Algorithm Theory

Chapter 7
Randomized Algorithms

Part V:
Basic Randomized Minimum Cut Algorithm

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 $(\boldsymbol{A}, \boldsymbol{B})$: # of edges crossing the cut

- For weighted graphs, total edge weight crossing the cut

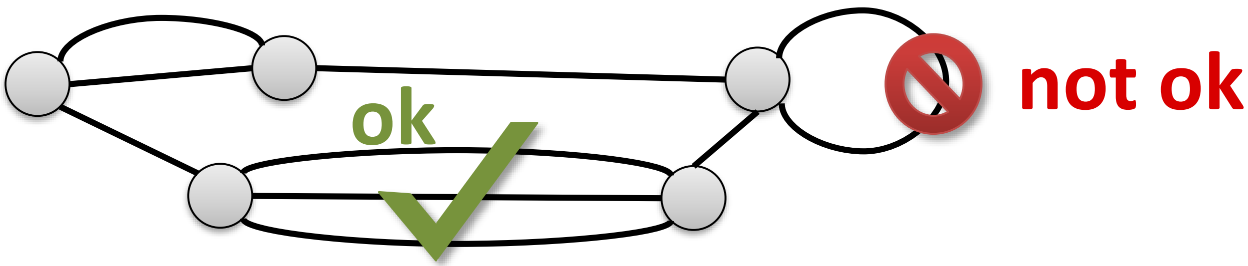
Goal: Find a cut of minimal size (i.e., of size $\lambda(G)$)

Maximum-flow based algorithm:

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

Edge Contractions

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



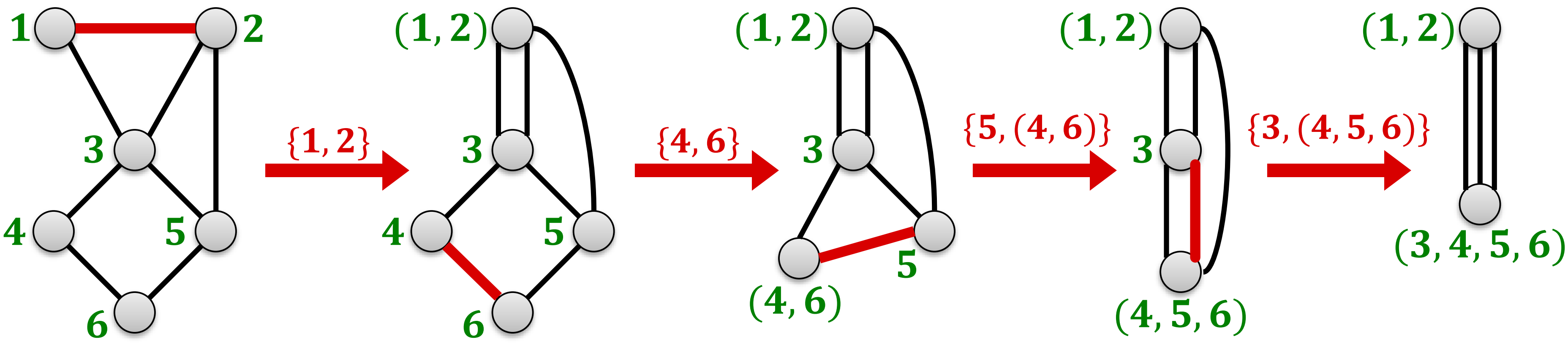
Contracting edge $\{u, v\}$:

- 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 $O(n)$.
- We will see this later.

Contractions and Cuts

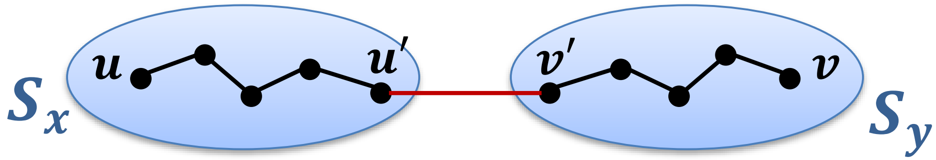
Lemma: If two original nodes $u, v \in V$ are merged into the same node of the contracted graph, there is a path connecting u and v in the original graph s.t. all edges on the path are contracted.

Proof:

- Any edge $\{x, y\}$ in the contracted graph corresponds to some edge in the original graph between two nodes u' and v' in the sets S_x and S_y represented by x and y.

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

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



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 $u \in A$, $v \in B$ 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)

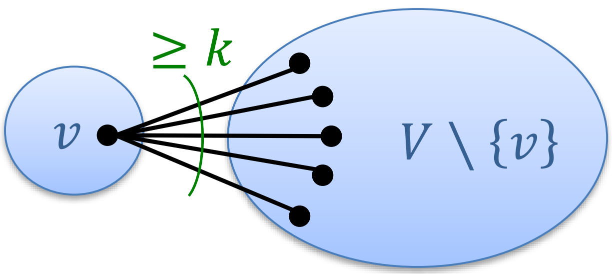
   In the end there are only 2 sets → output is (A, B)

Getting The Min Cut

Theorem: The probability that the algorithm outputs a specific minimum cut is at least $2/n(n-1) = 1/\binom{n}{2}$.

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.

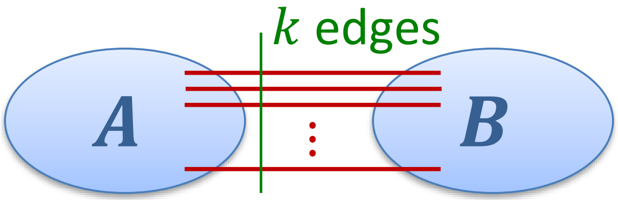


Proof:

- 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 $m = {}^1/_2 \cdot \sum_v \deg(v) \geq {}^1/_2 \cdot nk$

Getting The Min Cut

Theorem: The probability that the algorithm outputs a specific 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 specific 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)
  - Goal: show that $\mathbb{P}(\mathcal{E}_1 \cap \dots \cap \mathcal{E}_{n-2}) \geq 2/n(n-1)$.

$$\begin{aligned}
&\mathbb{P}\big(\text{alg. returns } (A, B)\big) \\
&\quad = \mathbb{P}(\mathcal{E}_1 \cap \mathcal{E}_2 \cap \dots \cap \mathcal{E}_{n-2}) \\
&\quad = \mathbb{P}(\mathcal{E}_1) \cdot \mathbb{P}(\mathcal{E}_2 \mid \mathcal{E}_1) \cdot \mathbb{P}(\mathcal{E}_3 \mid \mathcal{E}_1 \cap \mathcal{E}_2) \cdot \dots \cdot \mathbb{P}(\mathcal{E}_{n-2} \mid \mathcal{E}_1 \cap \mathcal{E}_2 \cap \dots \cap \mathcal{E}_{n-3})
\end{aligned}$$

$$\mathbb{P}(\mathcal{E}_i \mid \mathcal{E}_1 \cap \dots \cap \mathcal{E}_{i-1}) \geq 1 - \frac{2}{n+1-i} = \frac{n-i-1}{n-i+1}$$

Getting The Min Cut

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

Proof:

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

$$
\begin{aligned}
\mathbb{P}(\mathcal{E}) &= \mathbb{P}(\mathcal{E}_1 \cap \dots \cap \mathcal{E}_{n-2}) \\
&= \mathbb{P}(\mathcal{E}_1) \cdot \mathbb{P}(\mathcal{E}_2 \mid \mathcal{E}_1) \cdot \dots \cdot \mathbb{P}(\mathcal{E}_{n-2} \mid \mathcal{E}_1 \cap \dots \cap \mathcal{E}_{n-3}) \\
&\geq \frac{n-2}{n} \cdot \frac{n-3}{n-1} \cdot \frac{n-4}{n-2} \cdot \frac{n-5}{n-3} \cdot \frac{n-6}{n-4} \cdot \dots \cdot \frac{4}{6} \cdot \frac{3}{5} \cdot \frac{2}{4} \cdot \frac{1}{3} \\
&= \frac{2}{n(n-1)} = \frac{1}{\binom{n}{2}}
\end{aligned}
$$

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:

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

$\forall x \in \mathbb{R} : (1 + x) \le e^x$

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

- It remains to show that each instance can be implemented in $O(n^2)$ time.