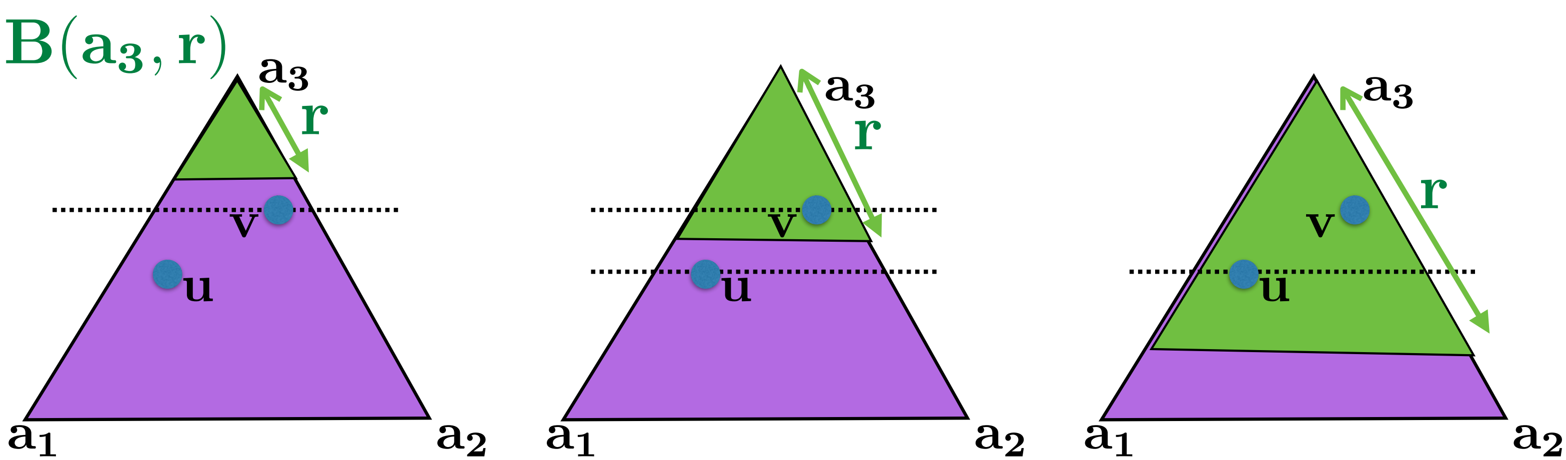
$$\mathbf{d(a_i,a_j) = 1}$$



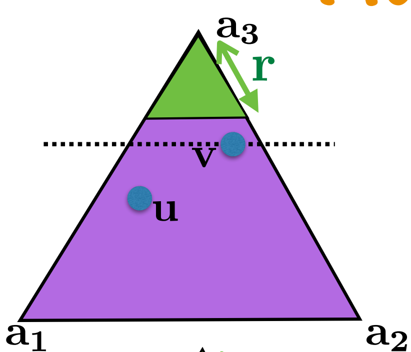
$$\mathbf{B(a_1,r) = \{u : d(a_1,u) \le r\}}$$

Pick random r, assign B(a3,r) to a3

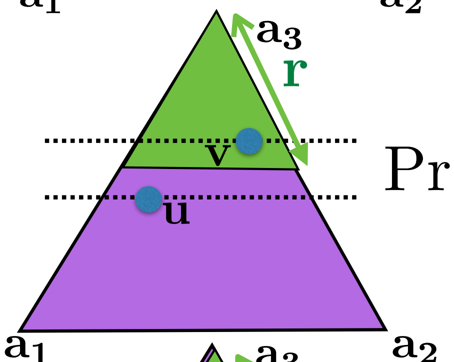




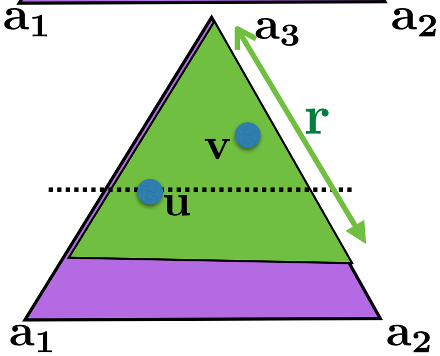
Pick random r, assign B(a3,r) to a3



$$\Pr[\mathbf{u},\mathbf{v} \notin \mathbf{B}(\mathbf{a_3},\mathbf{r})] = \Pr[\mathbf{r} < \mathbf{d}(\mathbf{a_3},\mathbf{v})] = \mathbf{d}(\mathbf{a_3},\mathbf{v})$$



$$\Pr[\mathbf{u},\mathbf{v}\ \textbf{separated by}\ \mathbf{B}(\mathbf{a_3},\mathbf{r})] = \mathbf{d}(\mathbf{a_3},\mathbf{u}) - \mathbf{d}(\mathbf{a_3},\mathbf{v})$$
$$= |\mathbf{u_3} - \mathbf{v_3}|$$



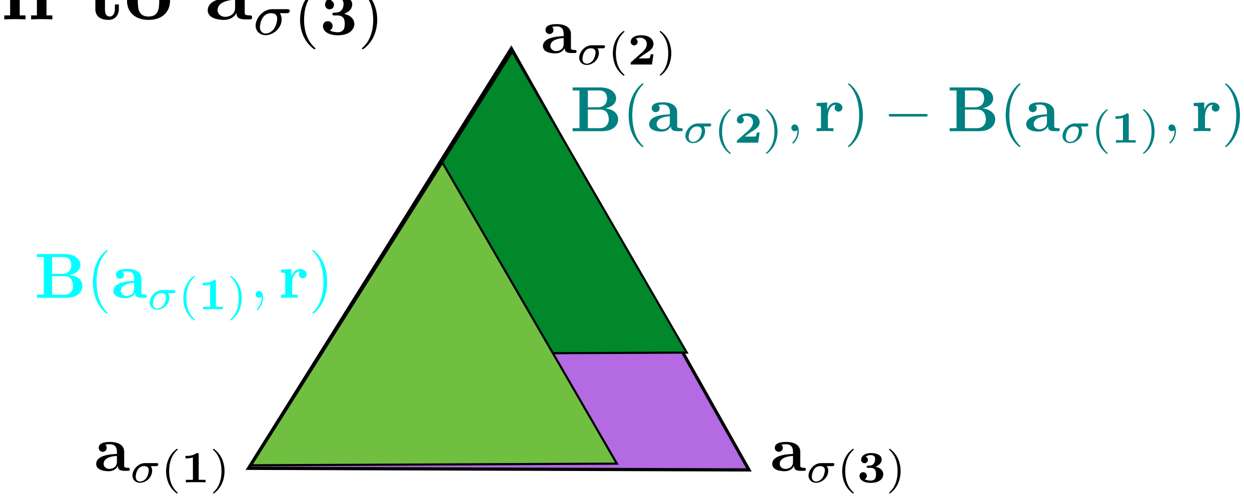
$$\Pr[\mathbf{u},\mathbf{v} \in \mathbf{B}(\mathbf{a_3},\mathbf{r})] = \mathbf{1} - \mathbf{d}(\mathbf{a_3},\mathbf{u})$$

Consider terminals in random order

$$\mathbf{a}_{\sigma(\mathbf{1})}, \mathbf{a}_{\sigma(\mathbf{2})}, \mathbf{a}_{\sigma(\mathbf{3})}$$

Assigning u

$$\mathbf{if\ d(a_{\sigma(1)}, u) < r\ then\ assign\ to\ a_{\sigma(1)}}$$
$$\mathbf{else\ if\ d(a_{\sigma(2)}, u) < r\ then\ assign\ to\ a_{\sigma(2)}}$$
$$\mathbf{else\ assign\ to\ a_{\sigma(3)}}$$



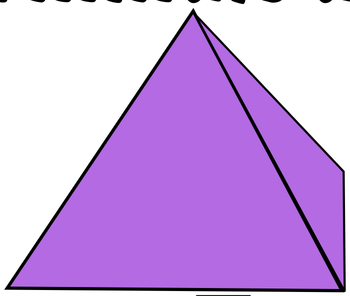
Full Algorithm for 3-way cut

Solve relaxation: embed vertices in triangle
Pick random permutation of terminals
-assign to first terminal a all vertices in B(a,r)
where r is random uniform in [0,1]
-assign to second terminal b all unassigned vertices in B(b,r)
-assign to third terminal remaining vertices

Algorithm

$$\mathbf{d(u,v) = \frac{1}{2}|x_u - x_v|_1}$$

LP relaxation: embed vertices in k-simplex with terminals at corners



$$\mathbf{x_i \geq 0,\ \ \sum_i x_i = 1}$$

$$\min \sum_{\mathbf{uv \in E}} \mathbf{c_{uv} d(u,v)} :$$

$$\mathbf{u \in \text{k-simplex} \ \ \forall u}$$

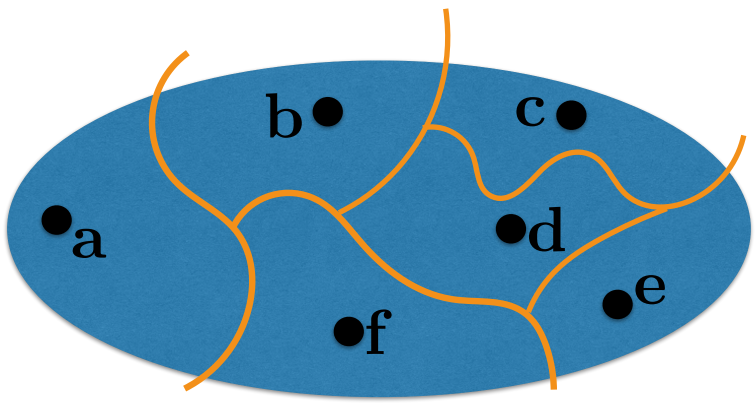
random ordering $\mathbf{a_{\sigma(1)}, a_{\sigma(2)}, \cdots, a_{\sigma(k)}}$ $\quad \mathbf{r \in [0,1]}$

$\mathbf{for\ \ i = 1, \ldots, k-1 :}$

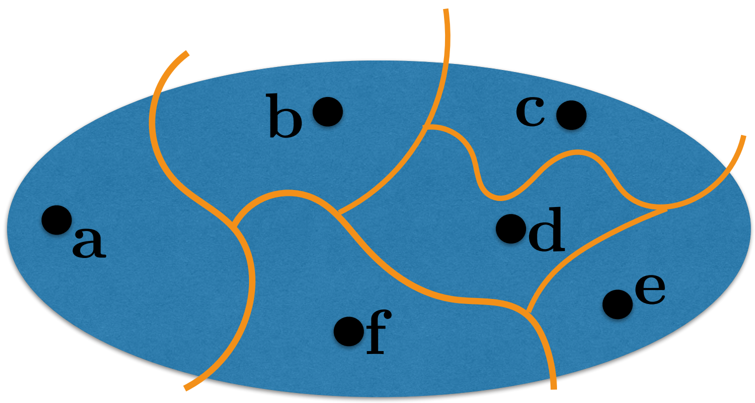
$\quad \mathbf{assign\ \ to\ \ a_{\sigma(i)}\ \ unassigned\ \ vertices\ \ of\ \ B(a_{\sigma(i)}, r)}$

$\mathbf{assign\ \ rest\ \ to\ \ a_{\sigma(k)}}$

Multiway cut, linear programming and randomized rounding



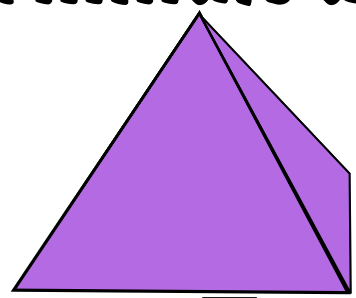
Multiway cut, linear programming and randomized rounding



Algorithm

$\mathbf{d(u,v) = \frac{1}{2}|x_u - x_v|_1}$

LP relaxation: embed vertices in k-simplex with terminals at corners



$$\min \sum_{\mathbf{uv \in E}} \mathbf{c_{uv} d(u,v)} :$$

$$\mathbf{u \in \text{k-simplex} \quad \forall u}$$

$\mathbf{x_i \geq 0, \ \sum_i x_i = 1}$

random ordering $\mathbf{a_{\sigma(1)}, a_{\sigma(2)}, \cdots, a_{\sigma(k)}}$ $\quad \mathbf{r \in [0,1]}$

$\mathbf{\text{for } i = 1, \ldots, k-1 :}$

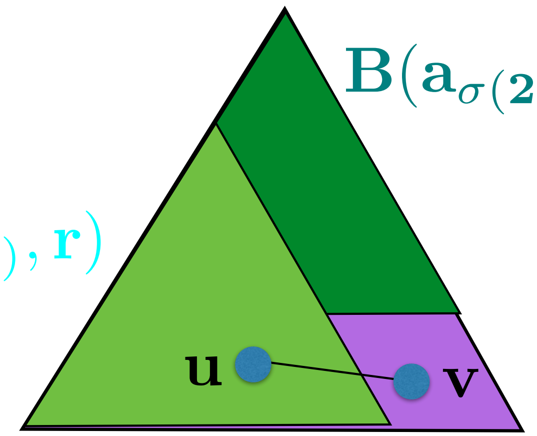
$\quad \mathbf{\text{assign to } a_{\sigma(i)} \text{ unassigned vertices of } B(a_{\sigma(i)}, r)}$

$\mathbf{\text{assign rest to } a_{\sigma(k)}}$

Analysis

$$\mathbf{E}[\mathbf{Output}] = \sum_{\mathbf{uv} \in \mathbf{E}} \mathbf{c_{uv}} \Pr(\mathbf{uv} \in \ \mathbf{Output})$$

$$\mathbf{a} = \mathbf{a}_\sigma(\mathbf{1}), \mathbf{b} = \mathbf{a}_{\sigma(\mathbf{2})}, \mathbf{c} = \mathbf{a}_{\sigma(\mathbf{3})}$$

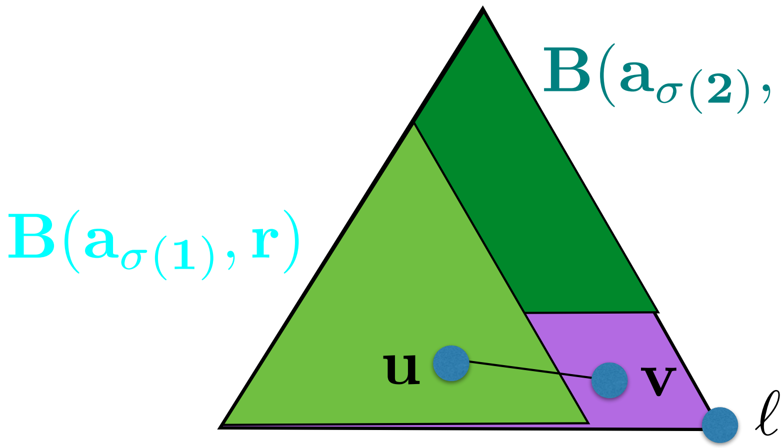


$\mathbf{uv} \in$ Output iff:

$$\min(\mathbf{d}(\mathbf{a},\mathbf{u}), \mathbf{d}(\mathbf{a},\mathbf{v})) < \mathbf{r} < \max(\mathbf{d}(\mathbf{a},\mathbf{u}), \mathbf{d}(\mathbf{a},\mathbf{v})) \quad \textbf{or}$$

$$[\mathbf{r} < \min(\mathbf{d}(\mathbf{a},\mathbf{u}), \mathbf{d}(\mathbf{a},\mathbf{v})) \ \textbf{and}$$

$$\min(\mathbf{d}(\mathbf{b},\mathbf{u}), \mathbf{d}(\mathbf{b},\mathbf{v})) < \mathbf{r} < \max(\mathbf{d}(\mathbf{b},\mathbf{u}), \mathbf{d}(\mathbf{b},\mathbf{v}))]$$



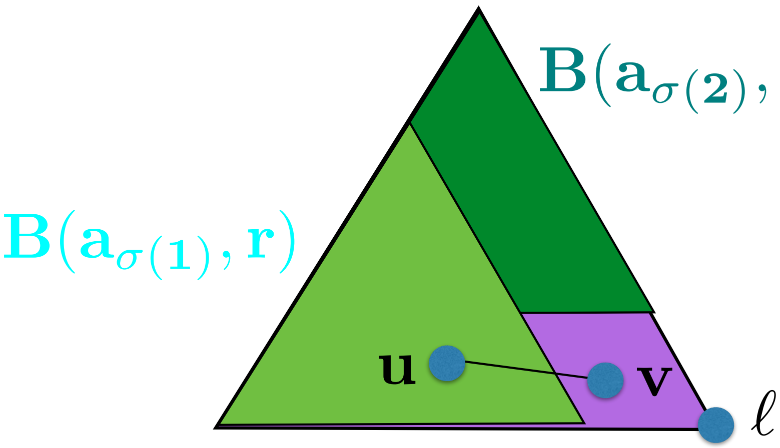
$\mathbf{uv} \in \mathbf{Output:}$

$\mathbf{a_i}$ **first to catch u or v and catches exactly one**

$$\ell = \mathbf{arg}\min\{\min(\mathbf{d}(\mathbf{a_i}, \mathbf{u}), \mathbf{d}(\mathbf{a_i}, \mathbf{v}))\}$$

$\mathbf{i} \neq \ell :$ **i precedes** ℓ **and** $\mathbf{B(a_i, r)}$ **separates them**

$$\frac{\mathbf{1}}{\mathbf{2}}|\mathbf{d}(\mathbf{a_i}, \mathbf{u}) - \mathbf{d}(\mathbf{a_i}, \mathbf{v})|$$

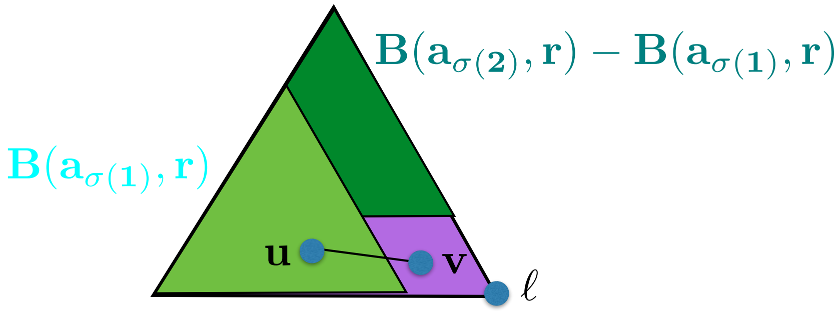


$\mathbf{uv \in}$ **Output:**
$\mathbf{a_i}$ **first to catch u or v**
and catches exactly one

$$\ell = \mathbf{arg}\min\{\min(\mathbf{d}(\mathbf{a_i}, \mathbf{u}), \mathbf{d}(\mathbf{a_i}, \mathbf{v}))\}$$

$\mathbf{i} = \ell :$ ℓ **not last and** $\mathbf{B(a_\ell, r)}$ **separates them**

$$\left(\mathbf{1} - \frac{\mathbf{1}}{\mathbf{k}}\right)\left|\mathbf{d}(\mathbf{a}_\ell, \mathbf{u}) - \mathbf{d}(\mathbf{a}_\ell, \mathbf{v})\right|$$

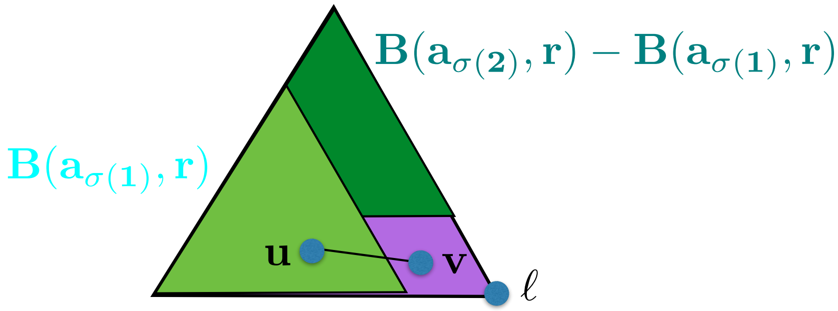


Together

$$(\mathbf{1} - \tfrac{\mathbf{1}}{\mathbf{k}})|\mathbf{d}(\mathbf{a}_\ell, \mathbf{u}) - \mathbf{d}(\mathbf{a}_\ell, \mathbf{v})| + \sum_{\mathbf{i}\neq\ell} \tfrac{\mathbf{1}}{\mathbf{2}}|\mathbf{d}(\mathbf{a_i}, \mathbf{u}) - \mathbf{d}(\mathbf{a_i}, \mathbf{v})|$$

$$= \tfrac{\mathbf{1}}{\mathbf{2}}(\mathbf{1} - \tfrac{\mathbf{2}}{\mathbf{k}})|\mathbf{d}(\mathbf{a}_\ell, \mathbf{u}) - \mathbf{d}(\mathbf{a}_\ell, \mathbf{v})| + \sum_{\mathbf{i}} \tfrac{\mathbf{1}}{\mathbf{2}}|\mathbf{d}(\mathbf{a_i}, \mathbf{u}) - \mathbf{d}(\mathbf{a_i}, \mathbf{v})|$$

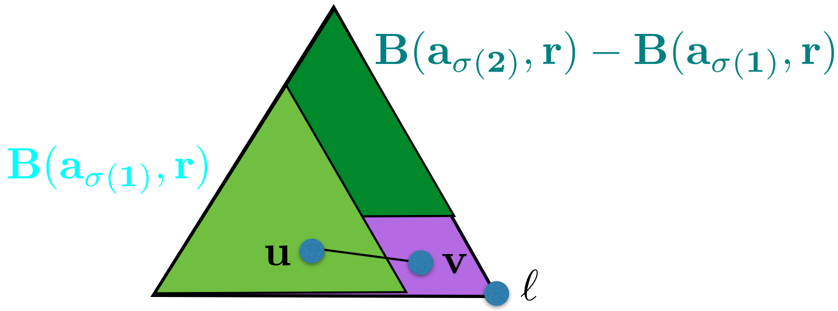
$$= \tfrac{\mathbf{1}}{\mathbf{2}}(\mathbf{1} - \tfrac{\mathbf{2}}{\mathbf{k}})|\mathbf{u}_\ell - \mathbf{v}_\ell| + \sum_{\mathbf{i}} \tfrac{\mathbf{1}}{\mathbf{2}}|\mathbf{u_i} - \mathbf{v_i}|$$



$$\sum_{\mathbf{u}} \mathbf{u_i} = \sum_{\mathbf{i}} \mathbf{v_i} = \mathbf{1}$$

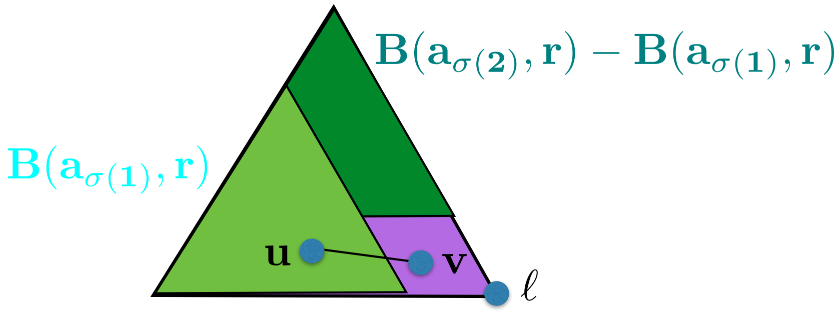
so

$$|\mathbf{u}_\ell - \mathbf{v}_\ell| \leq \frac{\mathbf{1}}{\mathbf{2}} \sum_{\mathbf{i}} |\mathbf{u_i} - \mathbf{v_i}|$$



Together

$$\left(\frac{1}{2} - \frac{1}{k}\right)|u_\ell - v_\ell| + \sum_i \frac{1}{2}|u_i - v_i|$$
$$\leq \sum_i \left(\frac{3}{4} - \frac{1}{2k}\right)|u_i - v_i|$$
$$= \left(\frac{3}{2} - \frac{1}{k}\right) \cdot \frac{1}{2}|u - v|_1$$

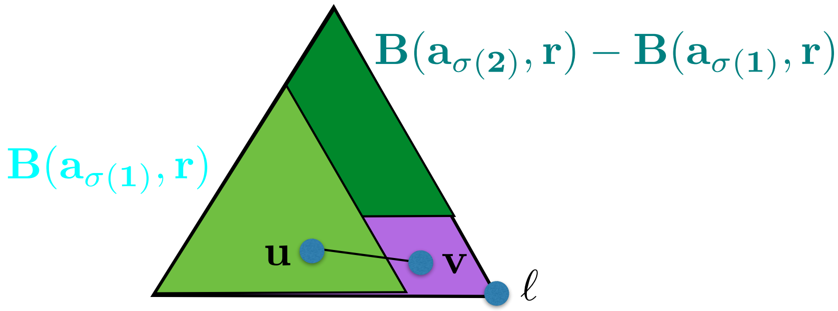


Sum

$$\mathbf{E}[\mathbf{Output}] = \sum_{\mathbf{uv} \in \mathbf{E}} \mathbf{c_{uv}} \Pr(\mathbf{uv} \in \mathbf{Output})$$

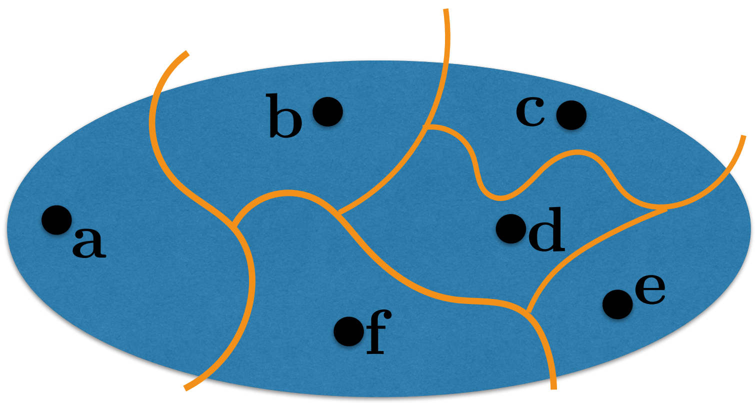
$$\leq \sum_{\mathbf{uv} \in \mathbf{E}} \mathbf{c_{uv}} \left(\frac{\mathbf{3}}{\mathbf{2}} - \frac{\mathbf{1}}{\mathbf{k}}\right) \cdot \frac{\mathbf{1}}{\mathbf{2}} |\mathbf{u} - \mathbf{v}|_{\mathbf{1}}$$

$$\leq \left(\frac{\mathbf{3}}{\mathbf{2}} - \frac{\mathbf{1}}{\mathbf{k}}\right) \cdot \mathbf{OPT}$$

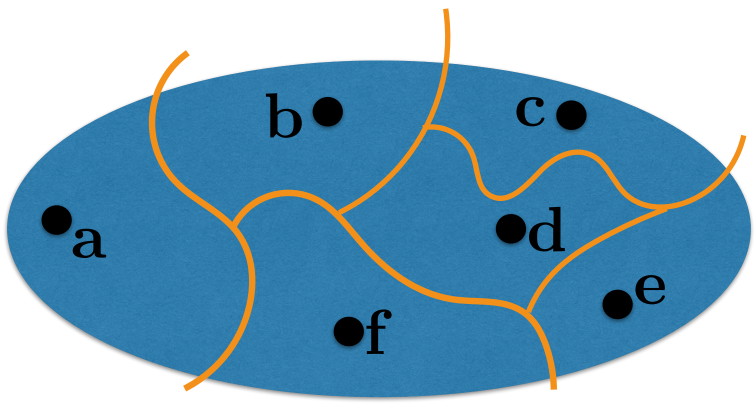


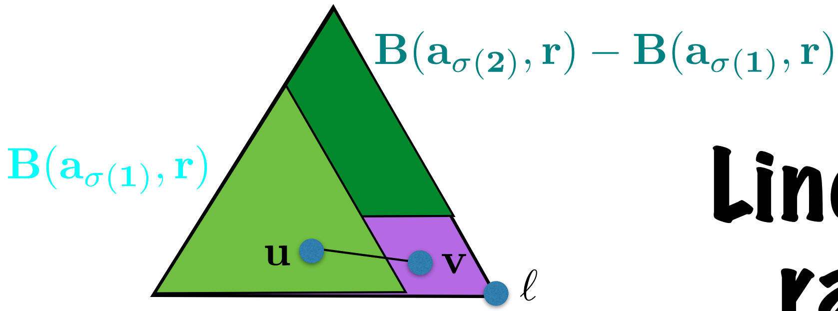
Linear programming and randomized rounding give a 3/2 - 1/k approximation for multicut

Multiway cut, linear programming and randomized rounding



Multiway cut, linear programming and randomized rounding





Linear programming and randomized rounding give a 3/2 - 1/k approximation for multicut

Can we do better?

k=3: 12/11

by using LP to find the rounding!

APX-hard: cannot get $1+\epsilon$

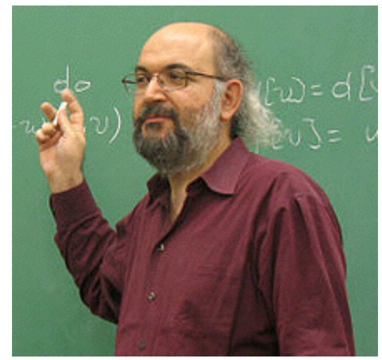
The story behind the story

Applications:

- “minimization of communication costs in parallel computing systems...
- assigning program modules to processors ...
- partitioning files among the nodes of a network...
- assigning users to base computers in a multicomputer environment...
- partitioning the elements of a circuit into the subcircuits that will go on different chips”

Elias Dalhaus

David Johnson



Mihalis Yannakakis

Christos Papadimitriou

Paul Seymour

APX hardness
approx with min cuts

Gruia Calinescu

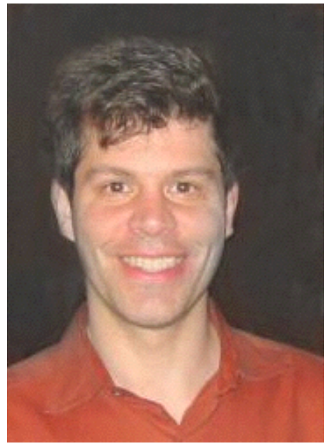
Howard Karloff

Yuval Rabani

Geometric embedding

3/2 - 1/k

David Karger



Philip Klein

Mikkel Thorup

Cliff Stein

Neal Young

12/11
rounding by
linear programming

Techniques

- rounding input
- linear programming relaxation
- randomized rounding
- probabilistic analysis techniques
- geometric interpretation

Problems

- Vertex cover
- Knapsack
- Bin packing
- Set cover
- Multiway cut

Approximation algorithms, Part II

LP duality

primal dual algorithms

semi-definite programming

Multiway cut, linear programming and randomized rounding