Chapter 7
Randomization

Algorithm Theory
WS 2019/20

Fabian Kuhn

Minimum Cut – Randomized Contraction Alg.

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

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

Basic randomized contraction algorithm:

while there are >2 nodes **do**

contract a uniformly random edge

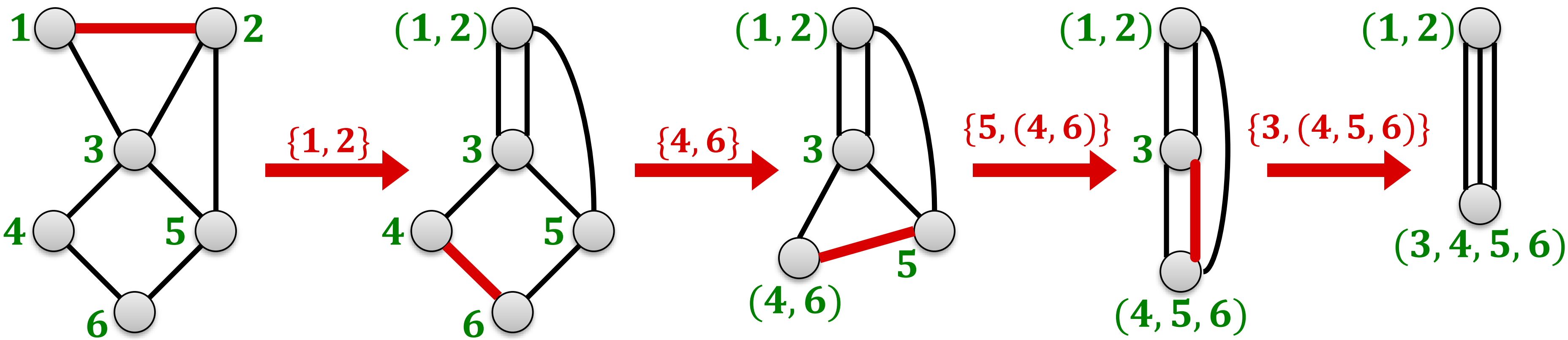
return cut induced by the last two remaining nodes

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

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 $(S_w, V \setminus S_w)$

Getting The Min Cut

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 $m = {}^{1}/{}_{2} \cdot \sum_v \deg(v)$

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

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

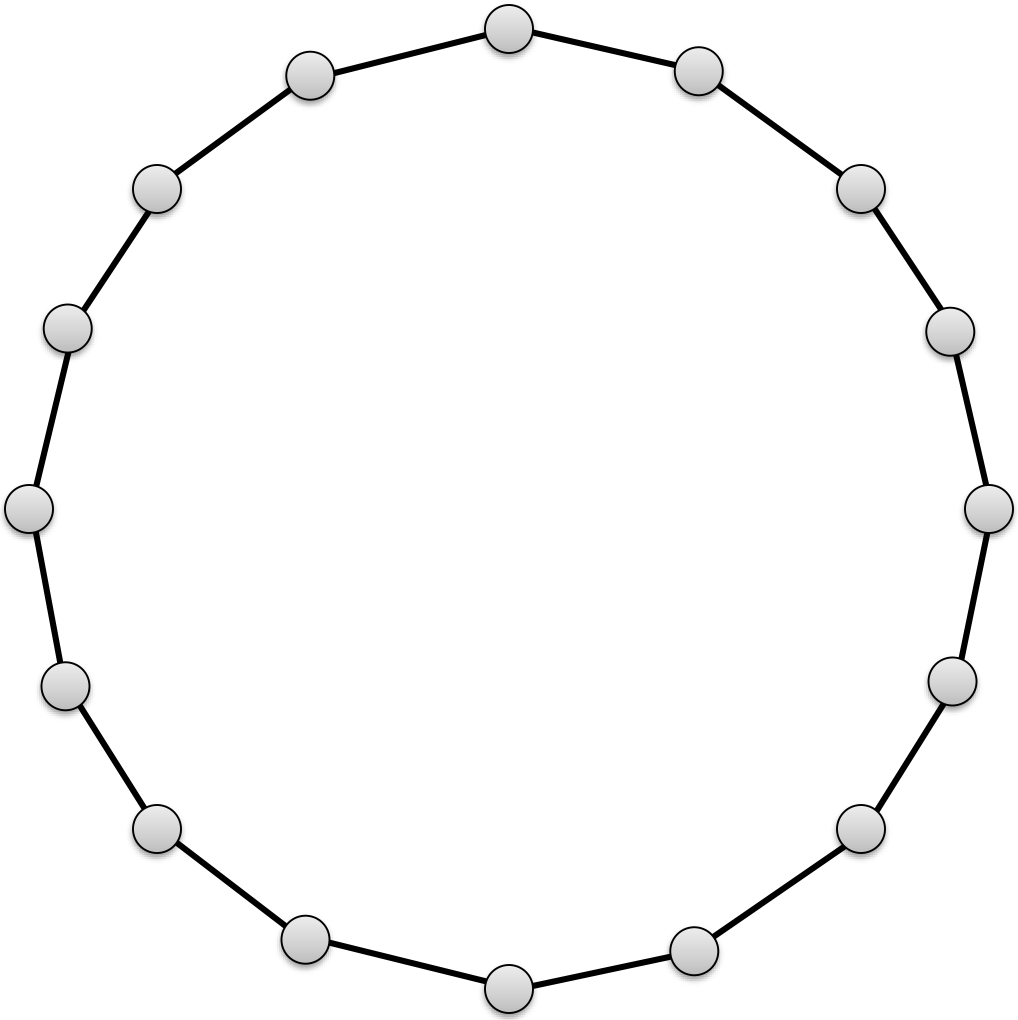
- Given a graph G, how many minimum cuts can there be?

- Or alternatively: If G has edge connectivity $k \geq 1$, 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 $i \in \{1, \dots, s\}$: $\mathbb{P}(\mathcal{C}_i) = 1/\binom{n}{2}$
- Events $\mathcal{C}_1, \dots, \mathcal{C}_s$ are disjoint:

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

Counting Larger Cuts

- In the following, assume that min cut has size $\lambda = \lambda(G)$
- How many cuts of size $\leq k = \alpha \cdot \lambda$ can a graph have?
- Consider a specific cut (A, B) of size $\leq k$
- As before, during the contraction algorithm:
  - min cut size $\geq \lambda$
  - total number of edges $\geq \lambda \cdot \#\text{nodes}/2$
  - cut (A, B) remains as long as none of its edges gets contracted
- Prob. that an edge crossing (A, B) is chosen in i^{th} contraction

$$= \frac{k}{\#\text{edges}} \leq \frac{2k}{\lambda \cdot \#\text{nodes}} = \frac{2\alpha}{n - i + 1}$$

For simplicity, in the following, assume that 2α is an integer

Counting Larger Cuts

Lemma: If $2\alpha \in \mathbb{N}$, the probability that cut (A, B) of size $\alpha \cdot \lambda$ survives the first $n - 2\alpha$ edge contractions is at least

$$\frac{(2\alpha)!}{n(n-1) \cdot \ldots \cdot (n - 2\alpha + 1)} \geq \frac{2^{2\alpha - 1}}{n^{2\alpha}}.$$

Proof:

- As before, event $\mathcal{E}_i$: cut (A, B) survives contraction i

Number of Cuts

Theorem: If $2\alpha \in \mathbb{N}$, the number of edge cuts of size at most $\alpha \cdot \lambda(G)$ in an n-node graph G is at most $n^{2\alpha}$.

Proof:

Remark: The bound also holds for general $\alpha \geq 1$.

Resilience To Edge Failures

- Consider a network (a graph) G with n nodes
- Assume that each link (edge) of G fails independently with probability p
- How large can p be such that the remaining graph is still connected with high probability or with probability $1 - \varepsilon$?

Maintaining Connectivity

- A graph $G = (V, E)$ is connected iff every edge cut (A, B) has size at least 1.

- We need to make sure that every cut keeps at least 1 edge

Resilience to Edge Failures

- Consider an edge cut (A, B) of size $k = \alpha \cdot \lambda(G)$
- Assume that each edge fails with probability $p \leq 1 - \frac{c \cdot \ln n}{\lambda(G)}$
- Hence each edge survives with probability $q \geq \frac{c \cdot \ln n}{\lambda(G)}$
- Probability that no edge crossing (A, B) survives

Maintaining All Cuts of a Certain Size

- The number of cuts of size $k = \alpha\lambda(G)$ is at most $n^{2\alpha}$.

Claim: If each edge survives with probability $q \geq \frac{c \cdot \ln(n)}{\lambda(G)}$, with probability at least $1 - n^{-2\alpha}$, for a given $k = \alpha\lambda(G)$, at least one edge of each cut of size k survives.

Maintaining Connectivity

Theorem: If each edge of a graph G independently survives with probability at least $\frac{(c+4)\cdot \ln n}{\lambda(G)}$, the remaining graph is connected with probability at least $1 - \frac{1}{n^c}$.