COL751 - Lecture 20

1 Computing Edge-Connectivity / Global-Min-Cut in $O(n^4 \log n)$ time

Let $G = (V, E)$ be an n-vertex undirected unweighted **multigraph without self-loops**. We present below a simple algorithm for computing global min-cut of G.

Lemma 1. *Let (X, X^c) be a min-cut of G, and $e \in E$ be a uniformly random edge in G. Then $Prob(e \in (X, X^c)) \leqslant \frac{2}{n}$.*

Proof: Let δ be the size of min-cut of G. Then degree of each vertex is at least δ, and so the number of edges is at least $n\delta/2$. Thus,

$$Prob(e \in (X, X^c)) \;=\; \frac{\delta}{m} \;\leqslant\; \frac{\delta}{n\delta/2} \;\leqslant\; \frac{2}{n}. \qquad \square$$

Lemma 2. *Let (X, X^c) be a min-cut of G, and H be a graph with two supernodes obtained on doing a sequence of $n-2$ edge contract operations (by picking a uniformly random edge each time). Then, probability edges in H corresponds to cut (X, X^c) is at least $1/\binom{n}{2}$.*

Proof: By Lemma 1, the probability edges in H corresponds to cut (X, X^c) is at least

$$\left(1 - \frac{2}{n}\right)\left(1 - \frac{2}{n-1}\right)\cdots\left(1 - \frac{2}{3}\right) \;=\; \frac{(n-2)!}{n!/2} \;\geqslant\; \frac{2}{n(n-1)}.$$

This proves the desired lower bound of $1/\binom{n}{2}$. $\square$

Based upon Lemma 2 we have the following algorithm for computing global minimum-cut.

for $i = 1$ *to* $4n^2 \log n$ **do**
    Initialize H as G;
    while $(|V(H)| > 2)$ **do**
        Pick a uniformly random edge e in H and contract it;
    (A_i, B_i) = Partition of G induced by two supernodes in H;
Return smallest cut among cuts $(A_1, B_1), \ldots, (A_{4n^2 \log n}, B_{4n^2 \log n})$;

Algorithm 1: Simple-Min-Cut(G)

The running time of above algorithm is $O(n^4 \log n)$ as each contraction takes $O(n)$ time in worst case. So, we have following result.

Theorem 1. *Algorithm 1 computes a global min-cut of G with probability $1 - 1/n^2$, and its running time is $O(n^4 \log n)$.*

Homework Use Lemma 2 to obtain a bound on total number of distinct global min-cuts in an unweighted graph G.

2 Global Min-Cut in $O(n^2 \log^3 n)$ time

Lemma 3. *In Algorithm 1 we if we perform merge/contract operations till k vertices are left, then probability of preserving a cut is $\approx k^2/n^2$.*

Proof: Let (X, X^c) be a min-cut of G. Probability of preserving this cut until k vertices are left is at least:

$$\left(1-\frac{2}{n}\right)\left(1-\frac{2}{n-1}\right)\cdots\left(1-\frac{2}{k+1}\right) = \frac{(n-2)!/(k-2)!}{n!/k!} = \frac{k(k-1)}{n(n-1)} \approx \frac{k^2}{n^2}.$$

This proves the claim. □

Observe that probability of preserving a cut is at least 1/4 till $n/2$ contractions. It is only when graph becomes small the probability of preserving the cut rapidly falls.

```
Initialize H_1, H_2 as G;
while (|V(H_1)|, |V(H_2)| > n/√2) do
    Pick a uniformly random edge e_1 in H_1, and contract it;
    Pick a uniformly random edge e_2 in H_2, and contract it;
(A_1, B_1) = Fast-Min-Cut(H_1);
(A_2, B_2) = Fast-Min-Cut(H_2);
Return smaller of the two cuts (A_1, B_1) and (A_2, B_2);
```

Algorithm 2: Fast-Min-Cut(G)

Lemma 4. *Running time of Algorithm 2 is $O(n^2 \log n)$.*

Proof: Running time satisfies the relation

$$T(n) = 2T(n/\sqrt{2}) + cn^2.$$

Solving this recursion gives $T(n) = O(n^2 \log n)$. □

Lemma 5. *The success probability of Fast-Min-Cut is $\frac{1}{\Theta(\log n)}$.*

Proof: Suppose $p(n)$ is success probability of Fast-Min-Cut on an input graph with n vertices. We have

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

So,

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

Let,

$$q(k) = \frac{1}{p(\sqrt{2}^k)}.$$

Then,

$$
\begin{aligned}
\frac{1}{q(k)} &= \frac{1}{q(k-1)} - \frac{1}{4\,q(k-1)^2} \\
\frac{1}{q(k)} &\geqslant \frac{1}{q(k-1)} - \frac{1}{3\,q(k-1)q(k)} \\
q(k-1) &\geqslant q(k) - \frac{1}{3} \\
q(k) &\leqslant \frac{k}{3} = \log_{\sqrt{2}} n.
\end{aligned}
$$

Thus, $p(n) \geqslant \frac{1}{\Theta(\log n)}$. □

In order to boost the probability, we need to run Algorithm 2 order $\log^2 n$ times. We thus get the following result.

Theorem 2. *There exists an algorithm that for any n vertex multigraph G computes with success probability $(1-1/n^2)$ a global min-cut of G in $O(n^2 \log^3 n)$ time.*