CSE 521: Design and Analysis of Algorithms I **Spring 2016**

Lecture 1: Contraction Algorithm

Lecturer: Shayan Oveis Gharan *March 28th* *Scribe: Mohammad Javad Hosseini*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

In the first few lectures, we will study several randomized algorithms for various optimization tasks. We start with a simple randomized algorithm to find the min-cut of a graph. The algorithm is introduce by David Karger [Kar93]. Then, we show an algorithm with improved running time by Karger and Stein [KS93].

1.1 Karger algorithm for the min-cut problem

Suppose we have a graph $G = (V, E)$. If we partition the graph to $(S, \bar{S})$, $E(S, \bar{S})$ are the edges that have one end-point in each partition (See Figure 1.1). The min-cut problem is the find a 2-partition (or a cut) with minimum number of edges, i.e., $\min_S |E(S, \bar{S})|$.

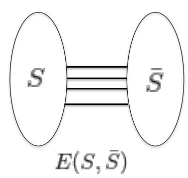

Figure 1.1: Partitioning of a graph

Karger's Algorithm Do the following $n-2$ times: Pick an edge in the graph uniformly at random and contract it (delete the loops).

In the following figure, contracting the edges in the specified order, will find the min-cut. Note that in this case, we were lucky and did not contract the edges that were in the min-cut.

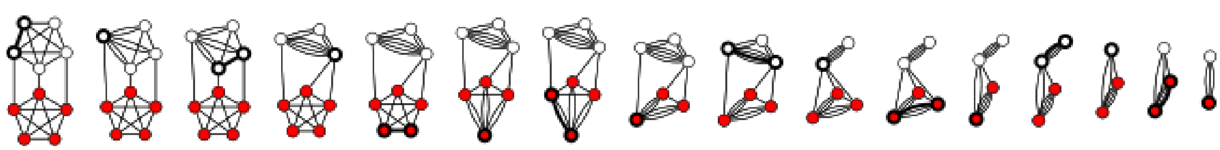

Figure 1.2: Illustration of Karger's algorithm [source: https://en.wikipedia.org/wiki/Karger's_algorithm]

In order to further understand the algorithm, we can think of its connection to clustering. Say you have K clusters such that most of the edges are inside the clusters. If we run Karger's Algorithm, at the time we have K super-nodes, we can consider them as the clusters. Note that since there are a few edges between the clusters, the algorithm is more likely to contract intra cluster edges (as opposed to the inter-cluster edges.

Theorem 1.1. *For any min-cut* $(S, \bar{S})$*, Karger's algorithm returns* $(S, \bar{S})$ *w.p at least* $\frac{1}{\binom{n}{2}} = \frac{2}{n(n-1)}$.

Proof. First, observe that the algorithm succeeds if we do not contract any edges of the min-cut in all steps of the algorithm. Secondly, observe that no matter which edges we contract, the value of the minimum cut never decreases in the entire run of the algorithm.

Let's say the value of the min-cut is C. Let $\mathcal{A}_i$ be the event that we contract an edge of min-cut at step i. Note that at the step i, if we have not touched any of the edges in the min-cut, each node has at C incident edges. Therefore, at the end of step i the graph has at least $(n-i)C/2$ edges.

We can write,

$$\begin{aligned}
&\mathbb{P}[\mathcal{A}_1] = \frac{C}{|E|} \le \frac{C}{nC/2} = \frac{2}{n} \\
&\mathbb{P}[\mathcal{A}_2|\neg\mathcal{A}_1] \le \frac{C}{(n-1)C/2} = \frac{2}{n-1} \\
&\dots \\
&\mathbb{P}[\mathcal{A}_{n-2}|\neg\mathcal{A}_1, \dots, \neg\mathcal{A}_{n-3}] \le \frac{C}{(3)C/2} = \frac{2}{3}
\end{aligned} \tag{1.1}$$

Therefore, we have:

$$\begin{aligned}
\mathbb{P}[success] = \mathbb{P}\left[\bigcap_{i=1}^{n-2}(\neg\mathcal{A}_i)\right] &= \mathbb{P}[\neg\mathcal{A}_1]\,\mathbb{P}[\neg\mathcal{A}_2|\neg\mathcal{A}_1]\dots\mathbb{P}[\neg\mathcal{A}_{n-2}|\neg\mathcal{A}_1,\dots,\neg\mathcal{A}_{n-3}] \\
&\ge (1-\frac{2}{n})(1-\frac{2}{n-1})\dots(1-\frac{2}{3}) \\
&= \frac{n-2}{n}\frac{n-3}{n-1}\frac{n-4}{n-2}\dots\frac{1}{3} = \frac{2}{n(n-1)}
\end{aligned}$$

This completes the proof of Theorem 1.1. □

Corollary 1.2. *Any graph has at most* $\binom{n}{2}$ *min-cuts.*

We should also note that the bound is tight for cycles.

1.1.1 Boosting up the probabilities

We can run independent copies of the algorithm to boost up the probabilities: Run Karger's algorithm K times and return the min cut found in all of these runs.

Observe that this method will succeed w.p. $1-(1-\frac{1}{\binom{n}{2}})^K$. We know that in general $1-x \le \exp(-x)$. Therefore, we have:

$$(1-\frac{1}{\binom{n}{2}})^K \le \exp(\frac{-K}{\binom{n}{2}}). \tag{1.2}$$

If we choose $K \approx n^2 \log n$, we have

$$P(success) \ge 1 - \frac{1}{n^2}. \tag{1.3}$$

We say an event happens with high probability, if the probability converges to 1 as the size of the problems goes to infinity. By the above calculation, for $K \geq n^2 \log n$, the algorithm succeeds with high probability.

Note that for each of the independent runs, the running time is $\mathcal{O}(n^2)$. Therefore, the total running time is $\mathcal{O}(n^4 \log n)$, which is slow! In the next section we see a nice idea to improve the running time to $\mathcal{O}(n^2 \log^3 n)$.

1.2 Improving the running time: Karger-Stein Algorithm (1996)

Theorem 1.3 ([KS93])**.** *There is an algorithm that runs in $\mathcal{O}(n^2 \log n^3)$ and finds the min-cut with high probability.*

The idea of the algorithm is quite simple. First, observe that for any ℓ,

$$\mathbb{P}\left[\bigcup_{i=1}^{n-\ell}\right] = \frac{\binom{\ell}{2}}{\binom{n}{2}}.$$

So, for $\ell = n/\sqrt{2}$, the cut $(S, \overline{S})$ remains after $n - \ell$ steps with probability at least

$$\frac{\frac{\ell(\ell-1)}{2}}{\frac{n(n-1)}{2}} \approx \frac{(n/\sqrt{2})^2}{(n^2} = 1/2.$$

In other words, the algorithm is doing a very good job in the first constant fraction of contraction steps. Only when we get down to a graph with a few nodes the probability of success decreases rapidly.

The idea is motivated from *fault tolerance* applications. If we have a system that fails with probability p, we should keep at least $1/p$ copies of our system to ensure that at least one copy remains safe in expectation. That is exactly the idea of the Karger, Stein algorithm. We run two copies of the contraction algorithm for $n - n/\sqrt{2}$ steps because the probability of failure is only $1/2$. After that for the next $m - m/\sqrt{2}$ steps where $m = n - n/\sqrt{2}$ we increase the number of independent copies to 4. This is because the probability of failure has gown down to $1/4$. We keep increasing the number of copies geometrically as the size of the graphs gets smaller and smaller. At the end we return the best cut found in all of the copies.

Algorithm Karger-Stein (1996): Run two independent copies of the Karger's algorithm for $n - \ell$ steps, for $\ell = n/\sqrt{2}$. Run the algorithm recursively on the two remaining graphs.

The following two claims bound the running time and the success probability of this algorithm.

Claim 1.4. *The running time is $\mathcal{O}(n^2 \log(n))$.*

Proof. Let $T(n)$ the the running time of the algorithm on a graph of size n.

$$\begin{aligned} T(n) &= \mathcal{O}(n^2) + 2T(\frac{n}{\sqrt{2}}) \\ &\approx n^2 + 2(\frac{n}{\sqrt{2}})^2 + 4(\frac{n}{\sqrt{2}^2})^2 + 8(\frac{n}{\sqrt{2}^3})^2 + \dots \\ &= n^2(1 + \frac{2}{\sqrt{2}^2} + \frac{2^2}{\sqrt{2}^4} + \frac{2^3}{\sqrt{2}^6} + \dots) \approx n^2 \log n. \end{aligned} \tag{1.4}$$

Note that here, for the sake of simplicity, we ignored some constants.

$\square$

Claim 1.5. $P\Big[\textit{success}\Big] \geq \frac{1}{\log n}$

Proof. Let's define $P(n)$ be the probability that the algorithm succeeds in a graph of size n. Next, we write a recurrence relation for $P(n)$. Note that $\frac{1}{2}P(n/\sqrt{2})$ is the probability that one of the two copies succeeds. So, $1 - \frac{1}{2}P(n/\sqrt{2})$ is the probability that one of the two copies fail. So, $(1 - \frac{1}{2}P(n/\sqrt{2}))^2$ is the probability that both copies fail, i.e., the algorithm fails. So,

$$P(n) = 1 - \left(1 - \frac{1}{2}P\left(\frac{n}{\sqrt{2}}\right)\right)^2. \tag{1.5}$$

We use induction to show that $P(n) \geq \frac{1}{\log n}$: By definition,

$$\begin{aligned} P(n) &= 1 - \left(1 + \frac{1}{4}P\left(\frac{n}{\sqrt{2}}\right)^2 - P\left(\frac{n}{\sqrt{2}}\right)\right) \\ &= P\left(\frac{n}{\sqrt{2}}\right) - \frac{1}{4}P\left(\frac{n}{\sqrt{2}}\right)^2 \end{aligned}$$

Now, by induction hypothesis we know $P(n/\sqrt{2}) \geq \frac{1}{\log(n/\sqrt{2})}$. So, we would like to use this lower bound for both of the terms in the RHS, and obtain a lower bound on $P(n)$. Unfortunately, the second term in the RHS has a negative coefficient, so we may increase the RHS when we replace $P(n/\sqrt{2})^2$ by $\frac{1}{\log^2(n/\sqrt{2})}$.

However, observe that the function $f(x) = x - \frac{x^2}{4}$ is an increasing function of x for $x \leq 1$. Therefore, if $x \geq a$, one can write $f(x) \geq a - a^2/4$. We do the same above and we write

$$P(n) \geq \frac{1}{\log(n/\sqrt{2})} - \frac{1}{4\log^2(n/\sqrt{2})}.$$

To prove the induction step we must show $P(n) \geq \frac{1}{\log n}$. So, it is enough to show that

$$\begin{aligned} & \frac{1}{\log(n/\sqrt{2})} - \frac{1}{4\log^2(n/\sqrt{2})} \geq \frac{1}{\log n} \\ \iff & \frac{1}{\log \frac{n}{\sqrt{2}}}\left(1 - \frac{1}{4\log \frac{n}{\sqrt{2}}}\right) \geq \frac{1}{\log n} \\ \iff & \frac{\log \frac{n}{\sqrt{2}}}{\log n} + \frac{1}{4\log \frac{n}{\sqrt{2}}} \leq 1 \\ \iff & \frac{\log n - \frac{1}{2}}{\log n} + \frac{1}{4\log \frac{n}{\sqrt{2}}} \leq 1 \\ \iff & \frac{1}{4\log \frac{n}{\sqrt{2}}} \leq \frac{1}{2\log n}. \end{aligned}$$

The last inequality holds trivially. This completes the proof of Claim 1.5 □

Now, to prove Theorem 1.3, it is enough to run $K = (\log n)^2$ independent copy of the Karger, Stein algorithm. Then we will find the min cut with probability at least $1 - 1/n$ as desired.

$$1 - \left(1 - \frac{1}{\log n}\right)^{(\log n)^2} \approx 1 - \frac{1}{n}.$$

References

[Kar93] D. R. Karger. "Global min-cuts in RNC, and other ramifications of a simple min-out algorithm". In: *SODA*. Austin, Texas, United States: Society for Industrial and Applied Mathematics, 1993, pp. 21–30 (cit. on p. 1-1).

[KS93] D. R. Karger and C. Stein. "An O (n^2) algorithm for minimum cuts". In: *STOC*. 1993, pp. 757–765 (cit. on pp. 1-1, 1-3).