Advanced Algorithms

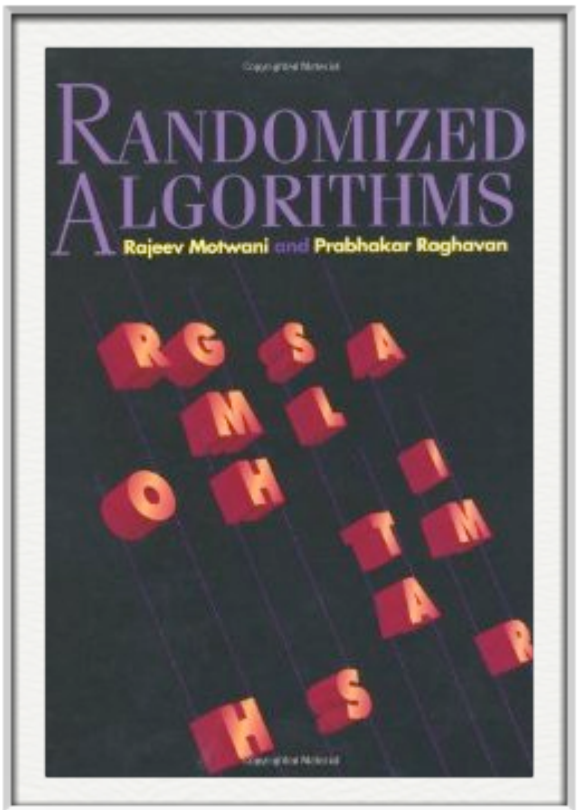
南京大学

尹一通

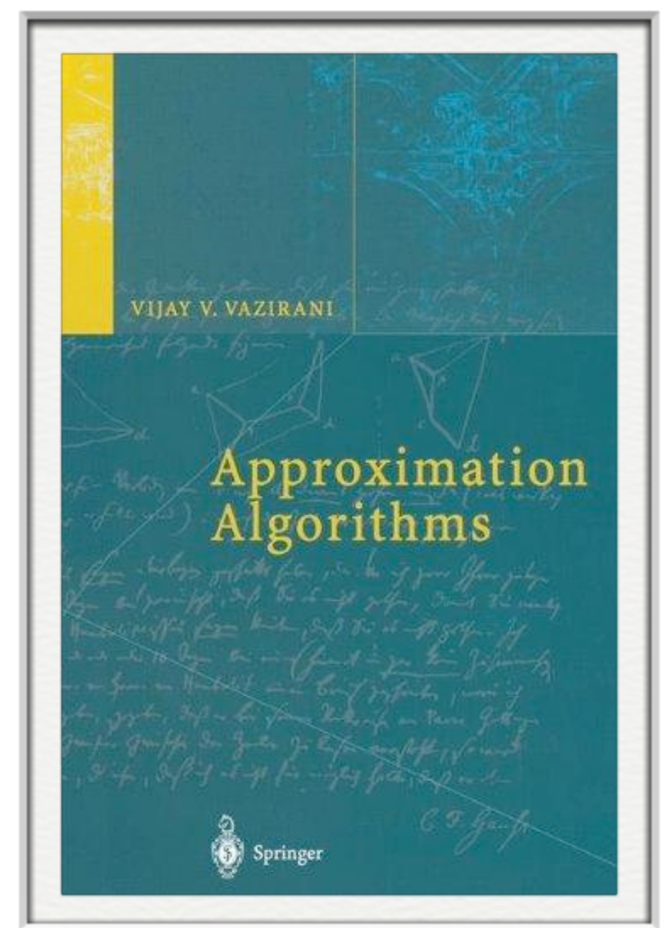
Course Info

- Instructor: 尹一通
    - yinyt@nju.edu.cn
- Office hour:
    - Wednesday, 4–6pm, 计算机系 804
- course homepage:
    - http://tcs.nju.edu.cn/wiki/

Textbooks

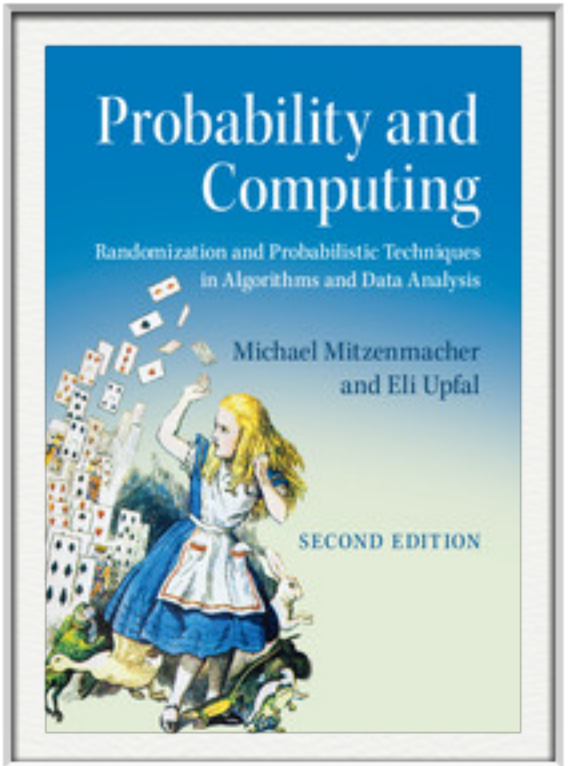


Rajeev Motwani and Prabhakar Raghavan.
Randomized Algorithms.
Cambridge University Press, 1995.



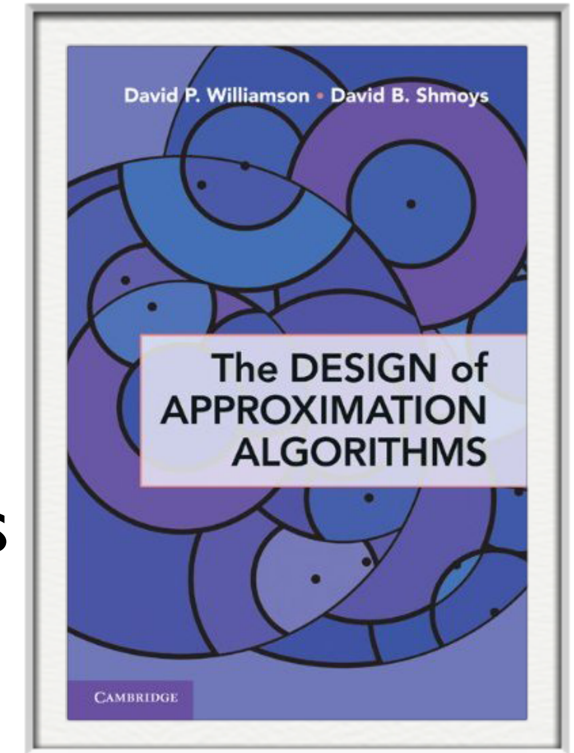
Vijay Vazirani
Approximation Algorithms.
Spinger-Verlag, 2001.

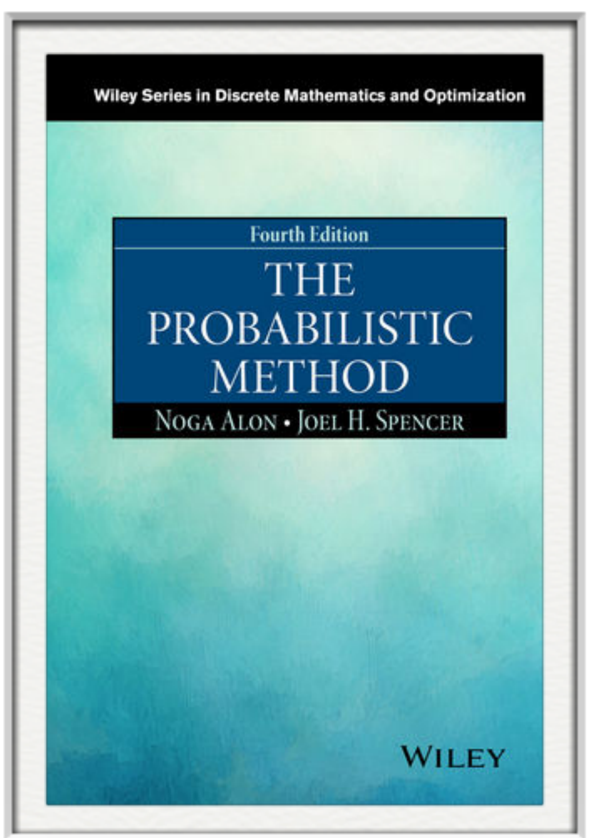
References



Mitzenmacher and Upfal.
Probability and Computing,
2nd Ed.

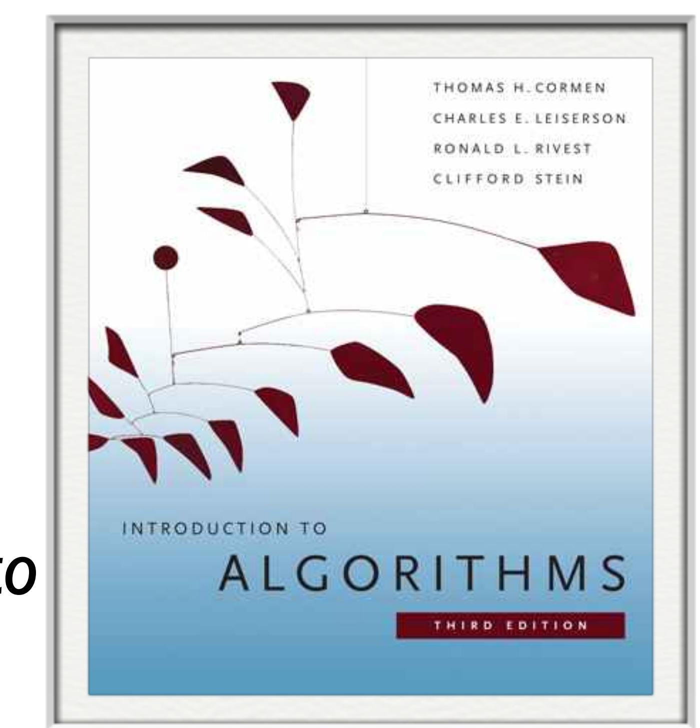
Williamson and Shmoys
The Design of
Approximation Algorithms





Alon and Spencer
The Probabilistic Method,
4th Ed.

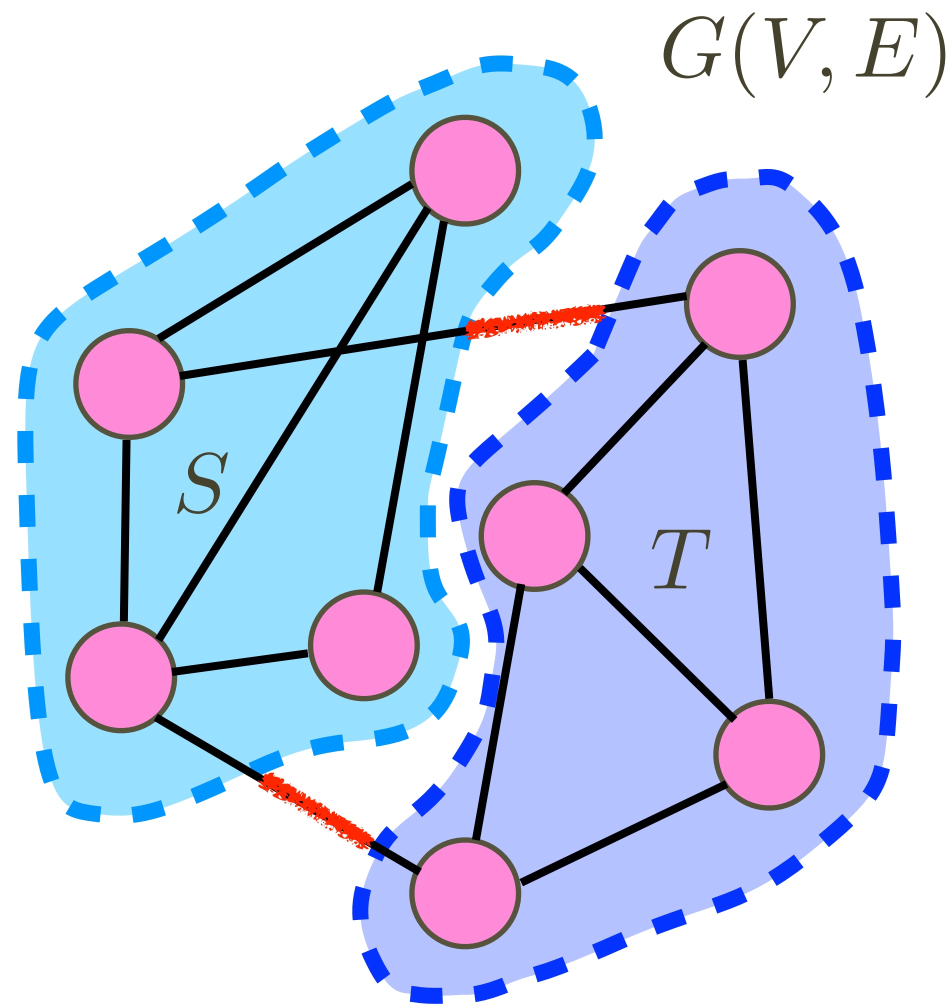
CLRS
Introduction to
Algorithms



“Advanced” Algorithms

Min-Cut

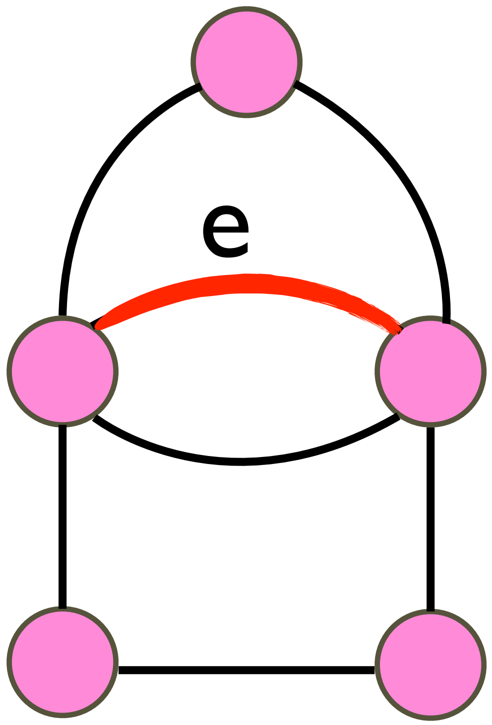
- Partition V into two parts: S and T
- Minimize the cut $E(S,T)$
- deterministic algorithm:
  - max-flow min-cut
  - best known upper bound: $O(mn + n^2 \log n)$



$$E(S,T) = \{uv \in E \mid u \in S, v \in T\}$$

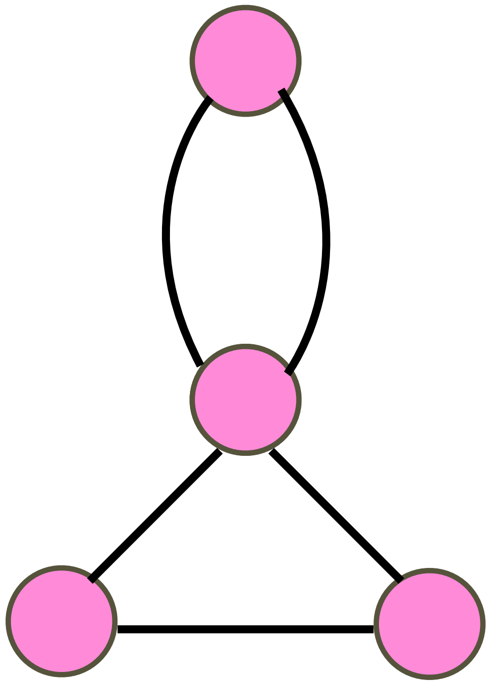
Contraction

- multigraph $G(V,E)$
- multigraph: allow parallel edges
- for an edge e, $\text{contract}(e)$ merges the two endpoints.



Contraction

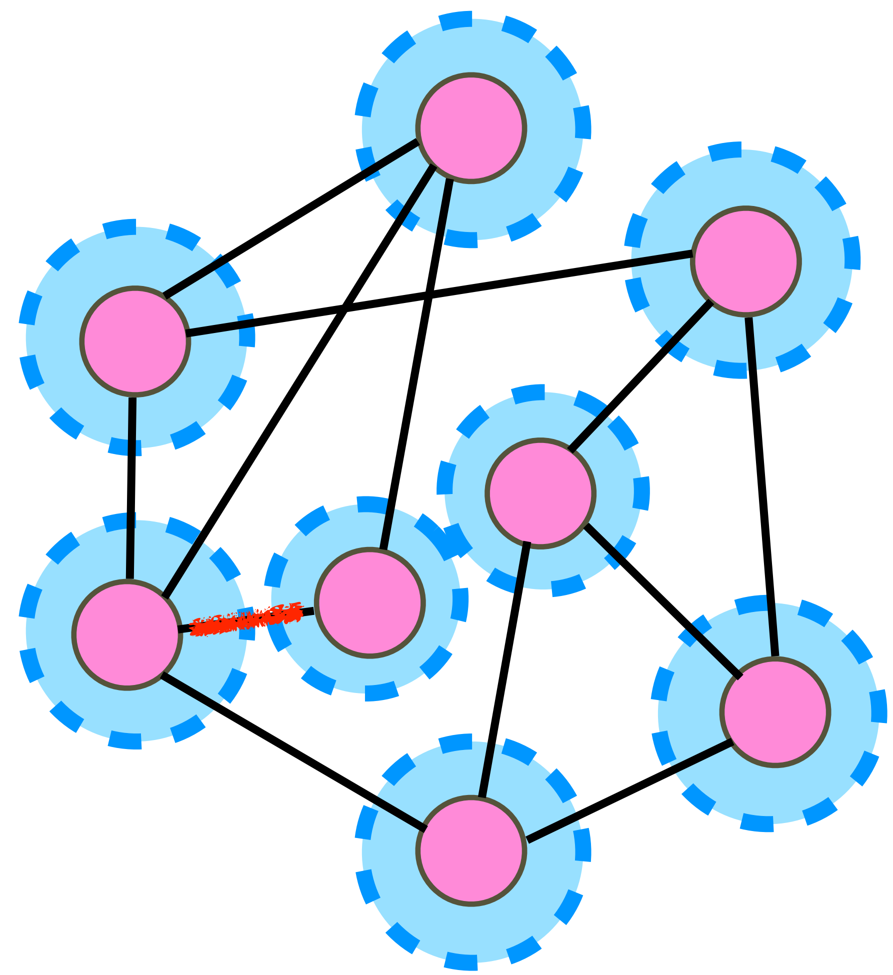
- multigraph $G(V,E)$
- multigraph: allow parallel edges
- for an edge e, $\text{contract}(e)$ merges the two endpoints.



Karger's min-cut Algorithm

MinCut (multigraph $G(V,E)$)

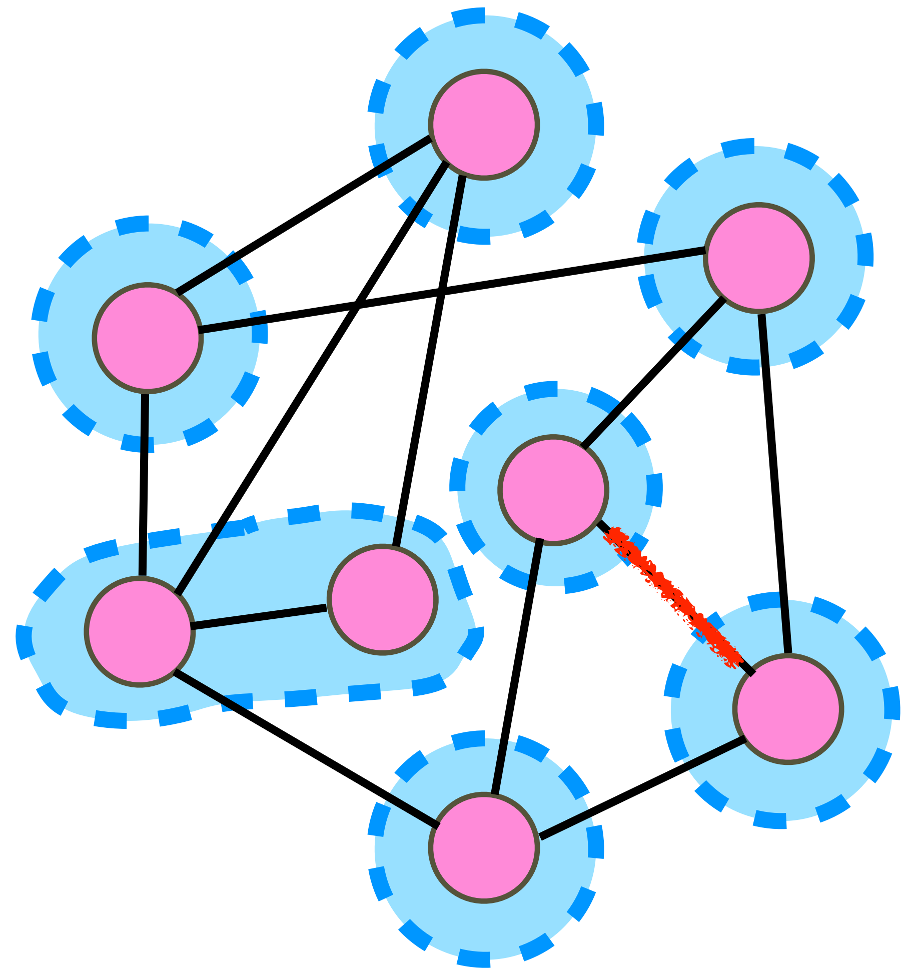
while $\|V\|>2$ do
choose a uniform $e \in E$;
contract(e);
return remaining edges;



Karger's min-cut Algorithm

MinCut (multigraph $G(V,E)$)

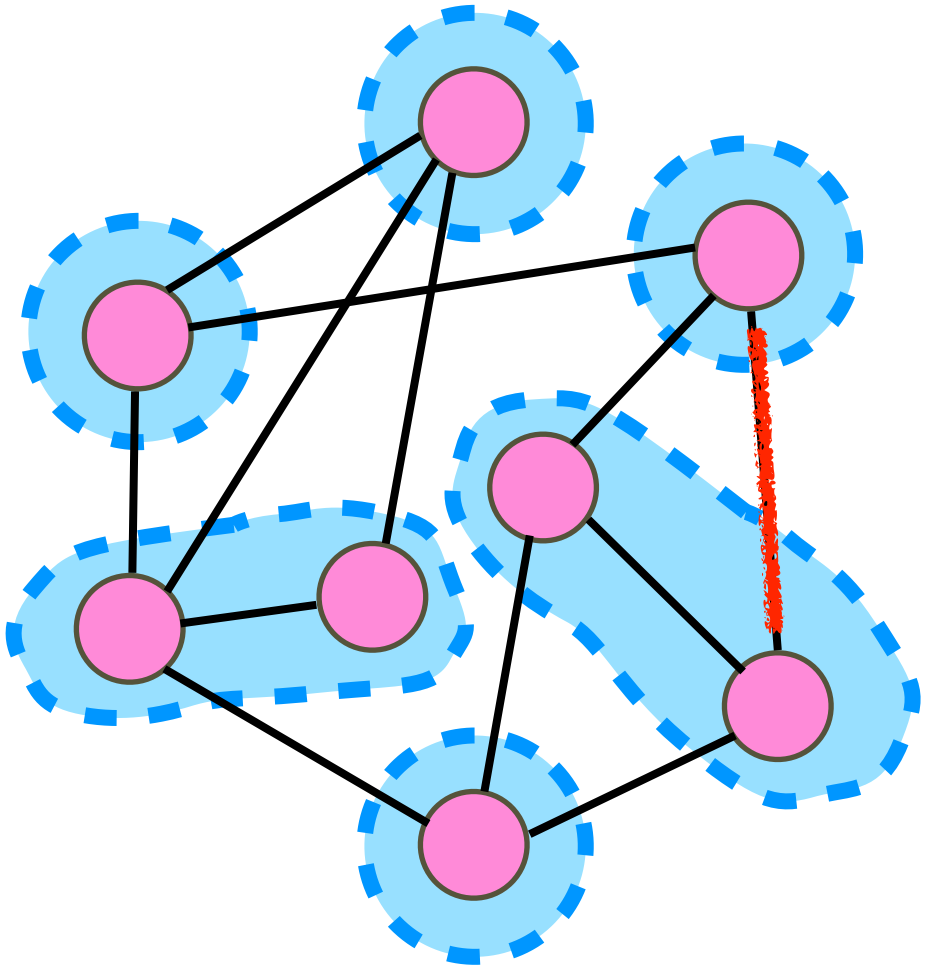
while $\|V\|>2$ do
  choose a uniform $e \in E$;
  contract(e);
return remaining edges;



Karger's min-cut Algorithm

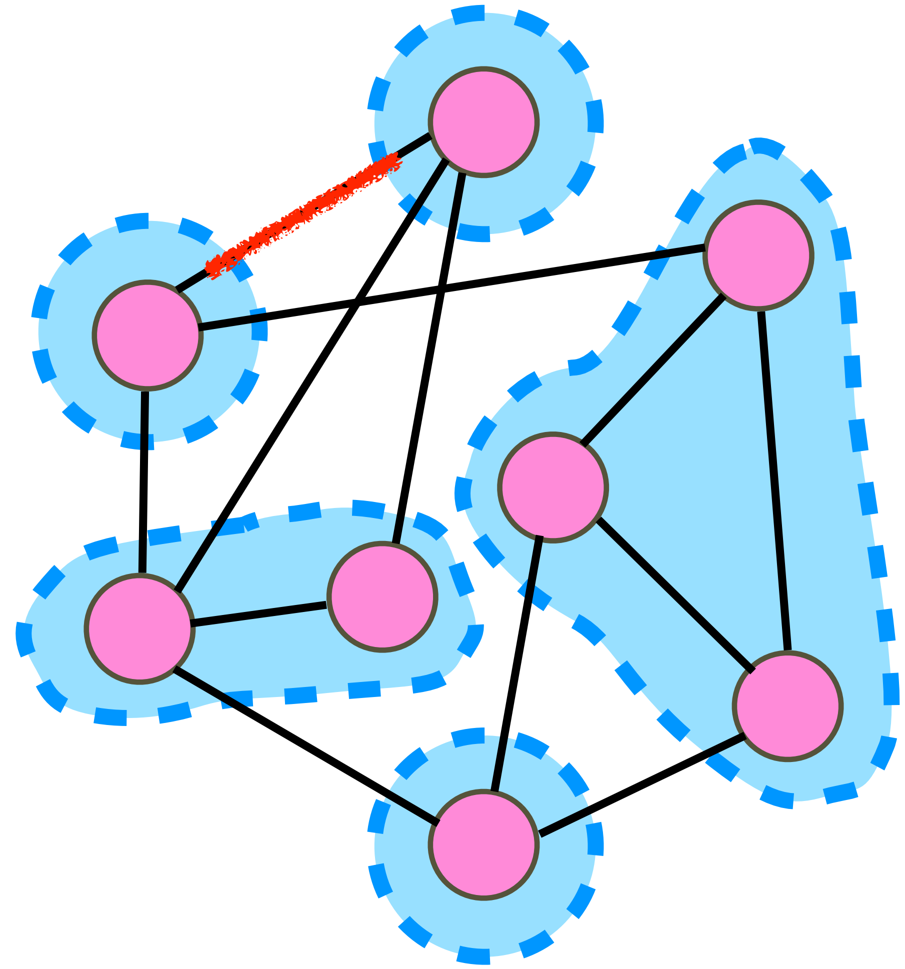
MinCut (multigraph $G(V,E)$)

while $\|V\|>2$ do
  choose a uniform $e \in E$;
  contract(e);
return remaining edges;



Karger's min-cut Algorithm

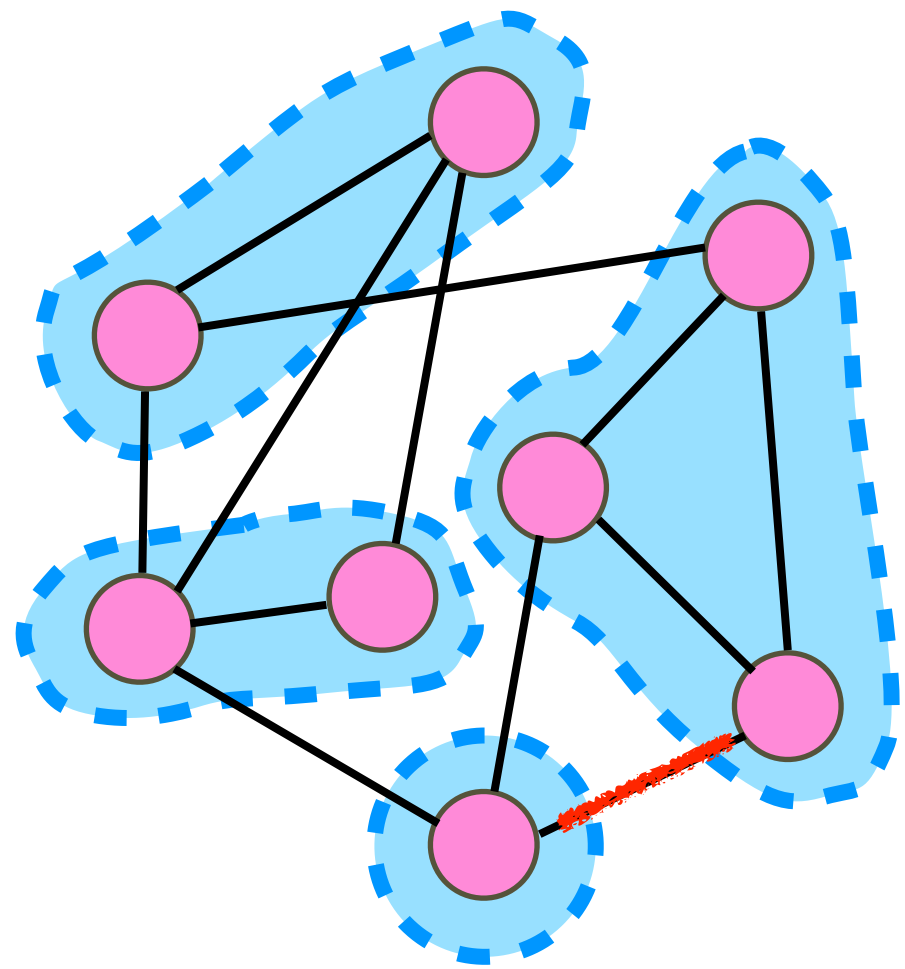
MinCut (multigraph $G(V,E)$)
while $\|V\|>2$ do choose a uniform $e \in E$; contract(e); return remaining edges;



Karger's min-cut Algorithm

MinCut (multigraph $G(V,E)$)

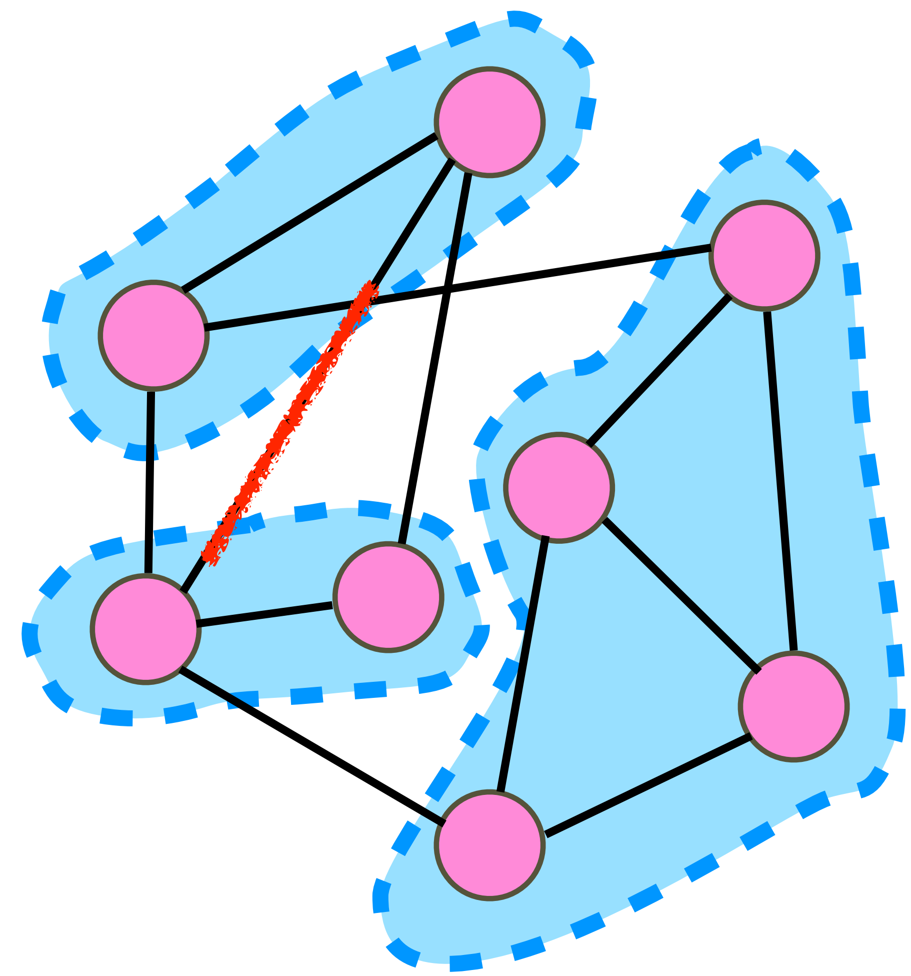
while $\|V\|>2$ do
choose a uniform $e \in E$;
contract(e);
return remaining edges;



Karger's min-cut Algorithm

MinCut (multigraph $G(V,E)$)

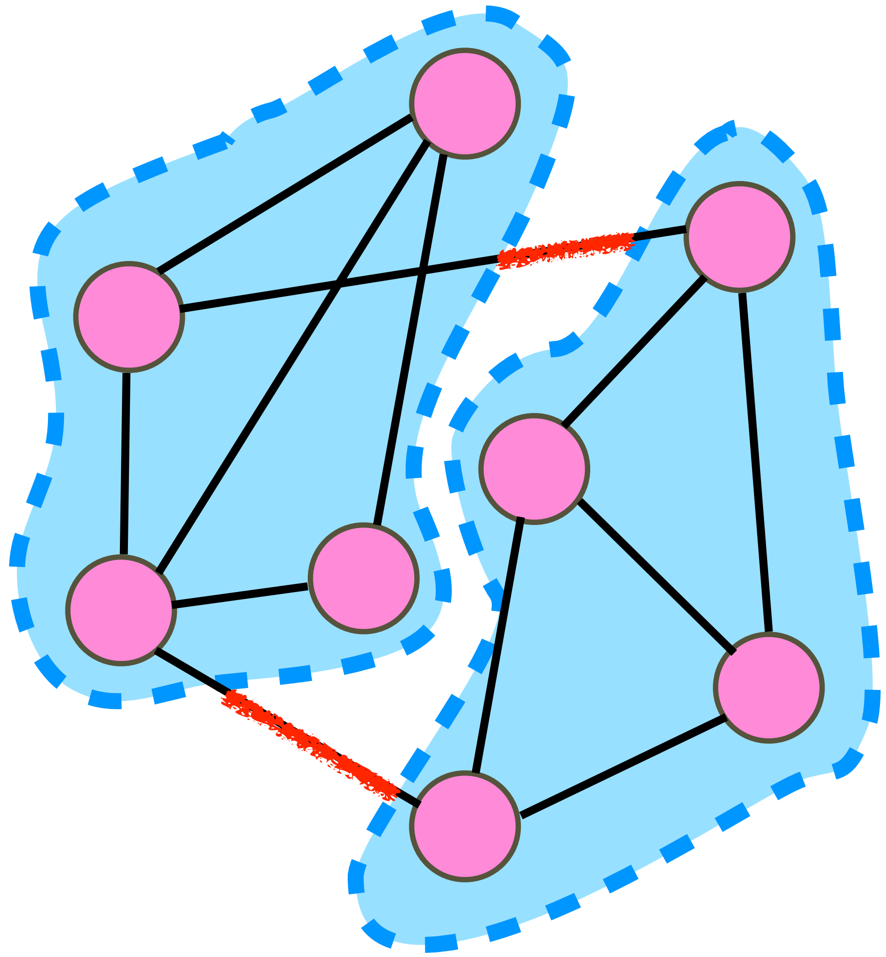
while $\|V\|>2$ do
  choose a uniform $e \in E$;
  contract(e);
return remaining edges;



Karger's min-cut Algorithm

MinCut (multigraph $G(V,E)$)

while $\|V\|>2$ do
  choose a uniform $e \in E$;
  contract(e);
return remaining edges;



edges returned

MinCut (multigraph $G(V,E)$)

while $\|V\|>2$ do
  choose a uniform $e \in E$;
  contract(e);
return remaining edges;

Theorem (Karger 1993)**:**

$$\Pr[\text{ a min-cut is returned }] \geq \frac{2}{n(n-1)}$$

repeat independently
for $n(n\text{-}1)/2$ times
and return the smallest cut

$$\Pr[\text{ fail to finally return a min-cut }]$$

$$= \Pr[\text{ fail to construct a min-cut in one trial }]^{n(n-1)/2}$$

$$\leq \left(1 - \frac{2}{n(n-1)}\right)^{n(n-1)/2} < \frac{1}{\mathrm{e}}$$

MinCut (multigraph $G(V,E)$)

while $|V|>2$ do
  choose a uniform $e \in E$;
  contract(e);
return remaining edges;

suppose $e_1, e_2, \ldots, e_{n-2}$ are contracted edges

initially: $G_1 = G$

i-th round:

$$G_i = \text{contract}(G_{i-1}, e_{i-1})$$

$$\left.\begin{array}{c} C \text{ is a min-cut in } G_{i-1} \\ e_{i-1} \notin C \end{array}\right\} \Rightarrow C \text{ is a min-cut in } G_i$$

C: a min-cut of G

$$\Pr[C \text{ is returned}] \geq \Pr[e_1, e_2, \ldots, e_{n-2} \notin C]$$

chain rule:
$$= \prod_{i=1}^{n-2} \Pr[e_i \notin C \mid e_1, e_2, \ldots, e_{i-1} \notin C]$$

suppose $e_1, e_2, \ldots, e_{n-2}$ are contracted edges

initially: $G_1 = G$ *i*-th round: $G_i = \mathrm{contract}(G_{i-1}, e_{i-1})$

$$\left.\begin{array}{c} C \text{ is a min-cut in } G_{i-1} \\ e_{i-1} \notin C \end{array}\right\} \Rightarrow C \text{ is a min-cut in } G_i$$

C: a min-cut of G

$$\Pr[C \text{ is returned}] \geq \prod_{i=1}^{n-2} \Pr[e_i \notin C \mid e_1, e_2, \ldots, e_{i-1} \notin C]$$

$\Downarrow$ C is a min-cut in G_i

$$C \text{ is a min-cut in } G(V,E) \Rightarrow |E| \geq \frac{1}{2}|C||V|$$

$$\geq \prod_{i=1}^{n-2}\left(1 - \frac{2}{(n-i+1)}\right)$$

Proof:

min-degree of $G \geq |C|$

$$= \prod_{i=1}^{n-2} \frac{n-i-1}{n-i+1} = \frac{2}{n(n-1)}$$

MinCut (multigraph $G(V,E)$)

while $\|V\|>2$ do
  choose a uniform $e \in E$;
  contract(e);
return remaining edges;

Theorem (Karger 1993)**:**

For any min-cut C,

$$\Pr[C \text{ is returned}] \geq \frac{2}{n(n-1)}$$

running time: $O(n^2)$

repeat *independently* for $O(n^2 \log n)$ times

returns a min-cut with probability $1\text{-}O(1/n)$

total running time: $O(n^4 \log n)$

Number of Min-Cuts

Theorem (Karger 1993)**:**

For any min-cut C,

$$\Pr[C \text{ is returned}] \geq \frac{2}{n(n-1)}$$

Corollary

The number of distinct min-cuts in a graph of n vertices is at most $n(n\text{-}1)/2$.

An Observation

MinCut (multigraph $G(V,E)$)

while $\|V\|>t$ do
  choose a uniform $e \in E$;
  contract(e);
return remaining edges;

C: a min-cut of G

$$\Pr[e_1, \ldots, e_{n-t} \notin C] = \prod_{i=1}^{n-t} \Pr[e_i \notin C \mid e_1, \ldots, e_{i-1} \notin C]$$

$$\geq \prod_{i=1}^{n-t} \frac{n-i-1}{n-i+1} = \frac{t(t-1)}{n(n-1)}$$

only getting bad when t is small

Fast Min-Cut

Contract (G, t)

while $\lvert V\rvert > t$ do
  choose a uniform $e \in E$;
  contract(e);
return remaining edges;

FastCut (G)

if $\lvert V\rvert \leq 6$ then return a min-cut by brute force;
else: (t to be fixed later)
  G_1= Contract(G,t);
  G_2= Contract(G,t); (independently)
   return min{FastCut(G_1), FastCut(G_2)};

FastCut (G)
if $\|V\| \le 6$ then return a min-cut by brute force;
else: (t to be fixed later)
G_1= Contract(G,t); (independently)
G_2= Contract(G,t);
return min{FastCut(G_1), FastCut(G_2)};

C: a min-cut in G

A: no edge in C is contracted during Contract(G,t)

$$\Pr[A] = \prod_{i=1}^{n-t} \Pr[e_i \not\in C \mid e_1, \ldots, e_{i-1} \not\in C]$$

$$\ge \prod_{i=1}^{n-t} \frac{n-i-1}{n-i+1} \ge \frac{t(t-1)}{n(n-1)} \ge \left(\frac{t-1}{n-1}\right)^2$$

FastCut (G)

if $|V|\leq 6$ then return a min-cut by brute force;

else: (t to be fixed later)

G_1= Contract(G,t);

G_2= Contract(G,t);

(independently)

return min{FastCut(G_1), FastCut(G_2)};

C: a min-cut in G

set $t = \left\lceil 1 + \frac{n}{\sqrt{2}} \right\rceil$

A: no edge in C is contracted during Contract(G,t)

$$\Pr[A] \geq \left(\frac{t-1}{n-1}\right)^2 \geq \frac{1}{2}$$

$$p(n) = \min_{G:|V|=n} \Pr[\,\mathrm{FastCut}(G) \text{ returns a mincut in } G\,]$$

succeeds

$$\geq 1 - (1 - \Pr[A]\Pr[\mathrm{FastCut}(G_1) \text{ succeeds} \mid A])^2$$

$$\geq 1 - \left(1 - \left(\frac{t-1}{n-1}\right)^2 p\left(t\right)\right)^2 \geq p\left(\left\lceil 1 + \frac{n}{\sqrt{2}} \right\rceil\right) - \frac{1}{4} p\left(\left\lceil 1 + \frac{n}{\sqrt{2}} \right\rceil\right)^2$$

FastCut (G)
if $\|V\|\leq 6$ then return a min-cut by brute force; else: set $t=\left\lceil 1+\frac{n}{\sqrt{2}}\right\rceil$ G_1= Contract(G,t); G_2= Contract(G,t); (independently) return min{FastCut(G_1), FastCut(G_2)};

$$p(n)=\min_{G:|V|=n}\Pr[\,\mathrm{FastCut}(G)\text{ returns a mincut in }G\,]$$

$$\geq p\left(\left\lceil 1+\frac{n}{\sqrt{2}}\right\rceil\right)-\frac{1}{4}p\left(\left\lceil 1+\frac{n}{\sqrt{2}}\right\rceil\right)^2$$

by induction: $p(n)=\Omega\left(\dfrac{1}{\log n}\right)$

running time: $T(n)=2T\left(\left\lceil 1+\frac{n}{\sqrt{2}}\right\rceil\right)+O(n^2)$

by induction: $T(n)=O(n^2\log n)$

FastCut (G)

if $|V| \le 6$ then return a min-cut by brute force;

else: set $t = \left\lceil 1 + \frac{n}{\sqrt{2}} \right\rceil$

G_1= Contract(G,t);

G_2= Contract(G,t);

(independently)

return min{FastCut(G_1), FastCut(G_2)};

Theorem (Karger-Stein 1996)**:**

FastCut runs in time $O(n^2 \log n)$ and returns a min-cut with probability $\Omega(1/\log n)$.

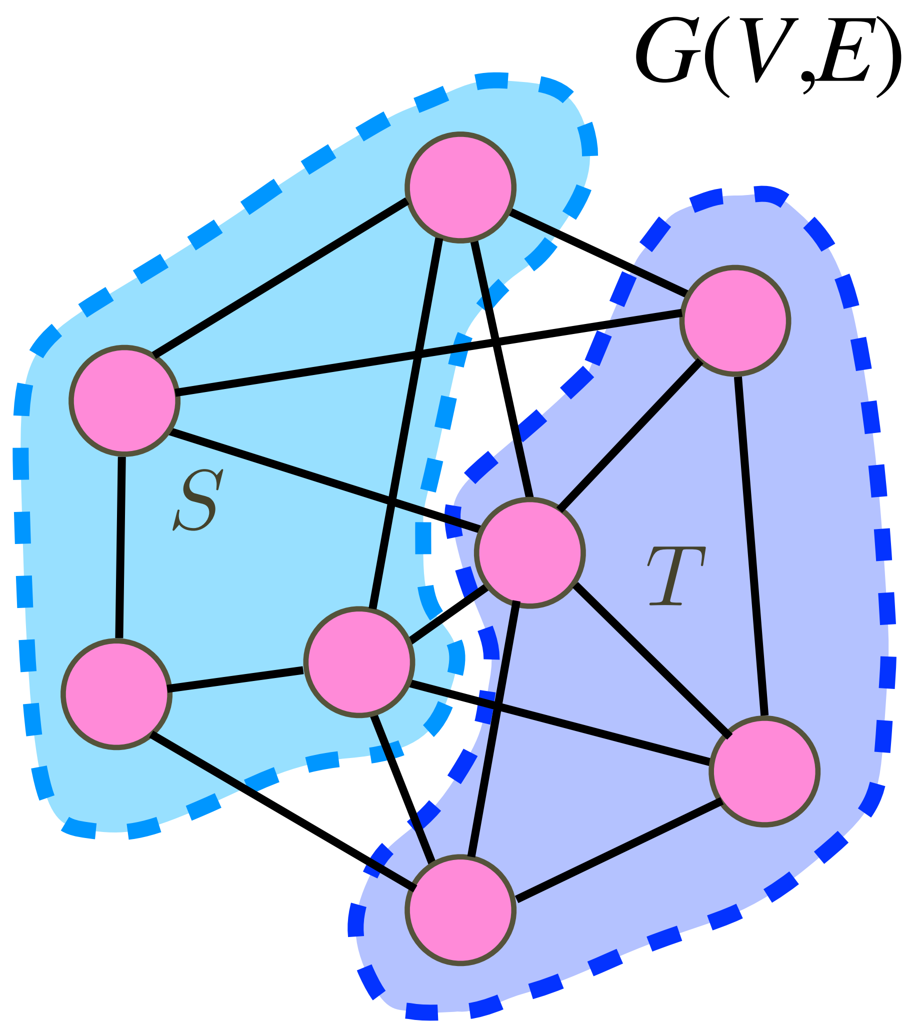
repeat *independently* for $O((\log n)^2)$ times

total running time: $O(n^2 \log^3 n)$

returns a min-cut with probability $1\text{-}O(1/n)$

Max-Cut

- Partition V into two parts: S and T
- Maximize the cut $E(S,T)$
- **NP-hard**
  - one of Karp's 21 NP-complete problems
- Approximation algorithms?



$$E(S,T) = \{uv \in E \mid u \in S, v \in T\}$$

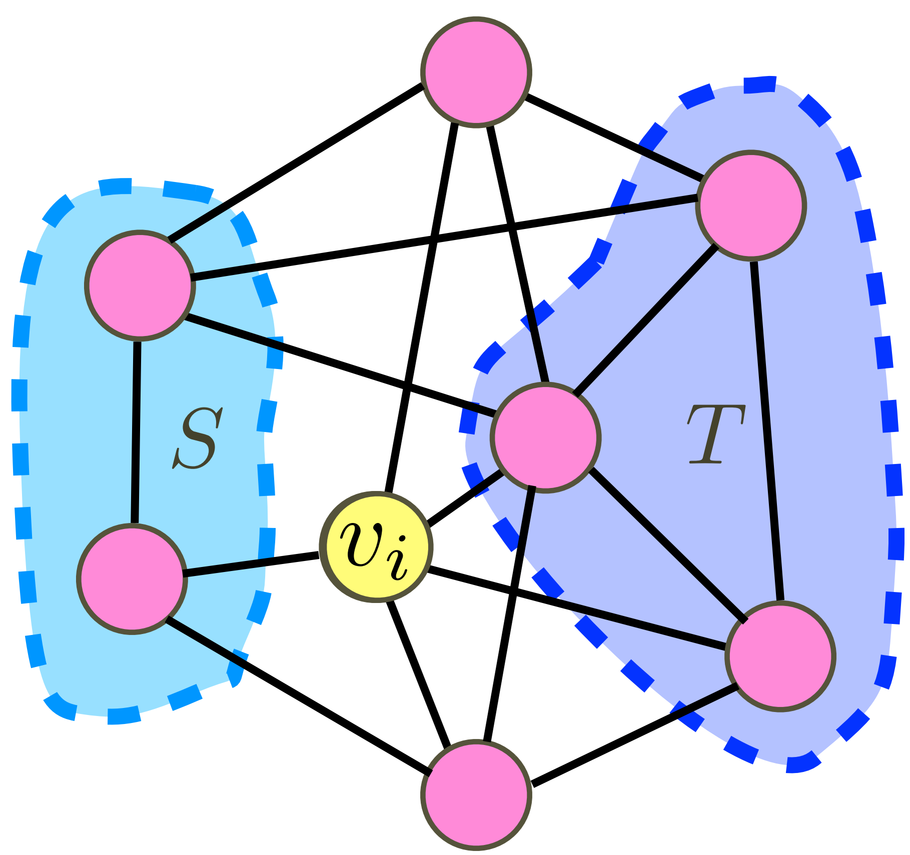
Greedy *Heuristics*

initially, $S = T = \varnothing$

for $i = 1, 2, \ldots, n$

v_i joins one of S, T

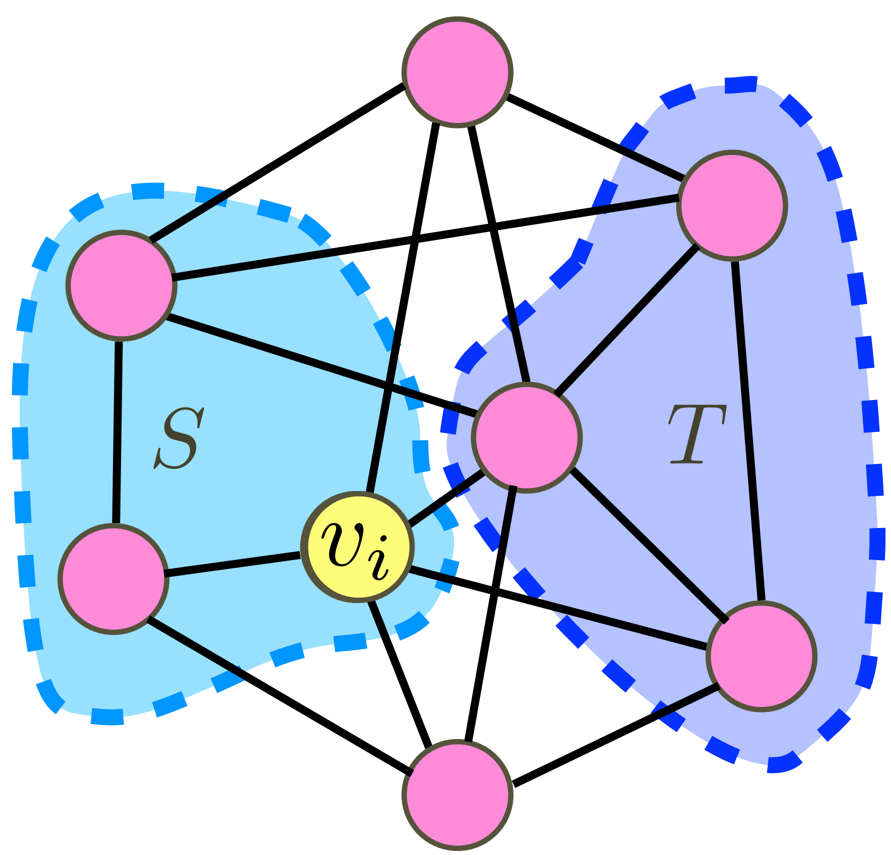
to maximize *current* $E(S,T)$



$E(S,T) = \{uv \in E \mid u \in S, v \in T\}$

Greedy *Heuristics*

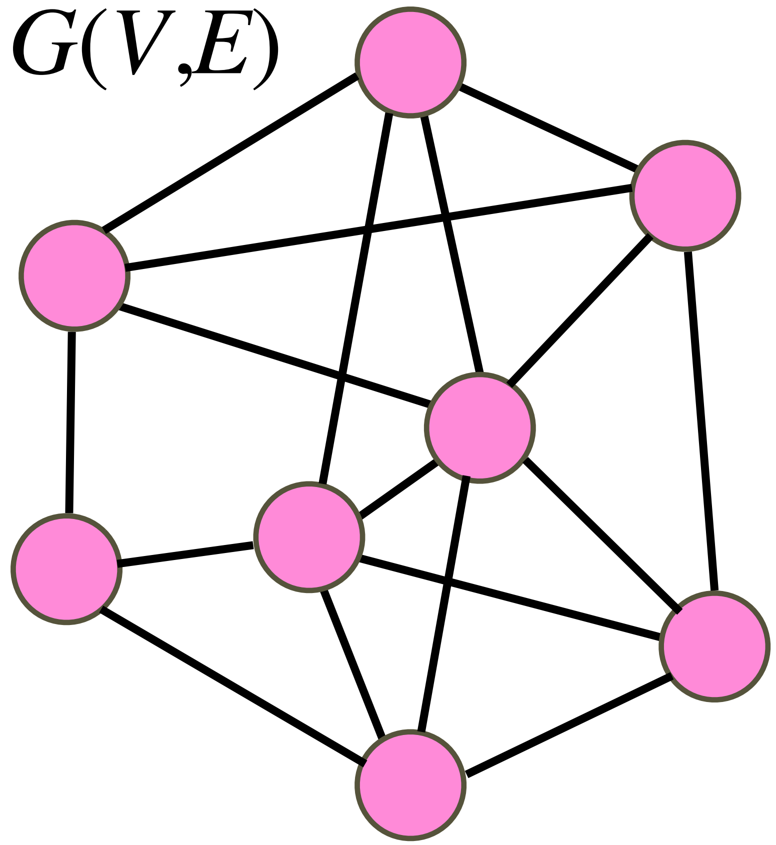
initially, $S = T = \emptyset$
for $i = 1, 2, \ldots, n$
  v_i joins one of S, T
  to maximize *current* $E(S,T)$



$$E(S,T) = \{uv \in E \mid u \in S, v \in T\}$$

Approximation Ratio

instance $G(V,E)$



algorithm A:

initially, $S=T=\emptyset$
for $i = 1,2, \ldots, n$
    v_i joins one of S, T
    to maximize *current* $E(S,T)$

OPT_G: value of maximum cut of G

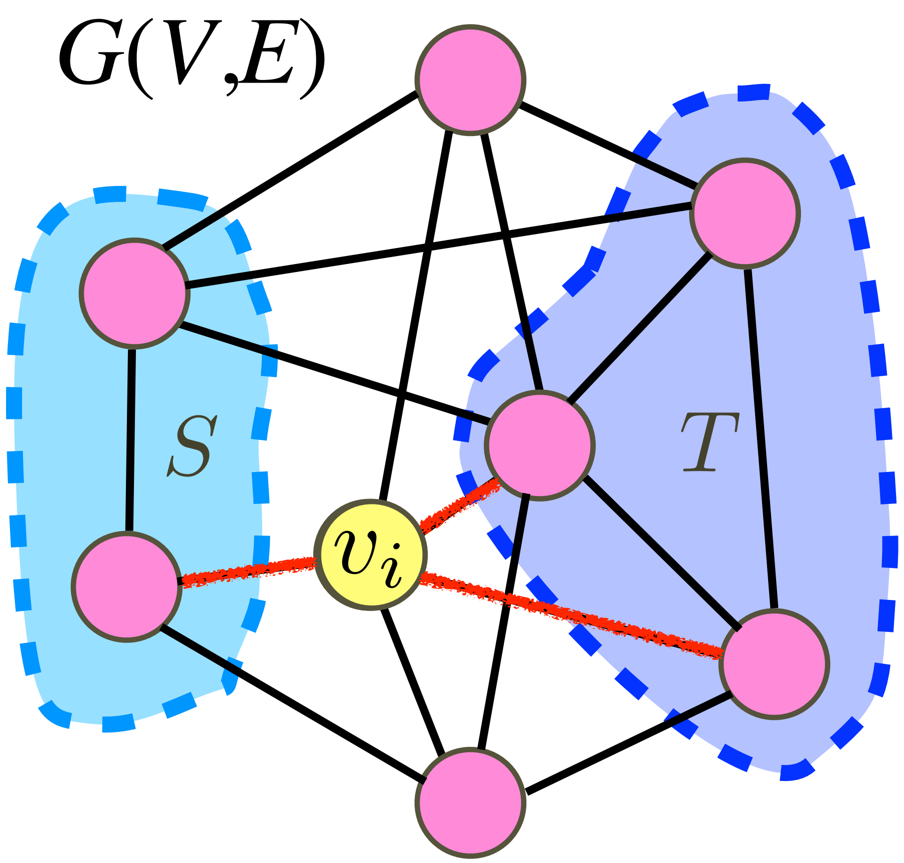
SOL_G: value of the cut returned by algorithm A on G

algorithm A has *approximation ratio* α if

$$\forall \text{ input } G, \quad \frac{\mathrm{SOL}_G}{\mathrm{OPT}_G} \geq \alpha$$

Approximation Algorithm

initially, $S = T = \emptyset$
for $i = 1, 2, \ldots, n$
  v_i joins one of S, T
  to maximize *current* $E(S,T)$



$E(S,T) = \{uv \in E \mid u \in S, v \in T\}$

$$\frac{\mathrm{SOL}_G}{\mathrm{OPT}_G} \geq \frac{\mathrm{SOL}_G}{|E|} \geq \frac{1}{2}$$

$\forall v_i$, $\geq 1/2$ of $|E(S_i, v_i)| + |E(T_i, v_i)|$ contributes to SOL_G

$$|E| = \sum_{i=1}^{n} \left(|E(S_i, v_i)| + |E(T_i, v_i)| \right)$$

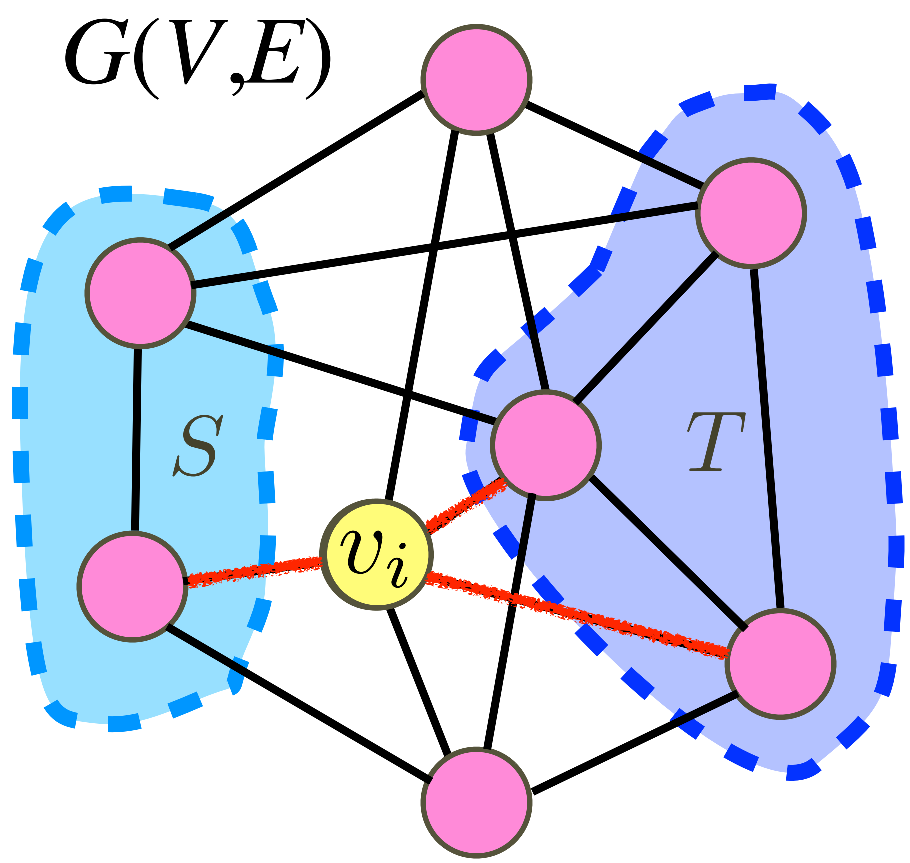
Approximation Algorithm

initially, $S=T=\emptyset$

for $i = 1,2, \ldots, n$

v_i joins one of S, T

to maximize *current* $E(S,T)$



$E(S,T) = \{uv \in E \mid u\in S, v\in T\}$

$$\frac{\mathrm{SOL}_G}{\mathrm{OPT}_G} \geq \frac{\mathrm{SOL}_G}{|E|} \geq \frac{1}{2}$$

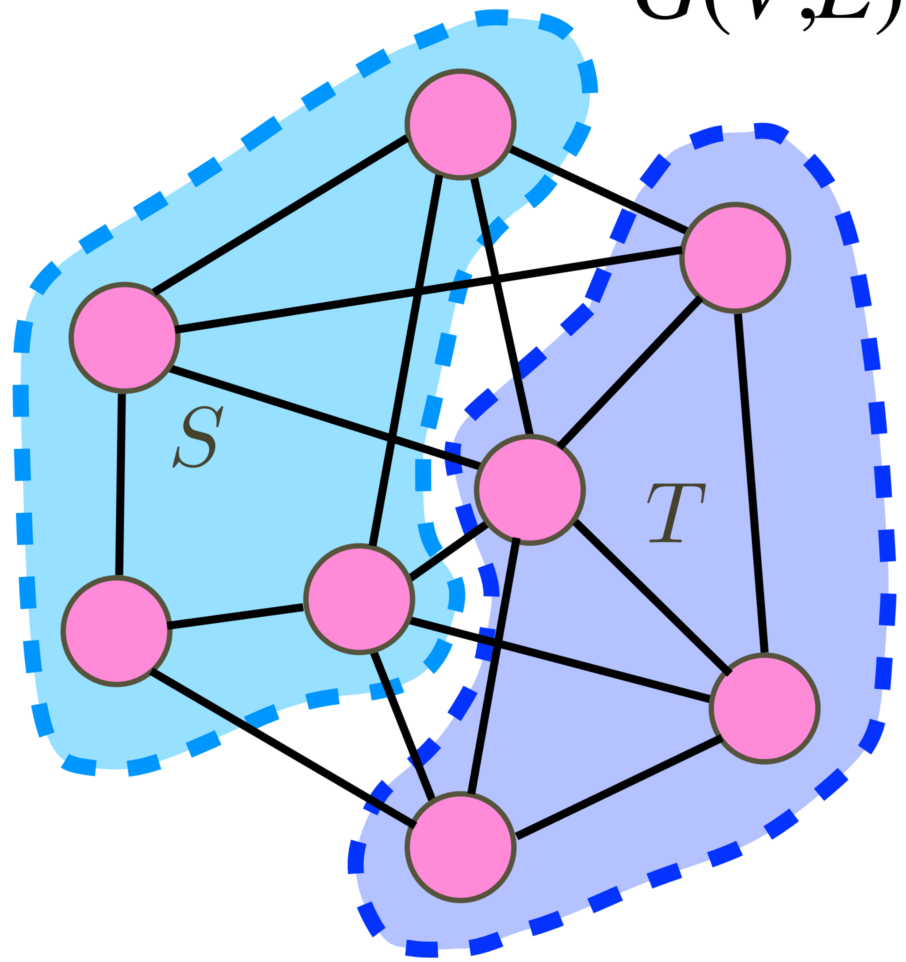
approximation ratio: $1/2$

running time: $O(m)$

Max-Cut

- Partition V into two parts: S and T
- Maximize the cut $E(S,T)$
- **NP-hard**
  - one of Karp's 21 NP-complete problems
- greedy algorithm: 0.5-approximation

$G(V,E)$



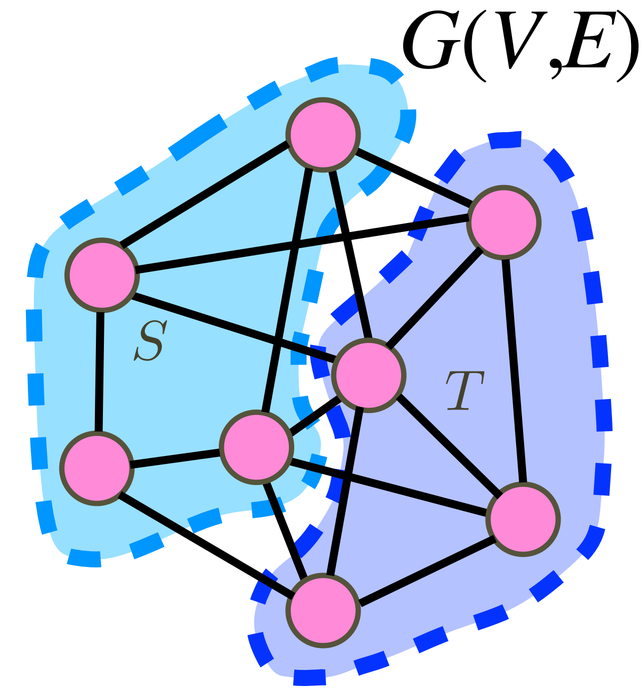
$$E(S,T) = \{uv \in E \mid u \in S, v \in T\}$$

Random Cut

for each vertex $v \in V$

uniform & independent $Y_v \in \{0,1\}$

$$Y_v = 1 \Rightarrow v \in S$$
$$Y_v = 0 \Rightarrow v \in T$$



for each edge $uv \in E$

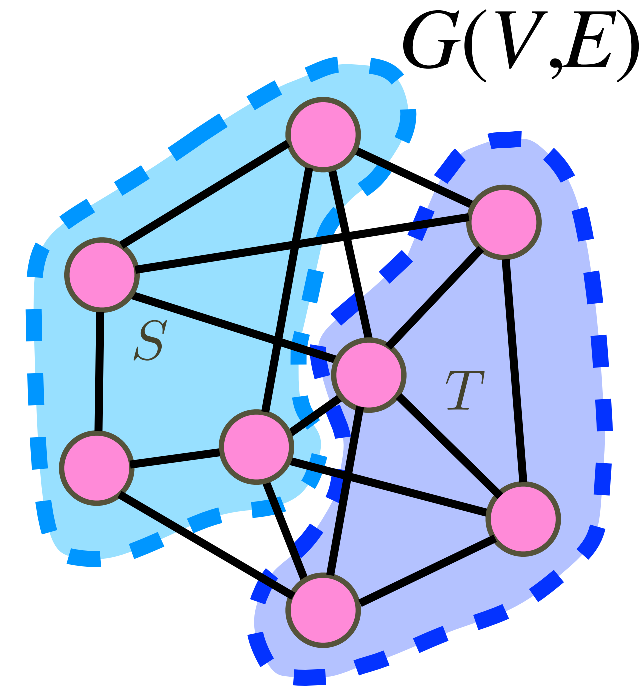
$$Y_{uv} = \begin{cases} 1 & Y_u \neq Y_v \\ 0 & Y_u = Y_v \end{cases} \qquad |E(S,T)| = \sum_{uv \in E} Y_{uv}$$

$$\mathbf{E}[|E(S,T)|] = \sum_{uv \in E} \Pr[Y_u \neq Y_v] = \frac{|E|}{2} \geq \frac{OPT}{2}$$

Random Cut

for each vertex $v \in V$

uniform & 2-wise independent $Y_v \in \{0,1\}$



$$Y_v = 1 \Rightarrow v \in S$$

$$Y_v = 0 \Rightarrow v \in T$$

for each edge $uv \in E$

$$Y_{uv} = \begin{cases} 1 & Y_u \neq Y_v \\ 0 & Y_u = Y_v \end{cases} \qquad |E(S,T)| = \sum_{uv \in E} Y_{uv}$$

$$\mathbf{E}[|E(S,T)|] = \sum_{uv \in E} \Pr[Y_u \neq Y_v] = \frac{|E|}{2} \geq \frac{OPT}{2}$$

Definition:

Events $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_n$ are **mutually independent** if for any subset $I \subseteq \{1, 2, \ldots, n\}$,

$$\Pr\left[\bigwedge_{i \in I} \mathcal{E}_i\right] = \prod_{i \in I} \Pr[\mathcal{E}_i].$$

Definition:

Random variables $X_1, X_2, \ldots, X_n$ are **mutually independent** if for any subset $I \subset [n]$ and any values x_i, where $i \in I$,

$$\Pr\left[\bigwedge_{i \in I} (X_i = x_i)\right] = \prod_{i \in I} \Pr[X_i = x_i].$$

k-wise Independence

Definition:
Events $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_n$ are k-**wise independent** if for any subset $I \subseteq \{1, 2, \ldots, n\}$, with $|I| \le k$

$$\Pr\left[\bigwedge_{i \in I} \mathcal{E}_i\right] = \prod_{i \in I} \Pr[\mathcal{E}_i].$$

Definition:
Random variables $X_1, X_2, \ldots, X_n$ are k-**wise independent** if for any subset $I \subset [n]$ and any values x_i, where $i \in I$, with $|I| \le k$

$$\Pr\left[\bigwedge_{i \in I}(X_i = x_i)\right] = \prod_{i \in I} \Pr[X_i = x_i].$$

pairwise: 2-wise

2-wise Independent Bits

uniform & independent bits: (random source)

$$X_1, X_2, \ldots, X_m \in \{0, 1\}$$

Goal: 2-wise independent uniform bits:

$$Y_1, Y_2, \ldots, Y_n \in \{0, 1\} \qquad n \gg m$$

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

nonempty subsets:

$$\emptyset \neq S_1, S_2, \ldots, S_{2^m - 1} \subseteq \{1, 2, \ldots, m\}$$

$$Y_j = \bigoplus_{i \in S_j} X_i$$

uniform & independent bits: $X_1, X_2, \ldots, X_m \in \{0,1\}$

nonempty subsets: $S_1, S_2, \ldots, S_{2^m-1} \subseteq \{1,2,\ldots,m\}$

$$Y_j = \bigoplus_{i \in S_j} X_i$$

2-wise independent uniform bits:

$$Y_1, Y_2, \ldots, Y_{2^m-1} \in \{0,1\}$$

$\log_2 n$ total random bits

⇒ n-1 pairwise independent bits

Derandomization

for each vertex $v \in V$

uniform & 2-wise independent $Y_v \in \{0, 1\}$

$Y_v = 1 \Rightarrow v \in S$

$Y_v = 0 \Rightarrow v \in T$

for each edge $uv \in E$

$$\mathbf{E}[|E(S,T)|] = \sum_{uv \in E} \Pr[Y_u \neq Y_v] = \frac{|E|}{2} \geq \frac{OPT}{2}$$

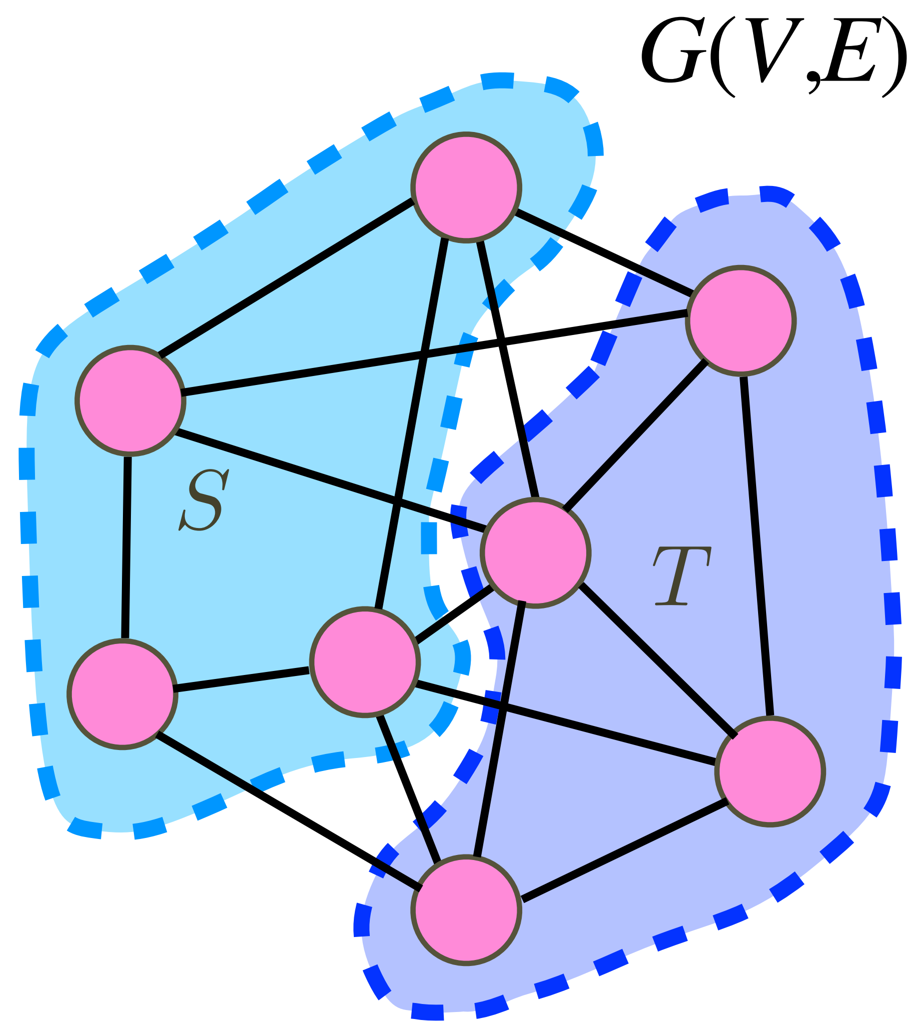
$V = \{v_1, v_2, \ldots, v_n\}$

$Y_{v_1}, Y_{v_2}, \ldots, Y_{v_n}$ constructed from $\lceil \log_2(n+1) \rceil$ bits

try all $2^{\lceil \log_2(n+1) \rceil} = O(n^2)$ possibilities!

Max-Cut

- Partition V into two parts: S and T
- Maximize the cut $E(S,T)$
- **NP-hard**
- greedy algorithm: 0.5-approx.
- best known approx. ratio for poly-time algorithms: 0.878~



$$E(S,T) = \{uv \in E \mid u \in S, v \in T\}$$

Mathematical Programming

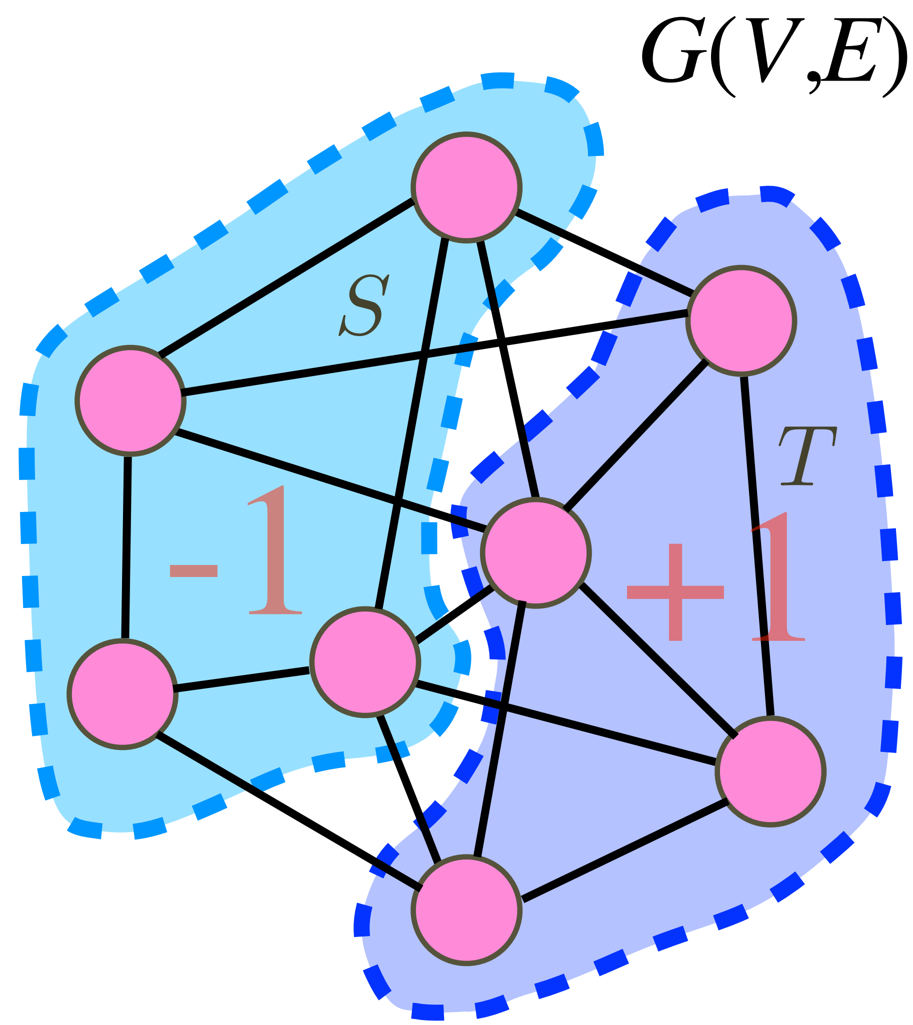
maximize:

$$\frac{1}{2}\sum_{uv \in E}(1 - y_u y_v)$$

subject to:

$$y_v \in \{-1, +1\}, \quad \forall v \in V$$

$$y_v = \begin{cases} -1 & v \in S \\ +1 & v \in T \end{cases}$$



$E(S,T) = \{uv \in E \mid u \in S, v \in T\}$

Relaxation

maximize: $\frac{1}{2}\sum_{uv\in E}(1-y_u y_v)$

subject to: $y_v \in \{-1,+1\}, \quad \forall v \in V$

semidefinite programming (SDP):

maximize: $\frac{1}{2}\sum_{uv\in E}(1-z_u \cdot z_v)$

subject to: $\|z_v\|_2 = 1, \quad \forall v \in V$

$z_v \in \mathbb{R}^n$

solvable in poly-time

$\mathrm{OPT}^{\mathrm{SDP}} \geq \mathrm{OPT}^{\text{max-cut}}$

Rounding

maximize: $\frac{1}{2}\sum_{uv\in E}(1 - z_u \cdot z_v)$

subject to: $\|z_v\|_2 = 1, \quad \forall v \in V$

$z_v \in \mathbb{R}^n$

optimal SDP solution: $z_v^* \in \mathbb{R}^n$ → randomized rounding → $v\in S$ or $v\in T$

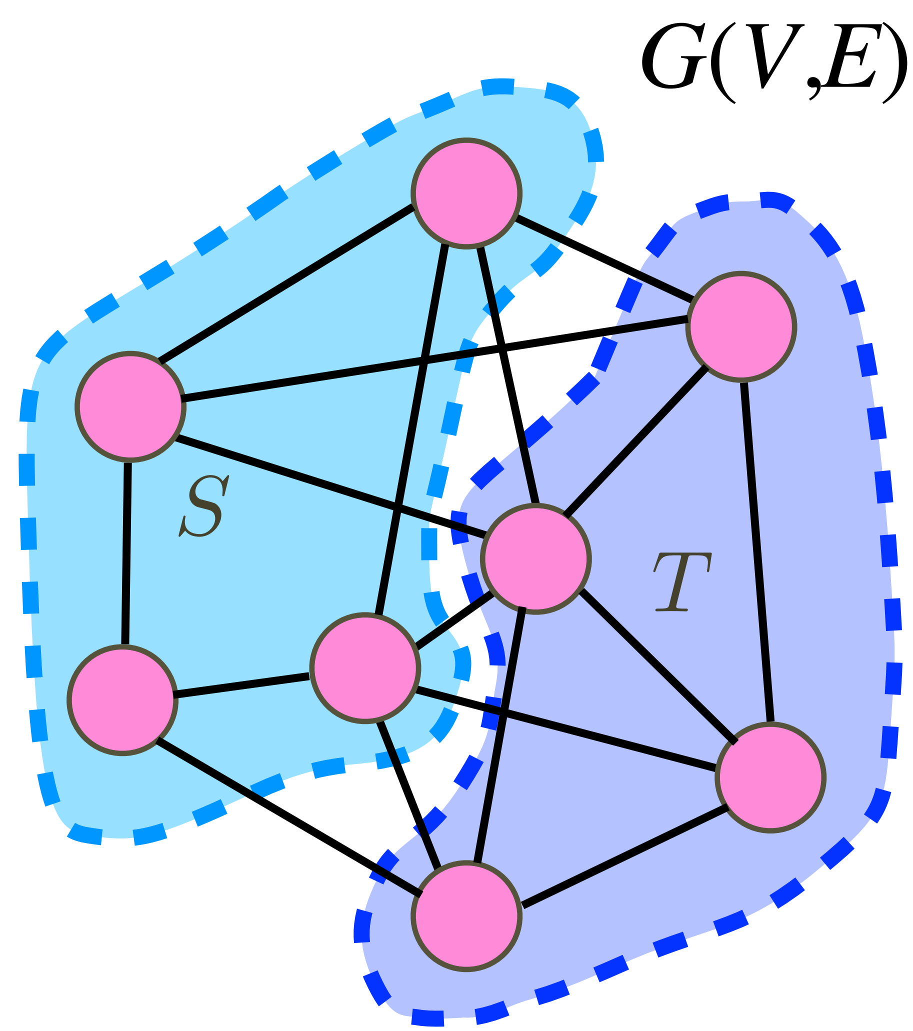
$$\mathbf{E}[|E(S,T)|] \geq \sum_{uv\in E} 0.878 \cdot \frac{1}{2}(1 - z_u^* \cdot z_v^*)$$

$$= 0.878 \ \mathrm{OPT}^{\mathrm{SDP}}$$

$$\geq 0.878 \ \mathrm{OPT}^{\text{max-cut}}$$

Max-Cut

- Partition V into two parts: S and T
- Maximize the cut $E(S,T)$
- **NP-hard**
- greedy algorithm: 0.5-approx.
- best known approx. ratio for poly-time algorithms: 0.878~
- unique game conjecture: no poly-time algorithm with approx. ratio >0.878~



$$E(S,T) = \{uv \in E \mid u \in S, v \in T\}$$