Chapter 7
Randomization

Algorithm Theory
WS 2015/16

Fabian Kuhn

Randomized Contraction Algorithm

Algorithm: to compute a min. cut in a graph

while there are > 2 nodes **do**

$O(n^2)$ 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 $O(n)$.

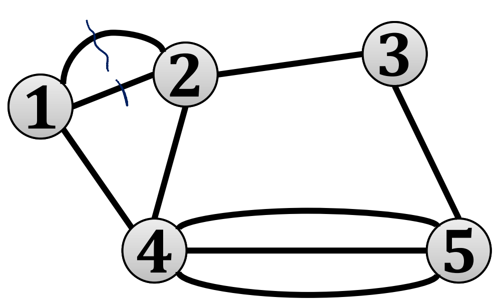
Implementing Edge Contractions

Edge Contraction:

- Given: multigraph with n nodes
  - assume that set of nodes is $\{1, \dots, n\}$
- Goal: contract edge $\{u, v\}$

Data Structure

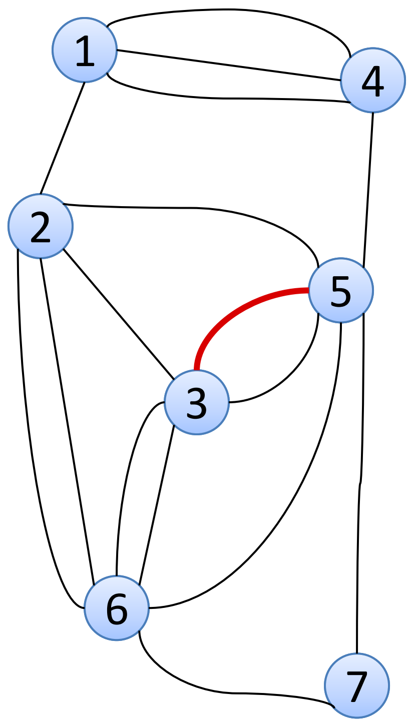
- We can use either adjacency lists or an adjacency matrix
- Entry in row i and column j: #edges between nodes i and j
- Example:



$$A = \begin{array}{cc} & \begin{matrix} & 2 & & & \end{matrix} \\ \begin{matrix} 1 \\ \\ \\ \\ \end{matrix} & \begin{pmatrix} 0 & 2 & 0 & 1 & 0 \\ 2 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 3 \\ 0 & 0 & 1 & 3 & 0 \end{pmatrix} \end{array}$$

Contracting An Edge

Example: Contract one of the edges between 3 and 5

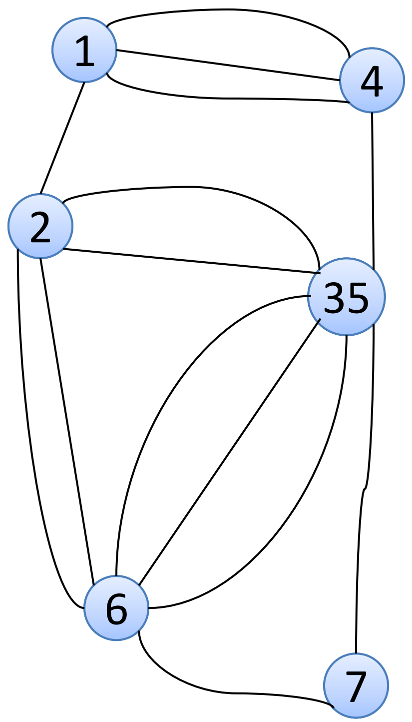


	1	2	3	4	5	6	7
1	0	1	0	3	0	0	0
2	1	0	1	0	1	2	0
3	0	1	0	0	2	2	0
4	3	0	0	0	1	0	0
5	0	1	2	1	0	1	1
6	0	2	2	0	1	0	1
7	0	0	0	0	1	1	0

{3,5}	0	2	~~0~~	1		3	1

Contracting An Edge

Example: Contract one of the edges between 3 and 5

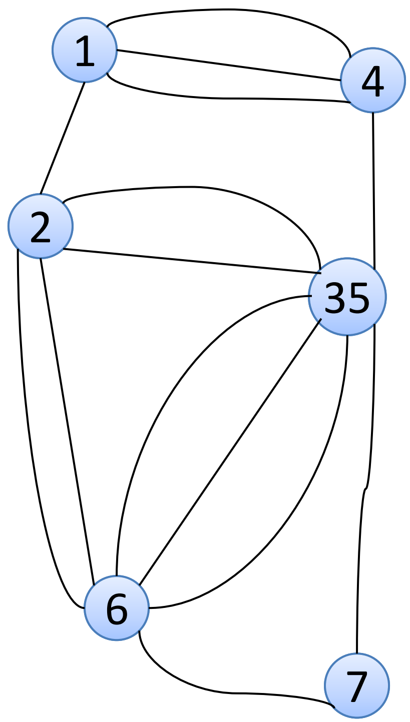


	1	2	3	4	5	6	7
1	0	1	0	3	0	0	0
2	1	0	1	0	1	2	0
3	0	1	0	0	2	2	0
4	3	0	0	0	1	0	0
5	0	1	2	1	0	1	1
6	0	2	2	0	1	0	1
7	0	0	0	0	1	1	0

{3,5}	0	2		1		3	1

Contracting An Edge

Example: Contract one of the edges between 3 and 5



	1	2	35	4		6	7
1	0	1	0	3		0	0
2	1	0	2	0		2	0
35	0	2	0	1		3	1
4	3	0	1	0		0	0
						0	
6	0	2	3	0		0	1
7	0	0	1	0		1	0

{3,5}	0	2		1		3	1

Contracting an Edge

Claim: Given the adjacency matrix of an n-node multigraph and an edge $\{u, v\}$, one can contract the edge $\{u, v\}$ in time $O(n)$.

- Row/column of combined node $\{u, v\}$ is sum of rows/columns of u and v

- Row/column of u can be replaced by new row/column of combined node $\{u, v\}$

- Swap row/column of v with last row/column in order to have the new $(n-1)$-node multigraph as a contiguous $(n-1) \times (n-1)$ submatrix

Finding a Random Edge

- We need to contract a uniformly random edge
- How to find a uniformly random edge in a multigraph?
  - Finding a random non-zero entry (with the right probability) in an adjacency matrix costs $O(n^2)$.

Idea for more efficient algorithm:

- First choose a random node u
  - with probability proportional to the degree (#edges) of u
- Pick a random edge of u
  - only need to look at one row → time $O(n)$

| 1 | 2 | 0 | 3 | 0 | 0 | 2 | - - - total #edges : d

$$\frac{1}{d}, \quad \left(1-\frac{1}{d}\right)\cdot\frac{2}{d-1} = \frac{2}{d}, \quad \left(1-\frac{3}{d}\right)\frac{3}{d-3} = \frac{3}{d}$$

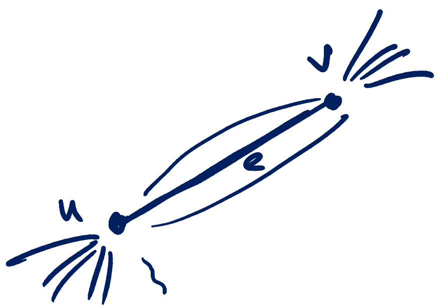
Choose a Random Node

Edge Sampling:

1. Choose a node $u \in V$ with probability

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

2. Choose a uniformly random edge of u $\leftarrow$ time $O(n)$



$$\mathbb{P}(\text{pick } e) = \frac{\deg(u)}{2m}\cdot\frac{1}{\deg(u)} + \frac{\deg(v)}{2m}\cdot\frac{1}{\deg(v)} = \frac{1}{2m} + \frac{1}{2m} = \frac{1}{m}$$

Choose a Random Node

- We need to choose a random node u with probability $\frac{\deg(u)}{2m}$
- Keep track of the number of edges m and maintain an array with the degrees of all the nodes
  - Can be done with essentially no extra cost when doing edge contractions

Choose a random node:

```
degsum = 0;
for all nodes u ∈ V:
    with probability deg(u)/(2m−degsum):
        pick node u; terminate
    else
        degsum += deg(u)
```

$$\frac{\deg(u)}{2m - \text{degsum}}$$

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.

2. 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

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

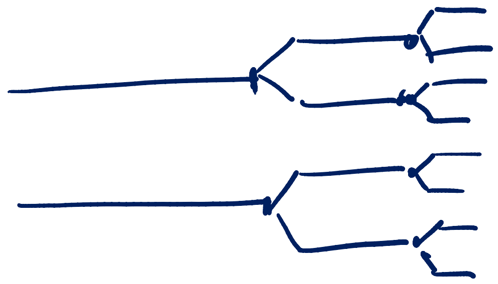
Improving the Contraction Algorithm

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

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

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

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:

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

$$= \frac{t}{n} \cdot \frac{t-1}{n-1}$$

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

$$\geq \frac{n}{\sqrt{2}} + 1$$

$$\geq \frac{{}^{n}/_{\sqrt{2}} + 1}{n} \cdot \frac{{}^{n}/_{\sqrt{2}}}{n-1} > \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} = \frac{1}{2}$$

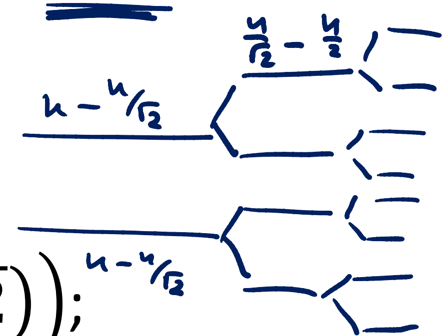
Better Randomized Algorithm

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 $\geq {}^{1}/_{2}$.

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

Success Probability

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

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

Recursion:

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

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 $1/\log_2 n$.

Proof (by induction on n):

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

Base case: $n = 2$ ✓

$$x - \frac{x^2}{4}$$

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

$$\stackrel{\text{I.H.}}{\geq} \frac{1}{\log(n/\sqrt{2})} - \frac{1}{4} \frac{1}{\log(n/\sqrt{2})^2} = \frac{1}{\log(n/\sqrt{2})}\left(1 - \frac{1}{4} \cdot \frac{1}{\log(n/\sqrt{2})}\right)$$

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

$$= \frac{4\log n - 3}{4\log^2 n - 4\log n + 1} \geq \frac{1}{\log n}$$

Running Time

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

Master thm:

$c = \log_b a$

$T(n) = a \cdot T(n/b) + O(n^c)$ $\hookrightarrow O(n^c \cdot \log n)$

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: $O(n)$

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

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

Running Time

$1-\varepsilon < e^{-\varepsilon}$

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

Proof:

(Master Thm)

$t = \Theta(\log^2 n)$

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

$$\left(1-\frac{1}{\log n}\right)^t < e^{-t/\log n} \overset{!}{=} \frac{1}{n^c} = e^{-c\ln n}$$

Remark:

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

succ. prob. $\frac{1}{\log n}$

If we want a min. cut w.h.p. $\left(1-\frac{1}{n^c}\right)$: we need $O(\log^2 n)$ rep.

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

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

Number of Minimum Cuts

G

$S \quad V \setminus S$

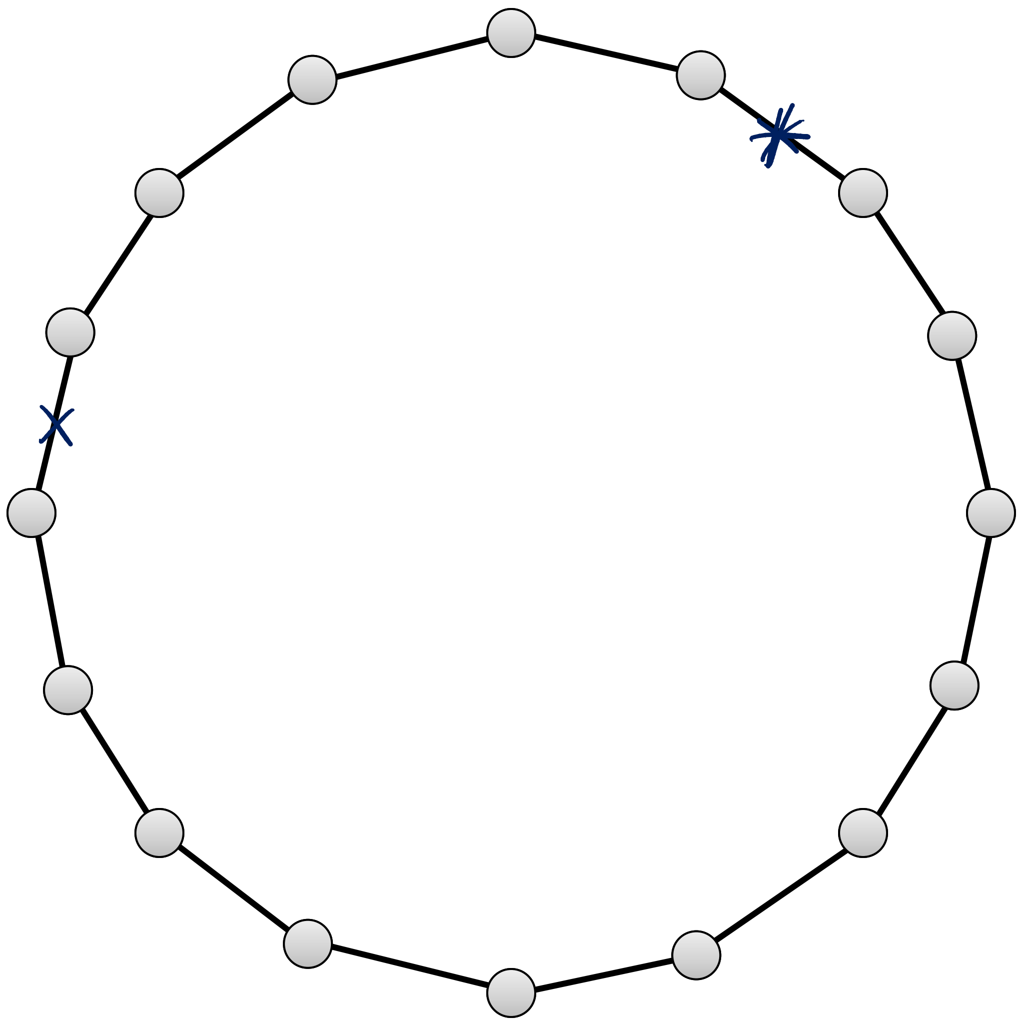
$\#\text{edges} = \lambda(G)$

- 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.

$2^{n-1} - 2$

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

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

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

can be generalized

#cuts of size $\leq \alpha \cdot \lambda(G)$

is at most $n^{2\alpha}$