Randomized min-cut algorithm

Consider the randomized min-cut algorithm discussed in class. We have seen that its probability of success is at least $1/\binom{n}{2}$, where n is the number of its vertices.

- Describe how to implement the algorithm when the graph is represented by adjacency lists, and analyze its running time. In particular, a contraction step can be done in $O(n)$ time.
- A weighted graph has a weight $w(e)$ on each edge e, which is a positive real number. The min-cut in this case is meant to be min-weighted cut, where the sum of the weights in the cut edges is minimum. Describe how to extend the algorithm to weighted graphs, and show that the probability of success is still $\geq 1/\binom{n}{2}$. [hint: define the weighted degree of a node]
- Show that running the algorithm multiple times independently at random, and taking the minimum among the min-cuts thus produced, the probability of success can be made at least $1 - 1/n^c$ or a constant $c > 0$ (hence, with high probability).

SOLUTION

First Point

Second Point

In the algorithm we change the way we choose an edge to contract: edges with high weight will have a greater probability to be chosen with respect to others with lower weight. We define $cost : E \longrightarrow \mathbb{R}$ as the function that associate the cost to a given edge.

For each edge $x \in E$ the probability to be choose for a contraction is given by

$$Pr[extract\ x] = \frac{cost(x)}{c(E)}$$
$$\texttt{where } c(E) = \sum_{e \in E} cost(e)$$

Now the probability of making an error, when extracting an edge at random, is the probability of extracting one of the "bad" edges.

$$Pr[error] = \sum_{e \in BAD} Pr[extract\ e] = \sum_{e \in BAD} \frac{cost(e)}{c(E)} = \frac{\sum_{e \in BAD} cost(e)}{c(E)} \qquad [1]$$

As usual bad edges (BAD) are those belonging to every *min cut*, so the total cost of bad edges is less or equal to the cost of *min cut.*

$$\sum_{e \in BAD} cost(e) \leq cost(min\_cut) \qquad [2]$$

For the same reasons explained in class, for each node v the cost of *min cut* must be less or equal to the sum of the costs of the edges connected at v.

$$\forall\ n \in V\ .\ c(min\_cut) \leq \sum_{e \in star(n)} cost(e) = cost(star(n))$$
$$\texttt{where } n' \in star(n) \ \texttt{ iff } \ (n', n) \in E$$

Where we refer to $\sum_{e \in star(n)} cost(e)$ as $cost(star(n))$. Now we can reformulate the *handshaking lemma* in the following way.

$$c(E) = \frac{\sum_{v \in V} cost(star(v))}{2}$$

Now we have all the ingredients for the proof:

$$c(E) = \frac{\sum_{v\in V} cost(star(v))}{2} \geq \frac{|V|cost(min\_cut)}{2}$$
$$cost(min\_cut) \leq \frac{2c(E)}{|V|} \qquad [3]$$

Finally

$$\begin{aligned} Pr[error] &= \frac{\sum_{e\in BAD} cost(e)}{c(E)} && [\texttt{for } 1] \\ &\leq \frac{cost(min\_cut)}{c(E)} && [\texttt{for } 2] \\ &\leq \frac{\frac{2c(E)}{|V|}}{c(E)} && [\texttt{for } 3] \\ &= \frac{2c(E)}{|V|c(E)} = \frac{2}{|V|} \end{aligned}$$

The probability of making an error when selecting an edge is the same, so the probability of success is still $\geq 1/\binom{n}{2}$ as for the case seen in class.

Third Point

We know that $P(n) \geq 1/\binom{n}{2}$ is the probability of choose well for N times. We have seen also that the probability of error is $\leq (1 - 1/\binom{n}{2})$. Then the probability of having an error each time in N repetitions of the algorithm is

$$leq(1 - \frac{1}{\binom{n}{2}})^N \approx e^{-\frac{2N}{n(n-1)}}$$

Given a c we want the probability of success to be $\geq 1 - 1/n^c$. This is equivalent to ask that the probability of error is $\leq 1/n^c$. Making some calculations:

$$\frac{1}{e^{\frac{2N}{n(n-1)}}} = e^{-\frac{2N}{n(n-1)}} \leq \frac{1}{n^c}$$
$$n^c \leq e^{\frac{2N}{n(n-1)}}$$
$$ln(n^c) \leq ln(e^{\frac{2N}{n(n-1)}}) = \frac{2N}{n(n-1)}$$
$$N \geq \frac{n(n-1)ln(n^c)}{2}$$

So, for each given c, to make the probability of success at least $1 - 1/n^c$ is sufficient to take N grater or equal than $\frac{1}{2}n(n-1)ln(n^c)$.