Chapter 6
Randomization

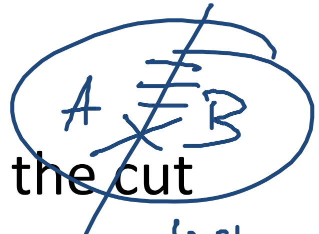
Algorithm Theory
WS 2013/14

Fabian Kuhn

Minimum Cut

Reminder: Given a graph $G = (V, E)$, a cut is a partition (A, B) of V such that $V = A \cup B$, $A \cap B = \emptyset$, $A, B \neq \emptyset$

Size of the cut (A, B): # of edges crossing the cut



- For weighted graphs, total edge weight crossing the cut

edge connectivity

Goal: Find a cut of minimal size (i.e., of size $\lambda(G)$)

Maximum-flow based algorithm:

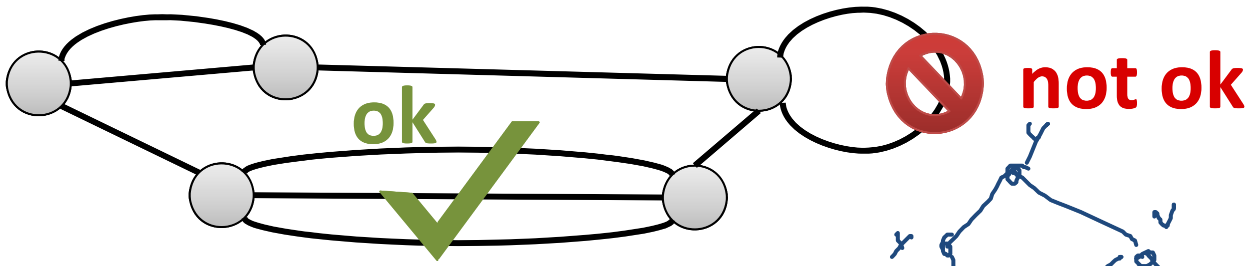
- Fix s, compute min s-t-cut for all $t \neq s$
- $O\big(m \cdot \lambda(G)\big) = O(mn)$ per s-t cut
- Gives an $O\big(mn\lambda(G)\big) = O(mn^2)$-algorithm

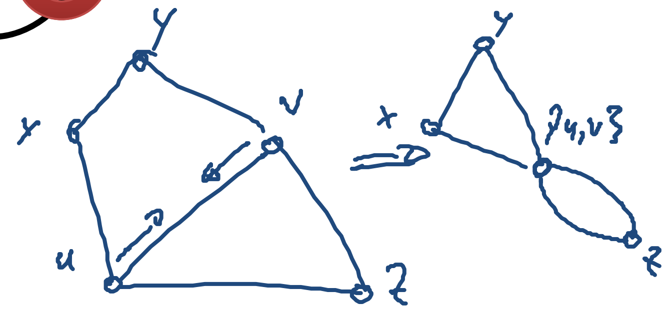
$O(n^4)$

Best-known deterministic algorithm: $O(mn + n^2 \log n)$

Edge Contractions

- In the following, we consider multi-graphs that can have multiple edges (but no self-loops)





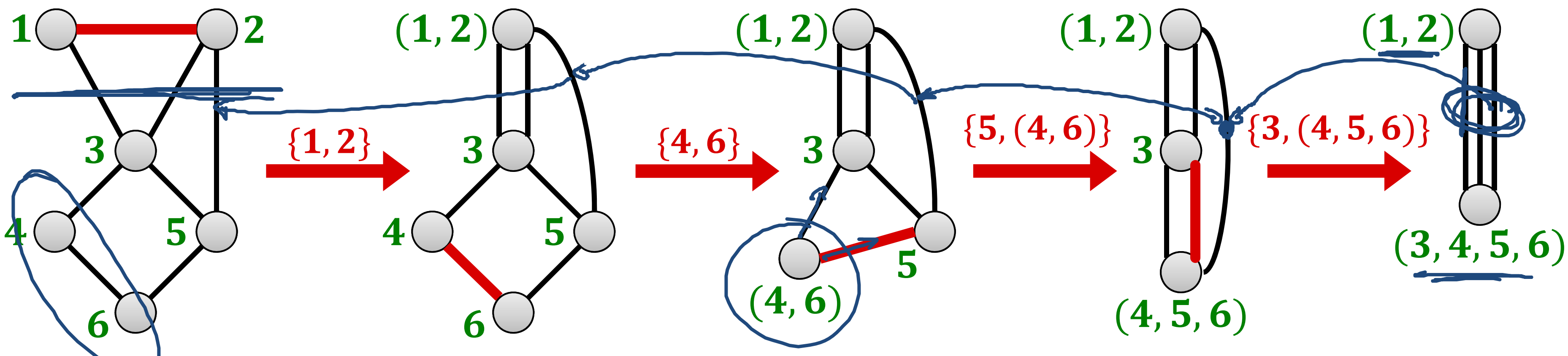
Contracting edge $\{u, v\}$:

- Replace nodes u, v by new node w
- For all edges $\{u, x\}$ and $\{v, x\}$, add an edge $\{w, x\}$
- Remove self-loops created at node w

Properties of Edge Contractions

Nodes:

- After contracting $\{u, v\}$, the new node represents u and v
- After a series of contractions, each node represents a subset of the original nodes



Cuts:

- Assume in the contracted graph, w represents nodes $S_w \subset V$
- The edges of a node w in a contracted graph are in a one-to-one correspondence with the edges crossing the cut $(S_w, V \setminus S_w)$

Randomized Contraction Algorithm

Algorithm:

while there are > 2 nodes **do**

    contract a uniformly random edge

return cut induced by the last two remaining nodes

(cut defined by the original node sets represented by the last 2 nodes)

Theorem: The random contraction algorithm returns a minimum cut with probability at least $1/O(n^2)$.

- We will show this next.

Theorem: The random contraction algorithm can be implemented in time $O(n^2)$.

- There are $n - 2$ contractions, each can be done in time $O(n)$.
- You will show this in the exercises.

Contractions and Cuts

Lemma: If two original nodes $u, v \in V$ are merged into the same node of the contracted graph, there is a path connecting u and v in the original graph s.t. all edges on the path are contracted.

Proof:

- Contracting an edge $\{x, y\}$ merges the node sets represented by x and y and does not change any of the other node sets.

- The claim the follows by induction on the number of edge contractions.

w $\qquad \underline{S_w}$ $\qquad S_{w,1}$ — $S_{w,2}$

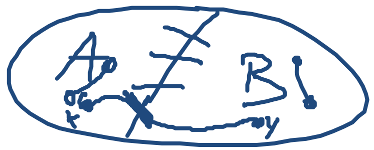
Contractions and Cuts

Lemma: During the contraction algorithm, the edge connectivity (i.e., the size of the min. cut) cannot get smaller.

Proof:

- All cuts in a (partially) contracted graph correspond to cuts of the same size in the original graph G as follows:
  - For a node u of the contracted graph, let S_u be the set of original nodes that have been merged into u (the nodes that u represents)
  - Consider a cut (A, B) of the contracted graph
  - (A', B') with
    $$A' := \bigcup_{u \in A} S_u, \qquad B' := \bigcup_{v \in B} S_v$$
    is a cut of G.
  - The edges crossing cut (A, B) are in one-to-one correspondence with the edges crossing cut (A', B').

Contraction and Cuts



Lemma: The contraction algorithm outputs a cut (A, B) of the input graph G if and only if it never contracts an edge crossing (A, B).

Proof:

1. If an edge crossing (A, B) is contracted, a pair of nodes $u \in A$, $v \in V$ is merged into the same node and the algorithm outputs a cut different from (A, B).

2. If no edge of (A, B) is contracted, no two nodes $u \in A$, $v \in B$ end up in the same contracted node because every path connecting u and v in G contains some edge crossing (A, B)

   In the end there are only 2 sets → output is (A, B)

Getting The Min Cut

$\sum_{v \in V} \deg(v) = 2 \cdot m$

Theorem: The probability that the algorithm outputs a minimum cut is at least $2/n(n-1)$.

To prove the theorem, we need the following claim:

Claim: If the minimum cut size of a multigraph G (no self-loops) is k, G has at least $kn/2$ edges.

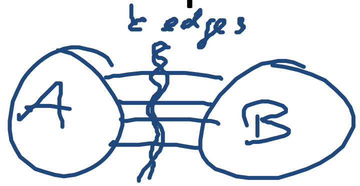
Proof:

- Min cut has size $k \Longrightarrow$ all nodes have degree $\geq k$
  - A node v of degree $< k$ gives a cut $(\{v\}, V \setminus \{v\})$ of size $< k$
- Number of edges $m = {}^{1}\!/_{2} \cdot \sum_{v} \deg(v) \geq \frac{nk}{2}$

Getting The Min Cut

$1, 2, \ldots, n-2$

Theorem: The probability that the algorithm outputs a minimum cut is at least $2/n(n-1)$.



Proof:

- Consider a fixed min cut (A, B), assume (A, B) has size k
- The algorithm outputs (A, B) iff none of the k edges crossing (A, B) gets contracted.
- Before contraction i, there are $n + 1 - i$ nodes
  → and thus $\geq (n + 1 - i)k/2$ edges
- If no edge crossing (A, B) is contracted before, the probability to contract an edge crossing (A, B) in step i is at most

$$\frac{k}{\frac{(n+1-i)k}{2}} = \frac{2}{n+1-i}.$$

Getting The Min Cut

Theorem: The probability that the algorithm outputs a minimum cut is at least $2/n(n-1)$.

Proof:

- If no edge crossing (A, B) is contracted before, the probability to contract an edge crossing (A, B) in step i is at most $^2/_{n+1-i}$.
- Event $\mathcal{E}_i$: edge contracted in step i is **not** crossing (A, B)

$$\mathbb{P}\Big(\mathcal{E}_i \;\Big|\; \underbrace{\mathcal{E}_1 \cap \dots \cap \mathcal{E}_{i-1}}_{\text{first } i-1 \text{ contr. preserve cut } (A,B)}\Big) \geq 1 - \frac{2}{n+1-i} = \frac{n-1-i}{n+1-i}$$

Getting The Min Cut

Theorem: The probability that the algorithm outputs a minimum cut is at least $2/n(n-1)$.

Proof:

$i = n-3$

- $\mathbb{P}(\mathcal{E}_i | \mathcal{E}_1 \cap \dots \cap \mathcal{E}_{i-1}) \geq 1 - \frac{2}{n-i+1} = \frac{n-1-i}{n+1-i}$
- No edge crossing (A, B) contracted: event $\mathcal{E} = \bigcap_{i=1}^{n-2} \mathcal{E}_i$

$$\mathbb{P}(\mathcal{E}) = \mathbb{P}(\mathcal{E}_1) \cdot \mathbb{P}(\mathcal{E}_2 | \mathcal{E}_1) \cdot \mathbb{P}(\mathcal{E}_3 | \mathcal{E}_1 \cap \mathcal{E}_2) \cdot \ldots \cdot \mathbb{P}(\mathcal{E}_{n-2} | \mathcal{E}_1 \cap \dots \cap \mathcal{E}_{n-3})$$

$$\geq \frac{\cancel{n-2}}{n} \cdot \frac{\cancel{n-3}}{n-1} \cdot \frac{\cancel{n-4}}{\cancel{n-2}} \cdot \frac{n-5}{\cancel{n-3}} \cdot \ldots \cdot \frac{\cancel{3}}{\cancel{5}} \cdot \frac{2}{\cancel{4}} \cdot \frac{1}{\cancel{3}} = \frac{2}{n(n-1)} = \frac{1}{\binom{n}{2}}$$

Randomized Min Cut Algorithm

Theorem: If the contraction algorithm is repeated $O(n^2 \log n)$ times, one of the $O(n^2 \log n)$ instances returns a min. cut w.h.p.

Proof:

- Probability to not get a minimum cut in $c \cdot \binom{n}{2} \cdot \ln n$ iterations:

$$\left(1 - \frac{1}{\binom{n}{2}}\right)^{c \cdot \binom{n}{2} \cdot \ln n} < e^{-c \ln n} = \frac{1}{n^c}$$

$$-\frac{1}{\binom{n}{2}} c \binom{n}{2} \ln n$$

Corollary: The contraction algorithm allows to compute a minimum cut in $O(n^4 \log n)$ time w.h.p.

- Each instance can be implemented in $O(n^2)$ time.
  ($O(n)$ time per contraction)

Can We Do Better?

- Time $O(n^4 \log n)$ is not very spectacular, a simple max flow based implementation has time $O(n^4)$.

However, we will see that the contraction algorithm is nevertheless very interesting because:

1. The algorithm can be improved to beat every known deterministic algorithm.

1. It allows to obtain strong statements about the distribution of cuts in graphs.

Better Randomized Algorithm

Recall:

- Consider a fixed min cut (A, B), assume (A, B) has size k
- The algorithm outputs (A, B) iff none of the k edges crossing (A, B) gets contracted.
- Throughout the algorithm, the edge connectivity is at least k and therefore each node has degree $\geq k$
- Before contraction i, there are $n + 1 - i$ nodes and thus at least $(n + 1 - i)k/2$ edges
- If no edge crossing (A, B) is contracted before, the probability to contract an edge crossing (A, B) in step i is at most

$$\frac{k}{\frac{(n+1-i)k}{2}} = \frac{2}{n+1-i}.$$

Improving the Contraction Algorithm

- For a specific min cut (A, B), if (A, B) survives the first i contractions,

$$\mathbb{P}(\text{edge crossing } (A, B) \text{ in contraction } i \cancel{+1}) \leq \frac{2}{n - i + 1}.$$

- **Observation:** The probability only gets large for large i

- **Idea:** The early steps are much safer than the late steps.

  Maybe we can repeat the late steps more often than the early ones.

Safe Contraction Phase

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

Lemma: A given min cut (A, B) of an n-node graph G survives the first $n - \underbrace{\left\lceil {}^{n}/_{\sqrt{2}} + 1 \right\rceil}_{t}$ contractions, with probability $> {}^{1}/_{2}$.

$\left(1 - \frac{1}{\sqrt{2}}\right) \cdot n$

Proof:

$\mathbb{P}(\mathcal{E}_i \mid \mathcal{E}_1 \cap \dots \cap \mathcal{E}_{i-1}) \geq \frac{n-1-i}{n+1-i}$

- Event $\mathcal{E}_i$: cut (A, B) survives contraction i
- Probability that (A, B) survives the first $n - t$ contractions:

$$\mathbb{P}(\mathcal{E}_1) \cdot \mathbb{P}(\mathcal{E}_2 \mid \mathcal{E}_1) \cdot \mathbb{P}(\mathcal{E}_3 \mid \mathcal{E}_1 \cap \mathcal{E}_2) \cdot \ldots \cdot \mathbb{P}(\mathcal{E}_{n-t} \mid \mathcal{E}_1 \cap \dots \cap \mathcal{E}_{n-t-1})$$

$$\geq \frac{n-2}{n} \cdot \frac{n-3}{n-1} \cdot \frac{n-4}{n-2} \cdot \ldots \cdot \frac{t}{t+2} \cdot \frac{t-1}{t+1} = \frac{t(t-1)}{n(n-1)}$$

$$= \frac{\left\lceil \frac{n}{\sqrt{2}} + 1 \right\rceil \cdot \left\lceil \frac{n}{\sqrt{2}} \right\rceil}{n(n-1)} \geq \frac{\left(\frac{n}{\sqrt{2}} + 1\right) \frac{n}{\sqrt{2}}}{n(n-1)} = \underbrace{\frac{\frac{n}{\sqrt{2}} + 1}{n}}_{> \frac{1}{\sqrt{2}}} \cdot \underbrace{\frac{\frac{n}{\sqrt{2}}}{n-1}}_{> \frac{1}{\sqrt{2}}} > \frac{1}{2}$$

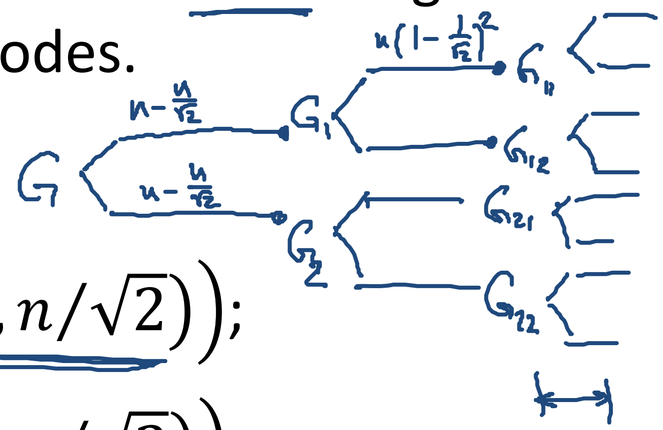
Better Randomized Algorithm

Let's simplify a bit:

- Pretend that $n/\sqrt{2}$ is an integer (for all n we will need it).
- Assume that a given min cut survives the first $n - {}^{n}/_{\sqrt{2}}$ contractions with probability $\geq {}^{1}/_{2}$.

$\mathbf{contract}(\boldsymbol{G}, \boldsymbol{t})$:

- Starting with n-node graph G, perform $n - t$ edge contractions such that the new graph has t nodes.



$\mathbf{mincut}(\boldsymbol{G})$:

1. $X_1 := \text{mincut}\left(\text{contract}(G, n/\sqrt{2})\right);$
2. $X_2 := \text{mincut}\left(\text{contract}(G, n/\sqrt{2})\right);$
3. **return** $\min\{X_1, X_2\};$

Success Probability

mincut(G):

$n - \frac{n}{\sqrt{2}}$

1. $X_1 := \text{mincut}\Big(\text{contract}\big(G, n/\sqrt{2}\big)\Big);$ $\leftarrow$ $1-(1-q)^2$
2. $X_2 := \text{mincut}\Big(\text{contract}\big(G, n/\sqrt{2}\big)\Big);$ $\leftarrow$
3. **return** $\min\{X_1, X_2\}$;

$\boldsymbol{P(n)}$: probability that the above algorithm returns a min cut when applied to a graph with n nodes.

- Probability that X_1 is a min cut $\geq \underbrace{\frac{1}{2}\cdot P\left(\frac{n}{\sqrt{2}}\right)}_{q}$

Recursion:

$$P(n) \geq 1-(1-q)^2 = 1-\left(1-\frac{1}{2}P\left(\frac{n}{\sqrt{2}}\right)\right)^2 = 1-\left(1-P\left(\frac{n}{\sqrt{2}}\right)+\frac{1}{4}\cdot P\left(\frac{n}{\sqrt{2}}\right)^2\right) = \underline{P\left(\frac{n}{\sqrt{2}}\right)-\frac{1}{4}\left[P\left(\frac{n}{\sqrt{2}}\right)\right]^2}$$

Success Probability

$\log\left(\frac{n}{\sqrt{2}}\right) = \log n - \frac{1}{2}$

Theorem: The recursive randomized min cut algorithm returns a minimum cut with probability at least $1/\log_2 n$.

Proof (by induction on n):

$$\longrightarrow \quad P(n) \geq \underbrace{P\left(\frac{n}{\sqrt{2}}\right)} - \frac{1}{4} \cdot \underbrace{P\left(\frac{n}{\sqrt{2}}\right)}^2, \qquad P(2) = 1$$

Base $(n=2)$

$$P(2) \geq \frac{1}{\log_2 2} = 1 \checkmark$$

$$P(n) \geq \frac{1}{\log n/\sqrt{2}} - \frac{1}{4} \frac{1}{\log^2(n/\sqrt{2})}$$

$$= \frac{1}{\log n - \frac{1}{2}} - \frac{1}{4\left(\log n - \frac{1}{2}\right)^2} = \frac{4\left(\log n - \frac{1}{2}\right) - 1}{4\left(\log n - \frac{1}{2}\right)^2} = \frac{4\log n - 3}{4\log^2 n - 4\log n + 1}$$

$$= \frac{4\log n - 4 + \frac{1}{\log n} + 1 - \frac{1}{\log n}}{4\log^2 n - 4\log n + 1} = \frac{1}{\log n} + \underbrace{\frac{1 - \frac{1}{\log n}}{4\log^2 n - \ldots}}_{>0} > \frac{1}{\log_2 n}$$

Running Time

1. $X_1 := \text{mincut}\left(\text{contract}\left(G, n/\sqrt{2}\right)\right)$;
2. $X_2 := \text{mincut}\left(\text{contract}\left(G, n/\sqrt{2}\right)\right)$;
3. **return** $\min\{X_1, X_2\}$;

Recursion:

- $T(n)$: time to apply algorithm to n-node graphs
- Recursive calls: $2T\left({}^{n}/_{\sqrt{2}}\right)$
- Number of contractions to get to ${}^{n}/_{\sqrt{2}}$ nodes: $O(n)$

$$T(n) = 2T\left(\frac{n}{\sqrt{2}}\right) + O(n^2), \qquad T(2) = O(1)$$

Running Time

Theorem: The running time of the recursive, randomized min cut algorithm is $O(n^2 \log n)$.

Proof:

- Can be shown in the usual way, by induction on n

Remark:

- The running time is only by an $O(\log n)$-factor slower than the basic contraction algorithm.
- The success probability is exponentially better!

#repetitions $O(\log^2 n)$ for high probability

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

Number of Minimum Cuts

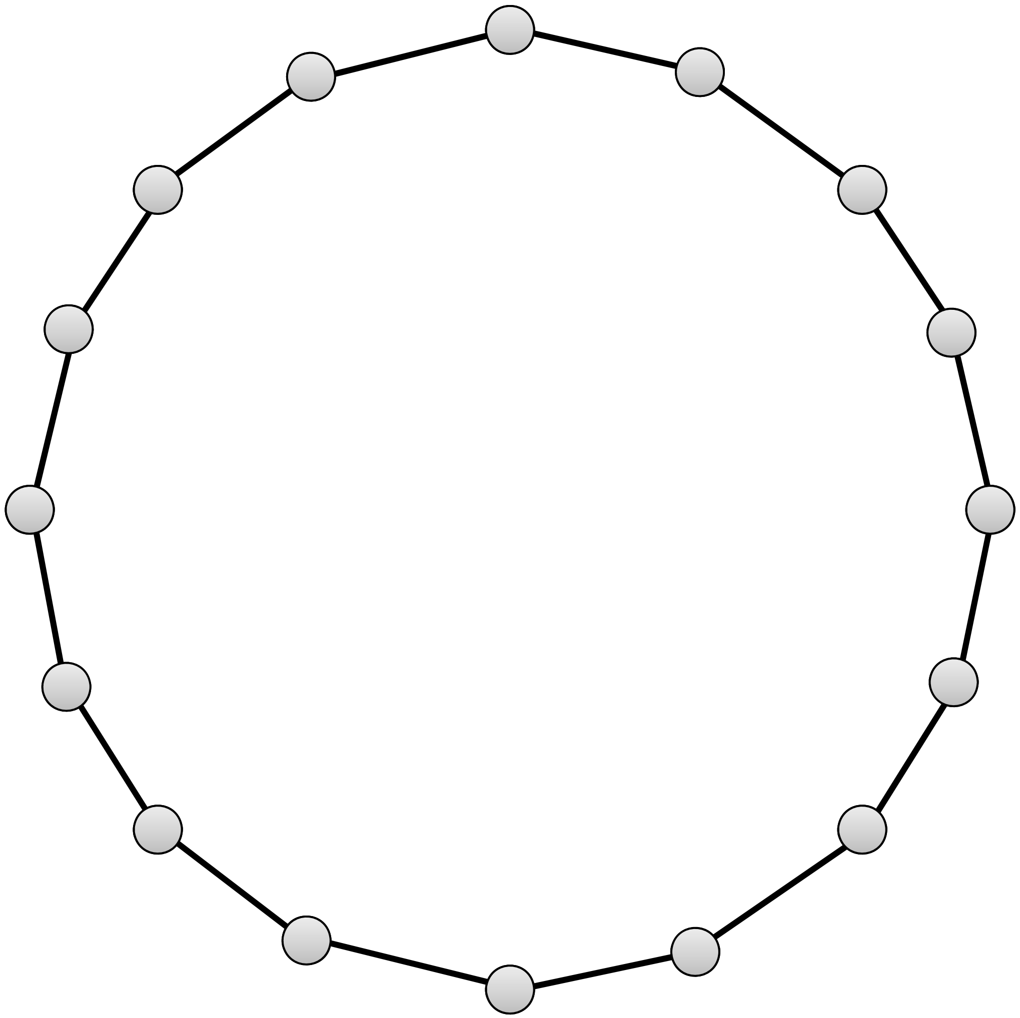
- Given a graph G, how many minimum cuts can there be?

- Or alternatively: If G has edge connectivity k, how many ways are there to remove k edges to disconnect G?

- Note that the total number of cuts is large.

Number of Minimum Cuts

Example: Ring with n nodes



- Minimum cut size: 2
- Every two edges induce a min cut
- Number of edge pairs:

$$\binom{n}{2}$$

- Are there graphs with more min cuts?

Number of Min Cuts

Theorem: The number of minimum cuts of a graph is at most $\binom{n}{2}$.

Proof:

$C_i \cap C_j = \emptyset$

- Assume there are s min cuts
- For $i \in \{1, \dots, s\}$, define event $\mathcal{C}_i$:

  $\mathcal{C}_i := \{\text{basic contraction algorithm returns min cut } i\}$

- We know that for $i \in \{1, \dots, s\}$: $\mathbb{P}(\mathcal{C}_i) \geq 1/\binom{n}{2}$

$A \quad B$

$\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B)$

- Events $\mathcal{C}_1, \dots, \mathcal{C}_s$ are disjoint:

$$1 \geq \mathbb{P}\left(\bigcup_{i=1}^{s} \mathcal{C}_i\right) = \sum_{i=1}^{s} \mathbb{P}(\mathcal{C}_i) \geq \frac{s}{\binom{n}{2}}$$

$$\Longrightarrow s \leq \binom{n}{2}$$