# Chapter 7
# Randomization

## Algorithm Theory
## WS 2015/16

**Fabian Kuhn**

# 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 **specific** minimum cut with probability at least $\frac{2}{n(n-1)}$.

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

- There are $n-2$ contractions, each can be done in time $O(n)$.

# 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.

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

- One instance consists of $n - 2$ edge contractions
- Each edge contraction can be carried out in time $O(n)$
  - Actually: $O(\text{current \#nodes})$
- Time per instance of the contraction algorithm: $O(n^2)$

# 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.

# Improving the Contraction Algorithm

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

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

- **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

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

**Proof:**

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

# 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. $X_1 \coloneqq \mathrm{mincut}\left(\mathrm{contract}\left(G, n/\sqrt{2}\right)\right)$;
2. $X_2 \coloneqq \mathrm{mincut}\left(\mathrm{contract}\left(G, n/\sqrt{2}\right)\right)$;
3. **return** $\min\{X_1, X_2\}$;

# Success Probability

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

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** $\min\{X_1, X_2\};$

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

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

# Running Time

1. $X_1 \coloneqq \mathrm{mincut}\left(\mathrm{contract}\left(G, n/\sqrt{2}\right)\right);$
2. $X_2 \coloneqq \mathrm{mincut}\left(\mathrm{contract}\left(G, n/\sqrt{2}\right)\right);$
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:**

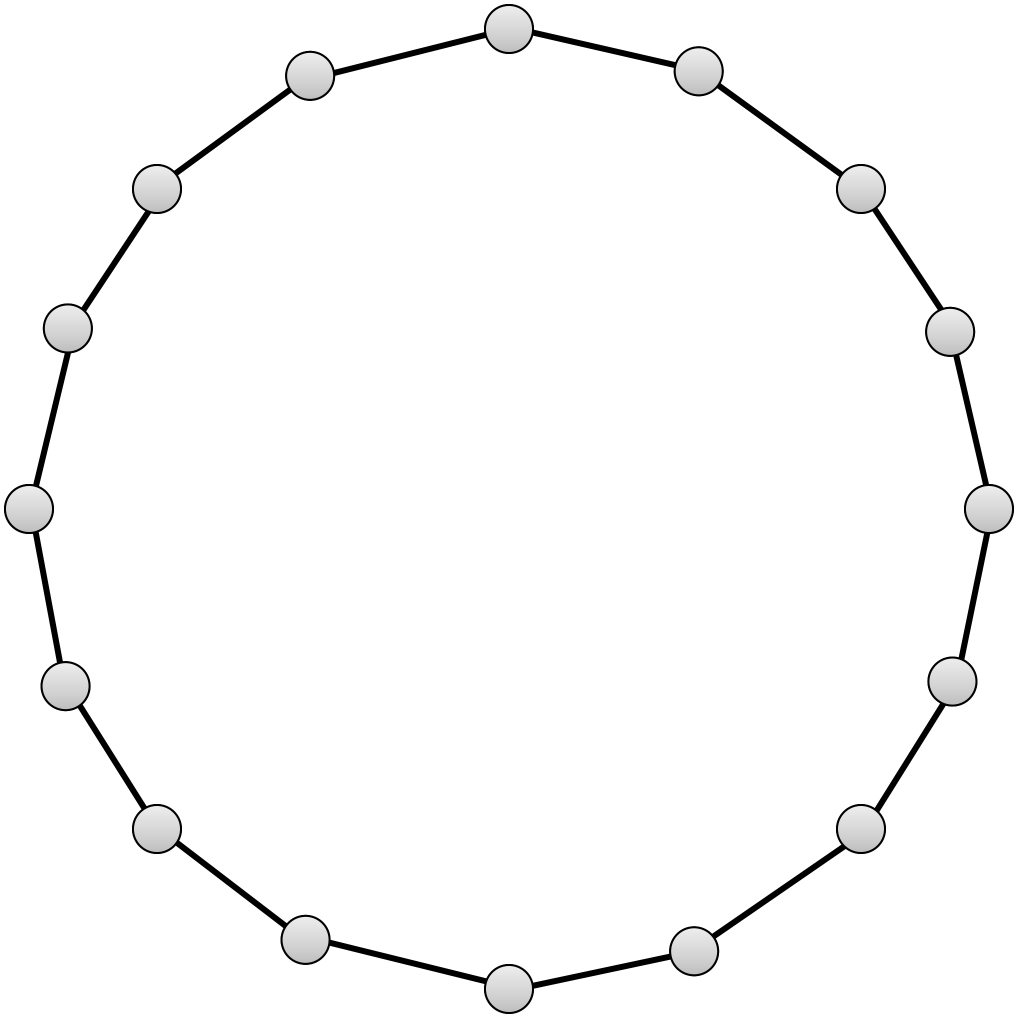
- The running time is only by an $O(\log n)$-factor slower than the basic contraction algorithm.
- The success probability is exponentially better!

# 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:**

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

  $\mathcal{C}_i \coloneqq \{\text{basic contraction algorithm returns min cut } i\}$
- We know that for $i \in \{1, \dots, s\}$: $\mathbb{P}(\mathcal{C}_i) \geq 1/\binom{n}{2}$
- Events $\mathcal{C}_1, \dots, \mathcal{C}_s$ are disjoint:

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