Algorithms, Probability, and Computing

Angelika Steger
Institut für Theoretische Informatik
steger@inf.ethz.ch
**Borůvka’s Algorithm** \((G):\)

\[
\forall v \in V: \text{compute } e_{\min}(v), \text{ insert } e_{\min}(v) \text{ in the MST and contract } e_{\min}(v) \\
\text{(removing loops and double edges by keeping only the cheapest edge)} \\
\text{recurse (until the graph contains only one vertex)}
\]

**Run time analysis:**

- #vertices \( n \)  \( \leq n/2 \)  \( \leq n/4 \)  \( \leq n/8 \)  ...  \( O(m \log n) \)
- #edges \( m \)  \( \leq m \)  \( \leq m \)  \( \leq m \)  ...  \( O(m) \)

If we could show:

\( \leq m/2 \)  \( \leq m/4 \)  \( \leq m/8 \)  ...  \( O(m) \)
**Randomized MST**

**Randomized Minimum Spanning Tree Algorithm (G):**

Perform three iterations of Borůvka’s algorithm (which reduces the number of vertices to at most $n/8$)

In the new graph select edges with probability $1/2$ and compute recursively a MSF for the graph consisting of the selected edges. Call this forest $T$.

Use `FINDHEAVY` to find all unselected edges that are not $T$-heavy.

Add all edges that are *not* $T$-heavy to $T$ and delete all other edges.

Recursively (until the graph contains only one vertex)

**Assumptions:**

- Run time of `FindHeavy` $\leq C_{FH}(n+m)$ (for a graph with $n$ vertices and $m$ edges)
- Run time of three Boruvka steps $\leq C_B(n+m)$ (for a graph with $n$ vertices and $m$ edges)

**Claim:**

Run time of Randomized MST $\leq C(n+m)$ (for a graph with $n$ vertices and $m$ edges)
**Randomized Minimum Spanning Tree Algorithm (G):**

Perform three iterations of Borůvka’s algorithm (which reduces the number of vertices to at most \( n/8 \))

In the new graph select edges with probability 1/2 and compute recursively a MSF for the graph consisting of the selected edges. Call this forest \( T \).

Use **FINDHEAVY** to find all unselected edges that are not \( T \)-heavy.

Add all edges that are *not* \( T \)-heavy to \( T \) and delete all other edges.

**Assumptions:**

- Run time of **FindHeavy** \( \leq C_{FH}(n+m) \) (for a graph with \( n \) vertices and \( m \) edges)
- Run time of three Borůvka steps \( \leq C_B(n+m) \) (for a graph with \( n \) vertices and \( m \) edges)

**Claim:** there exists \( C \) s.t.

- Run time of Randomized MST \( \leq C(n+m) \) (for a graph with \( n \) vertices and \( m \) edges)
BasicMinCut(G):
  while G has more than 2 vertices do
    pick a random edge e in G
    G ← G/e
  end while
  return the size of the only cut in G

run time:
  each iteration: $O(n)$
  total: $O(n^2)$
Correctness Analysis

Observation 1.1. Let $G$ be a multigraph and $e$ an edge of $G$. Then $\mu(G/e) \geq \mu(G)$. Moreover, if there exists a minimum cut $C$ in $G$ such that $e \notin C$, then $\mu(G/e) = \mu(G)$.

Lemma 1.2. Let $G$ be a multigraph with $n$ vertices. Then the probability of $\mu(G) = \mu(G/e)$ for a randomly chosen edge $e \in E(G)$ is at least $1 - \frac{2}{n}$.

Thus:

$$\Pr[ \text{BasicMinCut}(G) \text{ returns size of min cut } ] \geq \frac{n - 2}{n} \cdot \frac{n - 3}{n - 1} \cdot \frac{n - 4}{n - 2} \cdot \frac{2}{4} \cdot \frac{1}{3} \cdot p_0(2) = \frac{2}{n(n-1)}.$$
Probability Amplification

failure probability of $N$ repetitions:

$$\leq \left(1 - \frac{2}{n(n-1)}\right)^N \leq e^{-2N/n(n-1)}$$

$N = 10n(n-1)$: failure probability $\leq 10^{-8}$
run time $O(n^4)$

BasicMinCut

BasicMinCut($G$):
while $G$ has more than 2 vertices do
  pick a random edge $e$ in $G$
  $G \leftarrow G/e$
end while
return the size of the only cut in $G$
Claim: There exist algorithms $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \ldots$ s.t. $\forall i \geq 0$

- $\Pr[\mathcal{A}_i(G) = \mu(G)] \geq 1/2 \; \forall G$

- run time of $\mathcal{A}_i$ is $O(n^{f(i)})$, where $f(0) = 4$, $f(i+1) = 4(1 - 1/f(i))$

Proof: induction on $i$:

- $i = 0$: BasicMinCut
- $i \Rightarrow i+1$: contract until size $t = t(n)$, then call algorithm $\mathcal{A}_i$

apply probability amplification
**MinCut - Bootstrapping**

\[ \mathcal{A}_{i+1}(G) : \]
- set parameters \( t \) and \( N \) suitably
- repeat \( N \) times:
  - \( H \leftarrow \text{RandomContract}(G, t) \)
  - call \( \mathcal{A}_i(H) \)
- return smallest value

\[ \text{RandomContract}(G, t) : \]
- while \( |V(G)| > t \) do
  - for random \( e \in E(G) \)
    - \( G \leftarrow G/e \)
- end while
- return \( G \)

**Correctness of RandomContract:**

\[ \geq (1 - \frac{2}{n}) \cdot (1 - \frac{2}{n-1}) \cdot \ldots \cdot (1 - \frac{2}{t+1}) = \frac{t(t-1)}{n(n-1)} \]

**Correctness probability of one iteration:**

\[ \geq \frac{t(t-1)}{n(n-1)} \cdot \frac{1}{2} \]

**Failure probability of \( N \) repetitions:**

\[ \leq \left( 1 - \frac{t(t-1)}{2n(n-1)} \right)^N \leq e^{-N \cdot \frac{t(t-1)}{2n(n-1)}} \]