



Chapter 6

Randomization

Algorithm Theory
WS 2014/15

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

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

Maximum-flow based algorithm:

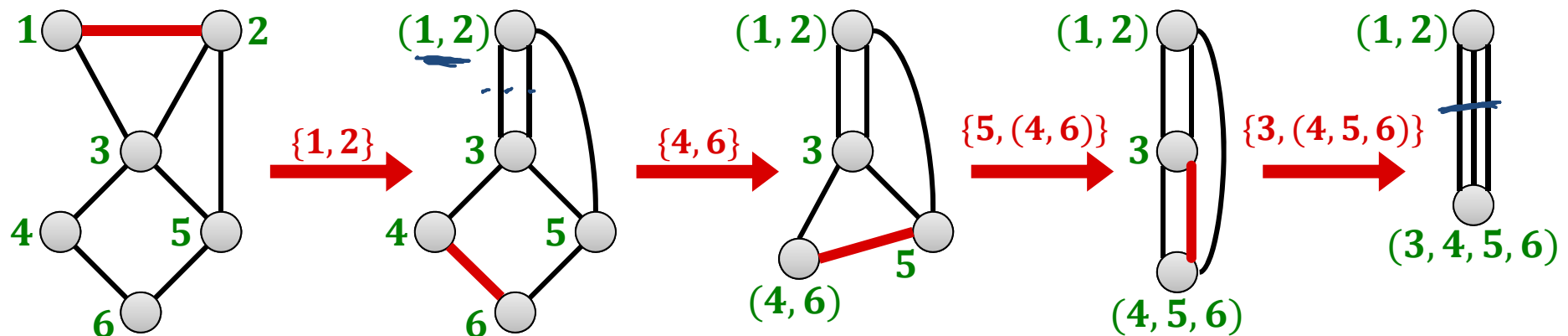
- Fix s , compute min s - t -cut for all $t \neq s$
- $O(m \cdot \lambda(G)) = O(mn)$ per s - t cut
- Gives an $O(mn\lambda(G)) = O(mn^2)$ -algorithm

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

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 $\frac{2}{n(n-1)}$.

- 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.

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 \rightarrow **output is (A, B)**

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:

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

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)

Improving the Contraction Algorithm

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

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

- **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 $n - \left\lfloor \frac{n}{\sqrt{2}} + 1 \right\rfloor$ contractions, with probability $\underline{\underline{> 1/2}}$.

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:

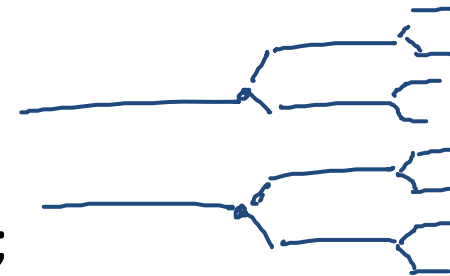
- Pretend that $\underline{n/\sqrt{2}}$ is an integer (for all n we will need it).
- Assume that a given min cut survives the first $n - n/\sqrt{2}$ contractions with probability $\geq 1/2$.

contract(G, t):

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

mincut(G):

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



Success Probability

mincut(G):

preserve min cut with prob. $\geq \frac{1}{2}$

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

$P(n)$: 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 $\geq \frac{1}{2} \cdot P(\frac{n}{\sqrt{2}})$

Recursion:

$$1 - \left(1 - \frac{1}{2} P(\frac{n}{\sqrt{2}})\right)^2 = P(\frac{n}{\sqrt{2}}) - \frac{1}{4} P(\frac{n}{\sqrt{2}})^2$$

Success Probability

$$P(n) \geq \frac{1}{\log_2 n}$$



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

Proof (by induction on n):

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

Base case: $n=2$ ✓

Ind. Step: $P(n) = P\left(\frac{n}{\sqrt{2}}\right) - \frac{1}{4} P\left(\frac{n}{\sqrt{2}}\right)^2$

$$\stackrel{\text{i.H.}}{\geq} \frac{1}{\log_2\left(\frac{n}{\sqrt{2}}\right)} - \frac{1}{4} \frac{1}{\log_2\left(\frac{n}{\sqrt{2}}\right)^2} = \frac{1}{\log_2\left(\frac{n}{\sqrt{2}}\right)} \left(1 - \frac{1}{4 \log_2\left(\frac{n}{\sqrt{2}}\right)}\right)$$

$$= \frac{1}{\log_2 n - \frac{1}{2}} \left(\frac{4 \log_2 n - 3}{4 \log_2 n - 2} \right) = \frac{4 \log_2 n - 3}{4 \log_2^2 n - 4 \log_2 n + 1}$$

Running Time

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

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

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + n^c$$

Recursion:

$$\text{if } c = \log_b a: \quad n^{\log_b a} \cdot \log n$$

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

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

$$\Rightarrow T(n) = O(n^2 \log n)$$

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 (Master Thm.)

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 to get a high prob. bound: $O(\log^2 n)$

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

best. det. alg.: $O(m \cdot n + n^2 \log n)$

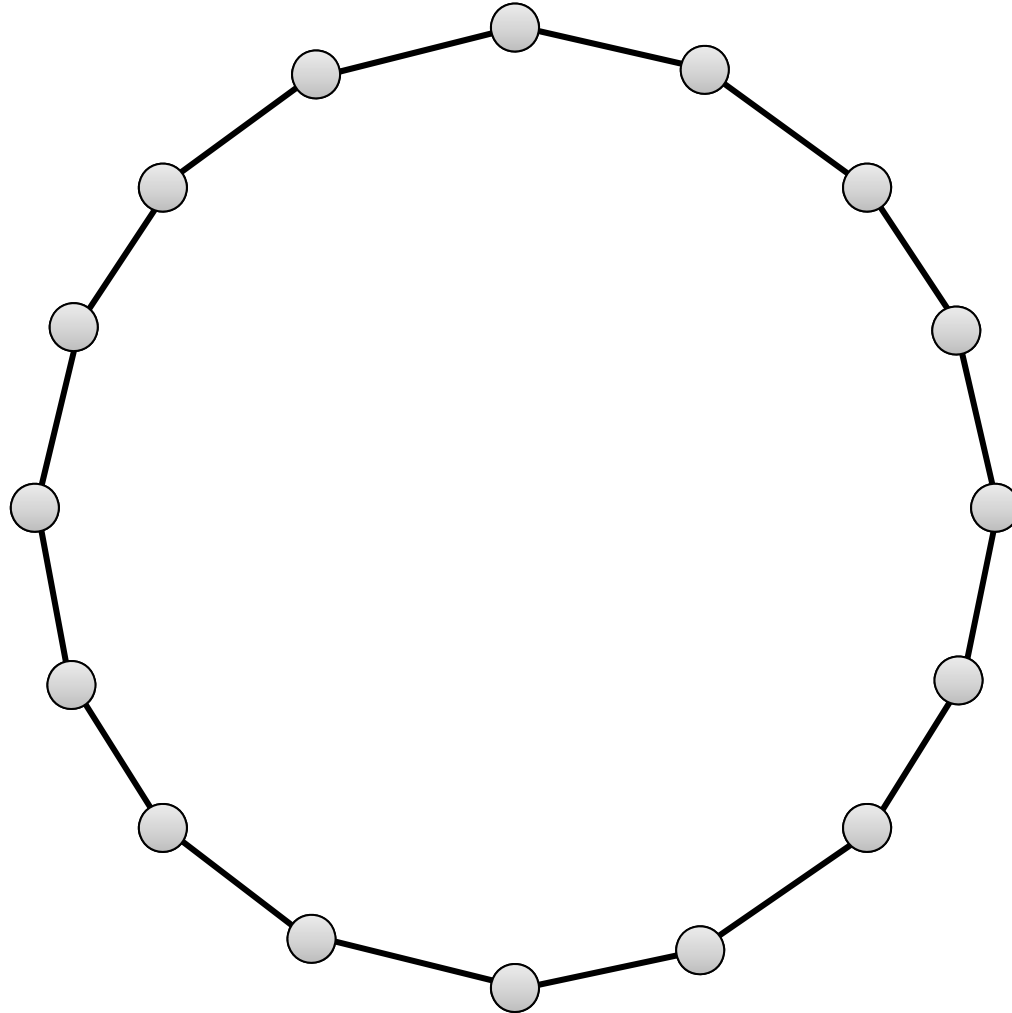
Number of Minimum Cuts

- 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.

$$\frac{1}{2}(2^n - 2) = \underline{\underline{2^{n-1} - 1}}$$

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 :

Can be generalized
 # of cuts of size $\leq \alpha \cdot \chi(G)$
 $O(n^{2\alpha})$

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

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

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

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

