Randomized Algorithms I

- Probability
- Contention Resolution
- Minimum Cut

Philip Bille

Randomized Algorithms I

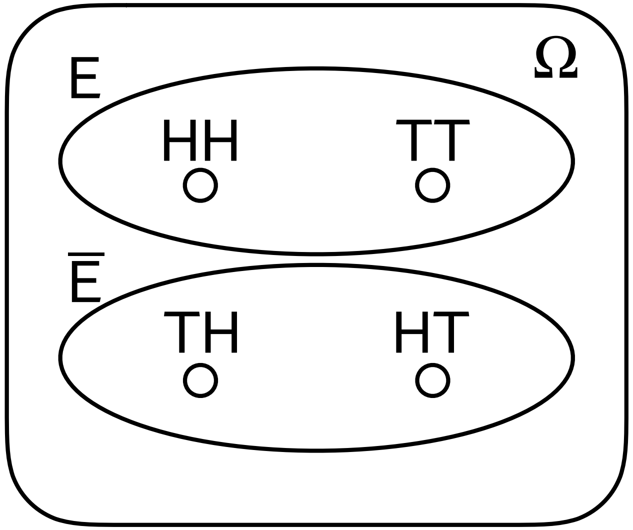
- Probability
- Contention Resolution
- Minimum Cut

Probability

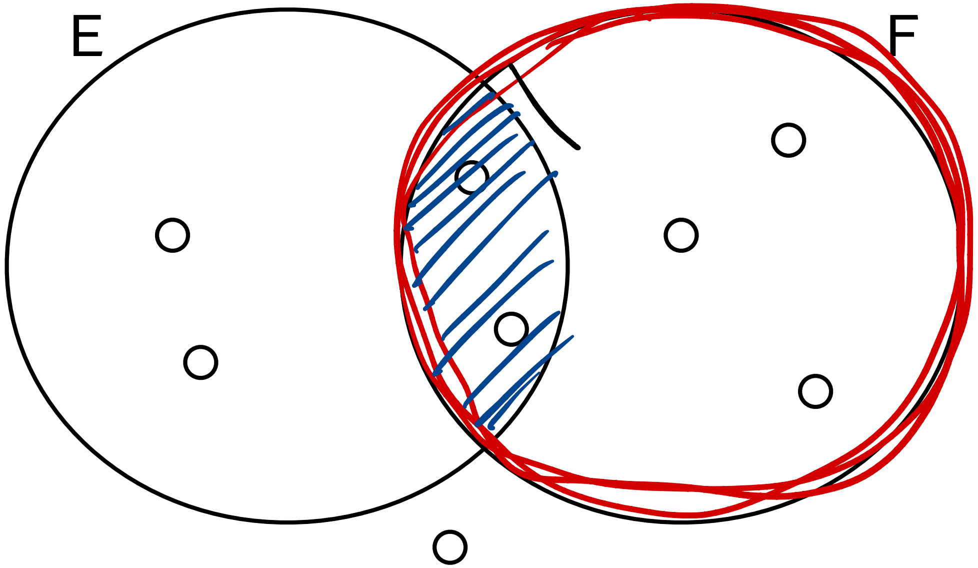
- Probability spaces.
  - Set of possible outcomes Ω.
  - Each element $i \in \Omega$ has probability $p(i) \geq 0$ and $\sum_{i \in \Omega} p(i) = 1$.
  - Event E is a subset of Ω and probability of E is $\Pr(E) = \sum_{i \in E} p(i)$.
  - The complementary event $\bar{E}$ is $\Omega - E$ and $\Pr(\bar{E}) = 1 - \Pr(E)$.

H = heads T = tails

- Example. Flip two fair coins.
  - $\Omega = \{HH, HT, TH, TT\}$.
  - $p(i) = 1/4$ for each outcome i.
  - Event E = "the coins are the same"
  - $\Pr(\bar{E}) = 1/2$.

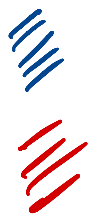


Probability



- Conditional probability.
  - What is the probability that event E occurs given that event F occurred?
  - The conditional probability of E given F:

$$\Pr(E \mid F) = \frac{\Pr(E \cap F)}{\Pr(F)}$$



- Example.
  - $\Pr(E \mid F) = \frac{\Pr(E \cap F)}{\Pr(F)} = \frac{2/8}{5/8} = \frac{2}{5}$

Probability

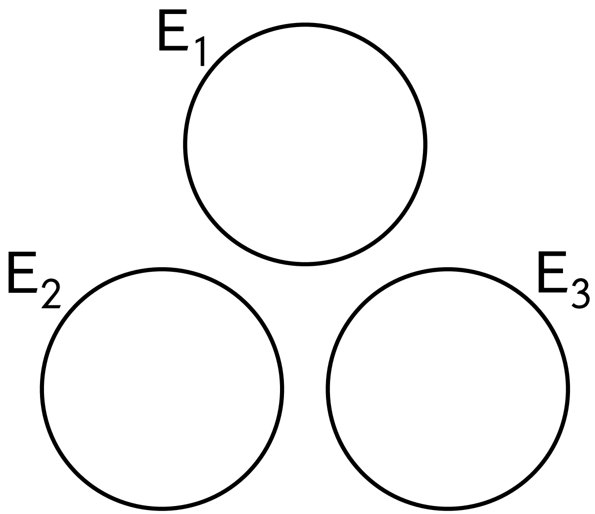
- Independence.
  - Events E and F are independent if information about E does not affect outcome of F and vice versa.

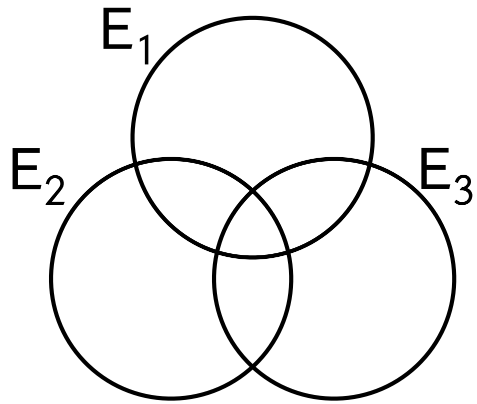
$$\Pr(E \mid F) = \Pr(E) \qquad \Pr(F \mid E) = \Pr(F)$$

- Same as $\Pr(E \cap F) = \Pr(E) \cdot \Pr(F)$

Probability

- Union bound.
  - What is the probability that any of event E_1, ..., E_k will happen, i.e., what is $\Pr(E_1 \cup E_2 \cup \cdots \cup E_k)$?





  - If events are disjoint, $\Pr(E_1 \cup \cdots \cup E_k) = \Pr(E_1) + \cdots + \Pr(E_k)$.
  - If events overlap, $\Pr(E_1 \cup \cdots \cup E_k) < \Pr(E_1) + \cdots + \Pr(E_k)$.
  - In both cases, the union bound holds:

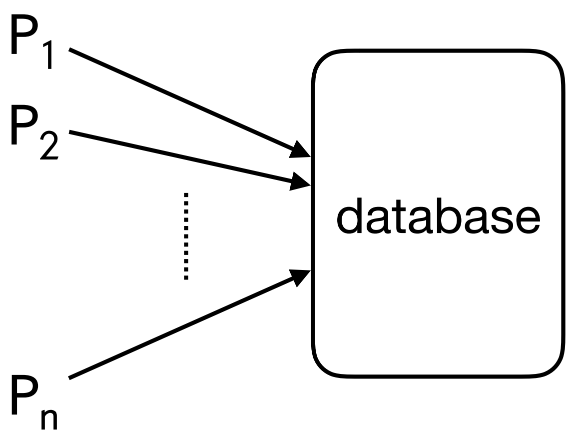
$$\Pr(E_1 \cup \cdots \cup E_k) \leq \Pr(E_1) + \cdots + \Pr(E_k)$$

Randomized Algorithms I

- Probability
- Contention Resolution
- Minimum Cut

Contention Resolution

- Contention resolution. Consider n processes $P_1, \ldots, P_n$ trying to access a shared database:
  - If two or more processes access database at the same time, all processes are locked out.
  - Processes cannot communicate.
- Goal. Come up with a protocol to ensure all processes will access database.
- Challenge. Need symmetry breaking paradigm.

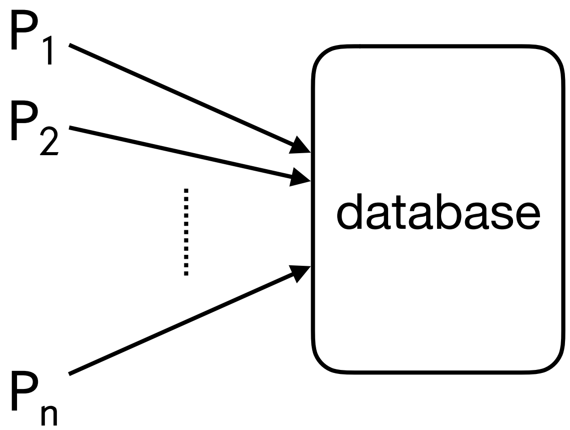


Contention Resolution

- Applications.
  - Distributed communication and interference.
  - Illustrates simplicity and power of randomized algorithms.

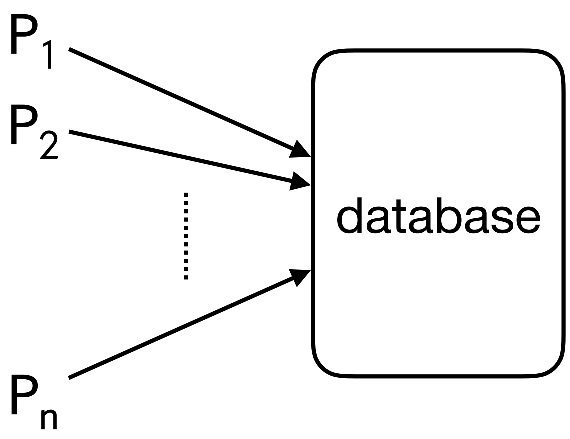
Contention Resolution

- Protocol. Each process accesses the database at time t with probability $p = 1/n$.



Contention Resolution

- Analysis. How do we analyze the protocol?



Contention Resolution

- Success for a single process in a single round.
  - $S_{i,t} =$ event that P_i successfully accesses database at time t.

$p = 1/n$

$$\Pr(S_{i,t}) = p(1-p)^{n-1} = \frac{1}{n}\left(1-\frac{1}{n}\right)^{n-1} \geq \frac{1}{en}$$

probability that process i requests access.

probability that no other process requests access.

$\left(1-\frac{1}{n}\right)^{n-1}$ converges to 1/e from above.

Contention Resolution

- Failure for a single process in rounds $1, \ldots, t$.
  - $F_{i,t}$ = event that P_i fails to access database in any of rounds $1, \ldots, t$.

$$\Pr\left(F_{i,t}\right) = \Pr\left(\bigcap_{r=1}^{t} \overline{S_{i,r}}\right) = \prod_{r=1}^{t} \Pr\left(\overline{S_{i,r}}\right) = \left(1 - \frac{1}{n}\left(1-\frac{1}{n}\right)^{n-1}\right)^{t} \leq \left(1 - \frac{1}{en}\right)^{t}$$

$\Pr(S_{i,r})$

probability that P_i does not succeed in round 1 and round 2 and ... and round t.

independence.

$\Pr\left(S_{i,t}\right) \geq \frac{1}{en}$

- $t = \lceil en \rceil \Rightarrow \Pr\left(F_{i,t}\right) \leq \left(1 - \frac{1}{en}\right)^{\lceil en \rceil} \leq \left(1 - \frac{1}{en}\right)^{en} \leq \frac{1}{e}$

- $t = \lceil en \rceil (c \ln n) \Rightarrow \Pr\left(F_{i,t}\right) \leq \left(\frac{1}{e}\right)^{c \ln n} = \frac{1}{n^c}$

$\left(1 - \frac{1}{n}\right)^{n}$ converges to $1/e$ from below.

Contention Resolution

- Failure for at least one process in rounds $1, \ldots, t$.
  - $F_t =$ event that at least one of n processes fails to access database in any of rounds $1, \ldots, t$.

$$\Pr(F_t) = \Pr\left(\bigcup_{i=1}^{n} F_{i,t}\right) \le \sum_{i=1}^{n} \Pr(F_{i,t}) \le n\left(1 - \frac{1}{en}\right)^t$$

probability that any one of $P_1, \ldots, P_n$ fails in rounds $1, \ldots, t$

union bound

$$\Pr(F_{i,t}) \le \left(1 - \frac{1}{en}\right)^t$$

  - $t = \lceil en \rceil 2 \ln n \Rightarrow \Pr(F_t) \le n\left(1 - \frac{1}{en}\right)^{\lceil en \rceil 2 \ln n} \le n\left(\frac{1}{e}\right)^{2 \ln n} = \frac{n}{n^2} = \frac{1}{n}.$
  - $\Rightarrow$ Probability that all processes successfully access the database after $\lceil en \rceil 2 \ln n$ rounds is at least $1 - 1/n$.

Contention Resolution

- Conclusion. After $\lceil en \rceil 2 \ln n$ rounds all processes have accessed database with probability at least $1 - 1/n$.

- Success probability.
  - For large n probability is very close to 1.
  - More rounds will further increase probability of success.

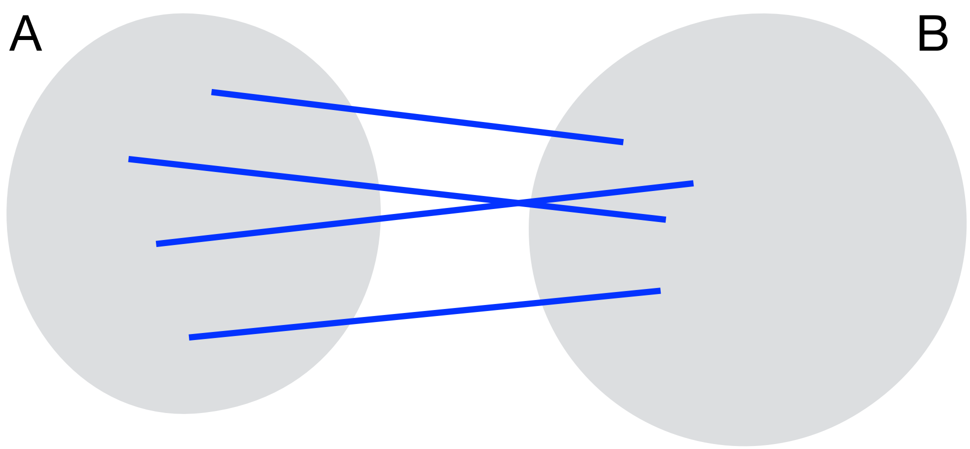
- Simplicity.
  - Very simple and effective protocol.
  - Difficult to solve deterministically.

Randomized Algorithms I

- Probability
- Contention Resolution
- Minimum Cut

Minimum Cut

- Graphs. Consider undirected, connected graph G = (V,E).
- Cuts.
  - A cut (A,B) is a partition of V into two non-empty disjoint sets A and B.
  - The size of a cut (A,B) is the number of edges crossing the cut.
  - A minimum cut is a cut of minimum size.



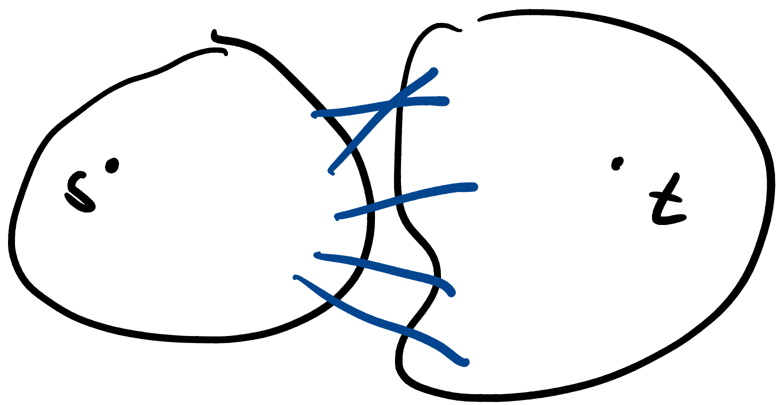
Minimum Cut

- Applications.
  - Network fault tolerance.
  - Image segmentation.
  - Parallel computation
  - Social network analysis.
  - ...

Minimum Cut

- Which solutions do we know?

Min s-t-Cut



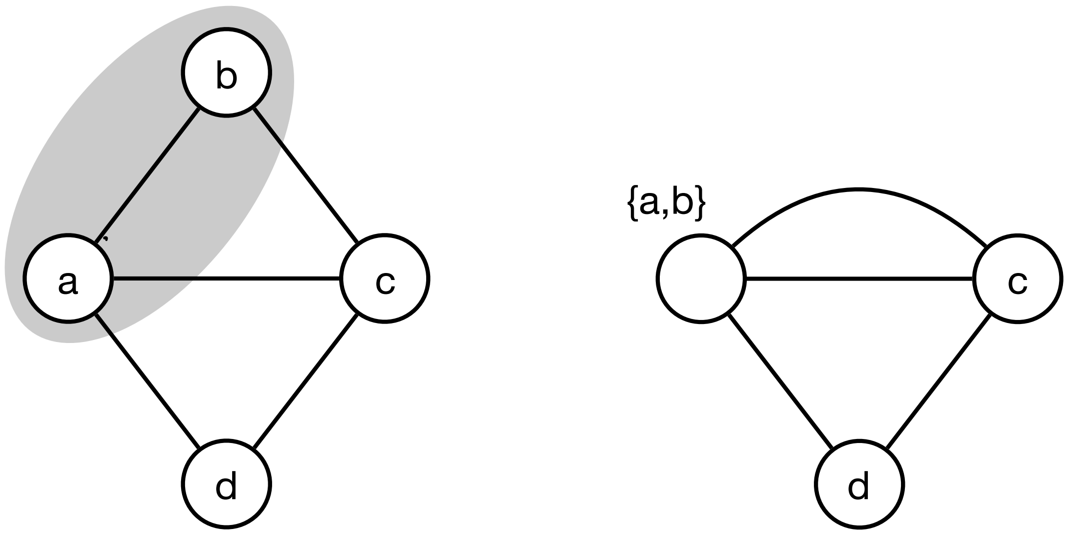
for all s, t with $s \neq t$:

Find min s-t-Cut

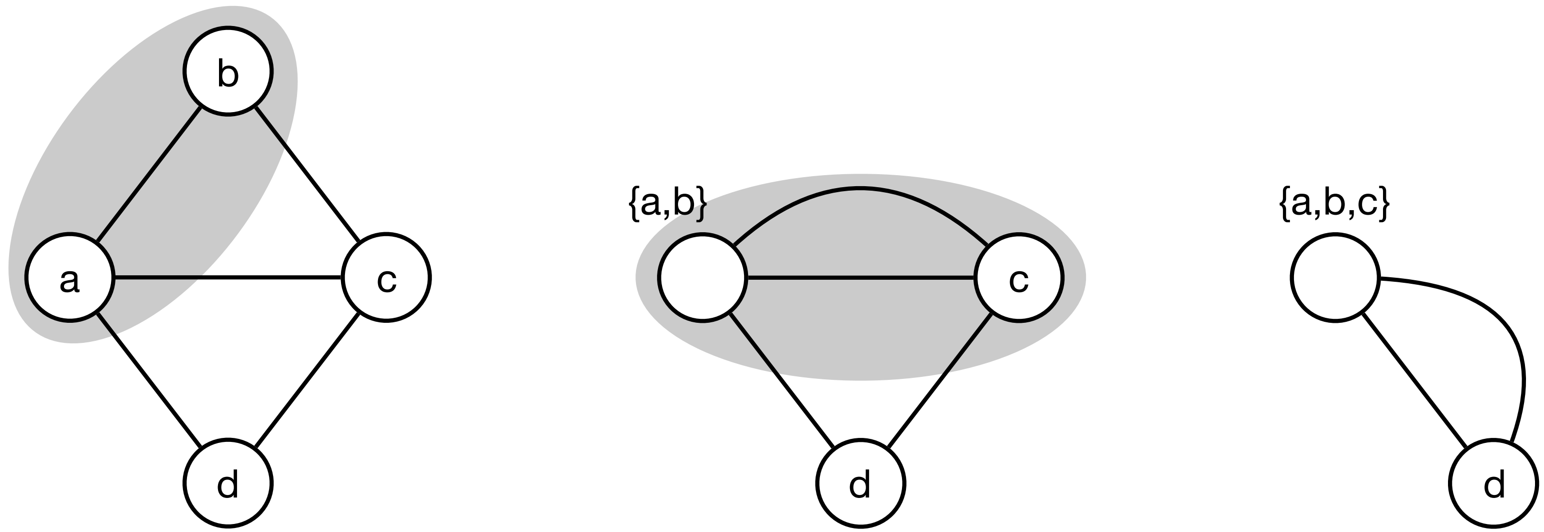
Output minimum

$$\frac{n^2 \cdot LZ(\text{MaxFlow})}{\overset{?}{=} \Omega(n^3)}$$

Minimum Cut



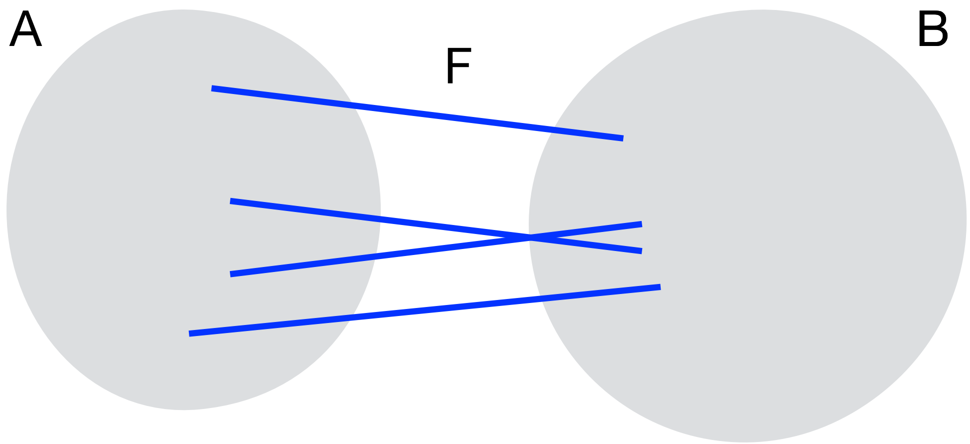
- Contraction algorithm.
  - Pick edge e = (u,v) uniformly at random.
  - Contract e.
    - Replace e by single vertex w.
    - Preserve edges, updating endpoints of u and v to w.
    - Preserve parallel edges, but remove self-loops.
  - Repeat until two vertices a and b left.
  - Return cut (all vertices contracted into a, all vertices contracted into b).



cut is ({a,b,c}, {d}) of size 2

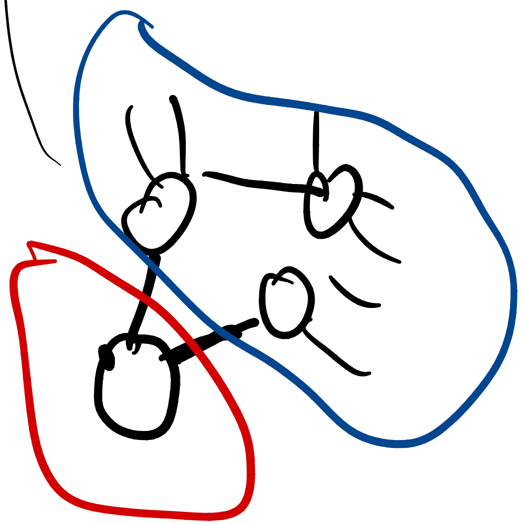
Minimum Cut

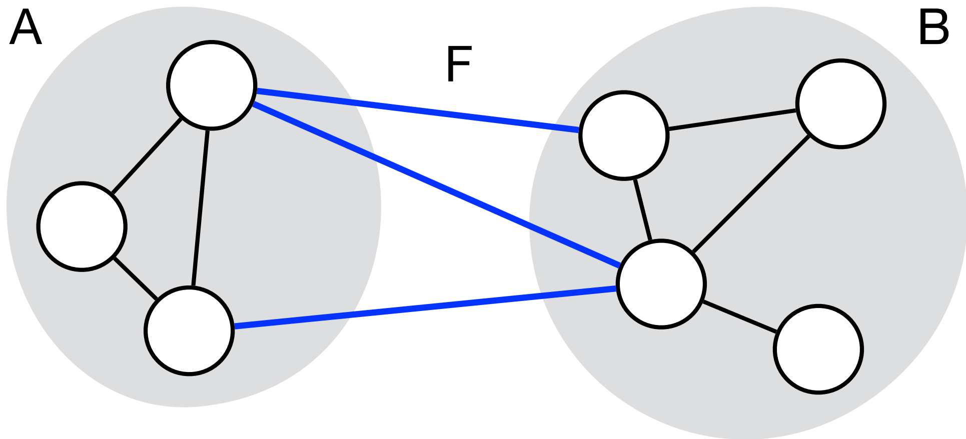
- Analysis.
  - Consider minimum cut (A,B) with crossing edges F.
  - What is the probability that the contraction algorithm returns (A,B)?



Minimum Cut

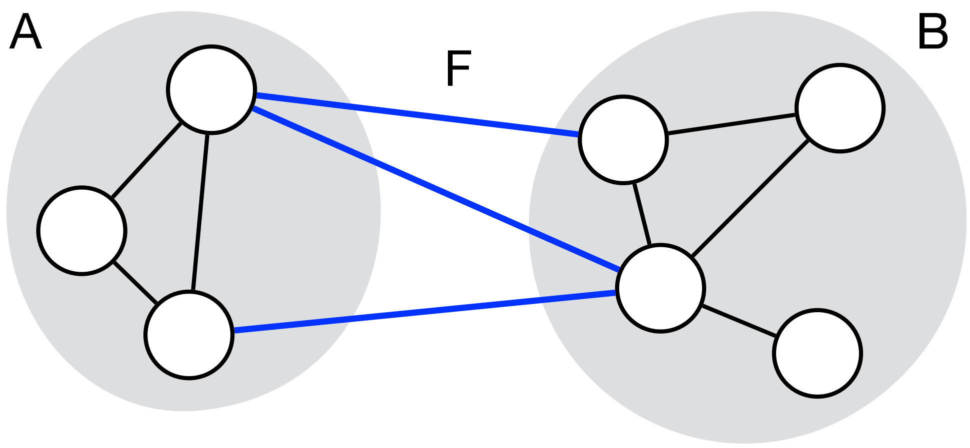
- Round 1.
  - What is the probability that we contract an edge from F in round 1?
  - Each vertex has deg $\geq |F|$ (otherwise smaller cut exists) $\Rightarrow \sum_{v \in V} \deg(v) \geq |F|n$.
  - $\sum_{v \in V} \deg(v) = 2m \Rightarrow m = \frac{\sum_{v \in V} \deg(v)}{2} \geq \frac{|F|n}{2}$.
  - Probability we contract edge from F is $= \frac{|F|}{m} \leq \frac{|F|}{|F|n/2} = \frac{2}{n}$.





Minimum Cut

- Round j+1.
  - What is the probability that we contract an edge in round $j+1$ from F, given that no edge from F was contracted in rounds $1, \ldots, j$?
  - G' is graph after j rounds with $n-j$ nodes and no edges from F was contracted in rounds $1, \ldots, j$.
  - Every cut in G' is a cut in G $\Rightarrow$ at least $|F|$ edges incident to every node in G'
  - $\Rightarrow$ G' contains at least $\dfrac{|F|(n-j)}{2}$ edges $\Rightarrow$ probability is $\leq \dfrac{|F|}{m} = \dfrac{2}{n-j}$.



Minimum Cut

- Success after all rounds.
  - E_j = event that an edge from F is not contracted in round j.
  - The probability that we return the correct minimum cut is $\Pr\left(E_{n-2} \cap \cdots \cap E_1\right)$.
  - We know:
    - $\Pr\left(E_1\right) \geq 1 - \frac{2}{n}$.
    - $\Pr\left(E_{j+1} \mid E_1 \cap \cdots \cap E_j\right) \geq 1 - \frac{2}{n-j}$.
  - Conditional probability definition + algebra $\Rightarrow \Pr\left(E_1 \cap \cdots \cap E_{\cancel{j+1}\; n-2}\right) \geq \frac{2}{n^2}$.

Minimum Cut

- Conclusion.
  - We return the correct minimum cut with probability $\geq 2/n^2$ in polynomial time.

- Probability amplification.
  - Correct solution only with very small probability
  - Run contraction algorithm many times and return smallest cut.
  - With $n^2 \ln n$ runs with independent random choices the probability of failure to find minimum cut is $\leq \left(1 - \frac{2}{n^2}\right)^{n^2 \ln n} \leq \left(\frac{1}{e}\right)^{2 \ln n} = \frac{1}{n^2}$.
- Time.
  - $\Theta(n^2 \log n)$ iterations that take $\Omega(m)$ time each.
  - More techniques and tricks $\Rightarrow$ $m \log^{O(1)} n$ time solution. [Karger 2000]

Minimum Cut

- Monte Carlo algorithm.
  - Randomized algorithm.
  - Guarantee on running time, likely to find correct answer.

MC = mostly correct

- Las Vegas algorithm.
  - Randomized algorithm.
  - Guaranteed to find the correct answer, likely to be fast.

Randomized Algorithms I

- Probability
- Contention Resolution
- Minimum Cut