Algorithms, Probability, and Computing

Angelika Steger
Institut für Theoretische Informatik

steger@inf.ethz.ch

MST

<u>BORŮVKA'S ALGORITHM(G):</u>

$\forall v \in V$: compute $e_{\min}(v)$, insert $e_{\min}(v)$ in the MST and contract $e_{\min}(v)$
(removing loops and double edges by keeping only the cheapest edge)

recurse (until the graph contains only one vertex)

run time analysis:

#vertices	n	$\leqq$ n/2	$\leqq$ n/4	$\leqq$ n/8	...	O(m log n)
#edges	m	$\leqq$ m	$\leqq$ m	$\leqq$ m	...	

if we could show:

$\leqq$ m/2	$\leqq$ m/4	$\leqq$ m/8	...	O(m)

Randomized MST

RANDOMIZED MINUMUM SPANNING TREE ALGORITHM(G):

Perform three iterations of Borůvka's algorithm (which reduces the number of vertices to at most $n/8$) $\leqq C_B(n+m)$

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. $\leqq C(n/8+m/2)$

Use FINDHEAVY to find all unselected edges that are not T-heavy. $\leqq C_{FH}(n/8+m)$

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

recurse (until the graph contains only one vertex) $\leqq C\,(n/8+n+?)$

Assumptions:

run time of FindHeavy $\leqq C_{FH}(n+m)$ (for a graph with n vertices and m edges)

run time of three Boruvka steps $\leqq C_B(n+m)$ (for a graph with n vertices and m edges)

Claim:

run time of Randomized MST $\leqq C(n+m)$ (for a graph with n vertices and m edges)

Randomized MST

RANDOMIZED MINUMUM SPANNING TREE ALGORITHM(G):

Perform three iterations of Borůvka's algorithm (which reduces the number of vertices to at most $n/8$) $\leqq C_B(n+m)$

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. $\leqq C(n/8+m/2)$

Use FINDHEAVY to find all unselected edges that are not T-heavy. $\leqq C_{FH}(n/8+m)$

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

recurse (until the graph contains only one vertex) $\leqq C\,(n/8+n+$**?**$)$

Assumptions:

run time of FindHeavy $\leqq C_{FH}(n+m)$ (for a graph with n vertices and m edges)

run time of three Boruvka steps $\leqq C_B(n+m)$ (for a graph with n vertices and m edges)

Claim: there exists C s.t.

run time of Randomized MST $\leqq C(n+m)$ (for a graph with n vertices and m edges)

BasicMinCut

```
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: $O(n^2)$

BasicMinCut

Correctness Analysis

```
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
```

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[BasicMinCut(G) returns size of min cut]

$$\geq \frac{n-2}{n} \cdot \frac{n-3}{n-1} \cdot \frac{n-4}{n-2} \cdots \frac{2}{4} \cdot \frac{1}{3} \cdot p_0(2) = \frac{2}{n(n-1)}.$$

BasicMinCut

Probability Amplification

```
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
```

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 $\leqq 10^{-8}$

run time $O(n^4)$

MinCut - Bootstrapping

Claim: There exist algorithms $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \ldots$ s.t. $\forall\, i \geqq 0$

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

- 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

$\underline{\mathcal{A}_{i+1}(G):}$

set parameters t and N suitably

repeat N times:

    $H \leftarrow \textsc{RandomContract}(G, t)$

    call $\mathcal{A}_i(H)$

return smallest value

$\underline{\textsc{RandomContract}(G, t):}$

while $|V(G)| > t$ do

    for random $e \in E(G)$

        $G \leftarrow G/e$

end while

return G

correctness of RandomContract:

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

correctness probability of one iteration:

$$\geq \frac{t(t-1)}{n(n-1)} \cdot 1/2$$

failure probability of N repetitions:

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