Randomized Contraction

1 Introduction

The **minimum cut** problem is defined as follows: Given an undirected graph $G = (V, E)$ possibly containing parallel edges, compute the cut with the fewest number of crossing edges, called the minimum cut.

2 Algorithm

```
def MinCut(G):
    while len(V) > 2:
        randomly select remaining (u,v) edge
        merge/contract into single vertex
        remove self loops
    return cut represented by remaining 2 vertices.
```

It is important to note that this is a randomized algorithm that has a non-negligible chance of failure - thus, it is necessary to run the algorithm many times with different RNG seeds to guarantee a correct result.

3 Analysis

To analyze the success rate and runtime of the randomized contraction algorithm, we need to start with three givens. First, fix a graph $G = (V, E)$ with n vertices and m edges, with a min-cut (A, B), and let k be the number of crossing edges forming the set F. Two important observations arise immediately:

1. If we contract an edge in F, the algorithm will not return the correct result.
2. If we only contract edges contained within A and B, we will get the min-cut of the graph.

Let S_i represent the event that an edge in F is contracted in iteration i. Observing that the degree of each vertex is at least k,

$$\sum_v degree(v) = 2m \tag{1}$$

$$\sum_v degree(v) \geq kn \tag{2}$$

$$m \geq \frac{kn}{2} \tag{3}$$

$$Pr[S_1] = \frac{2}{n} \tag{4}$$

For the second iteration:

$$Pr[\neg S_1 \wedge \neg S_2] = Pr[\neg S_1 | \neg S_2] \times Pr[\neg S_1] \tag{5}$$

$$Pr[\neg S_1 \wedge \neg S_2] \geq 1 - \frac{k}{\text{remaining edges}} \times (1 - \frac{2}{n}) \tag{6}$$

Remember that $m \geq \frac{kn}{2}$.

$$Pr[\neg S_1 | \neg S_2] = 1 - \frac{2}{n-1} \tag{7}$$

Now we can generalize for all iterations.

$$Pr[\neg S_1 \wedge \ldots \wedge \neg S_{n-2}] = \left(1 - \frac{2}{n}\right)\left(1 - \frac{2}{n-1}\right)\ldots\left(1 - \frac{2}{n-(n-3)}\right) \tag{8}$$

$$= \frac{n-2}{n} \cdot \frac{n-3}{n-1} \cdot \frac{n-4}{n-2} \cdots \frac{2}{4} \cdot \frac{1}{3} \tag{9}$$

$$= \frac{2}{n(n-1)} \geq \frac{1}{n^2} \tag{10}$$

Thus, the chance of success on a single attempt with randomized selection is very unlikely.

3.1 Repeated Trials

Due to the significant failure rate, finding the minimum cut actually involves running the algorithm many times and remembering the smallest cut. Let T_i be the event that (A, B) is found on the i^{th} try and N be the number of times the algorithm is run.

$$Pr[fail] = Pr[\neg T_1 \wedge \neg T_2 \wedge \ldots \wedge \neg T_N] \tag{11}$$

$$= \prod_{i=1}^{N} Pr[\neg T_i] \leq \left(1 - \frac{1}{n^2}\right)^N \tag{12}$$

Using the fact that $\forall x \in \mathbb{R}, 1 + x \leq e^x$:

- For $N = n^2, Pr[fail] \leq e\left(e^{-\frac{1}{n^2}}\right)^{n^2} = \frac{1}{e} \approx 30\%$
- For $N = n^2 \cdot \ln(n), Pr[fail] \leq \frac{1}{n}$

Runtime is therefore $\Omega(n^2 m)$ with slow constants.