CSCE-658 Randomized Algorithms

Lecture #2, January 26, 2016

Lecturer: Professor Jianer Chen

2 Karger-Stein's MIN-CUT algorithm

We now present an improvement on Karger's algorithm for the MIN-CUT problem. The algorithm is due to Karger and Stein [12].

Recall the algorithm CONTRACTION we discussed in Section 1:

Algorithm 4 CONTRACTION Input: An undirected graph G; Output: A cut of G;

1. $G_0 = G; h = 0;$

2. while G_h has more than 2 vertices do

randomly pick an edge e_h in G_h ;

 $G_{h+1} = G_h/e_h; h = h+1;$

3. return all edges in G_{n-2} .

As we have shown in section 1, the probability $\Pr[E_h | \bigcap_{i=1}^{h-1} E_i]$ that, under the condition that the first (h-1)-st iterations of the algorithm do not pick an edge in the min-cut C of G, the probability that the h-th iteration does not pick an edge in C is at least $1-2/n_{h-1} = (n-h-1)/(n-h+1)$, where $n_{h-1} = n - h + 1$ is the number of vertices in the graph G_{h-1} . When n_{h-1} is larger, this probability is good. However, when n_{h-1} gets smaller (for example, when h = n - 2, $n_{(n-2)-1} = n_{n-3} = 3$), the probability $1 - 2/n_{h-1}$ becomes bad: for instance, for h = n - 2, $1 - 2/n_{h-1} = 1/3$. As a result, we were only able to show that the probability that the algorithm CONTRACTION returns a min-cut of the input graph is not smaller than $2/(n(n-1) \approx 2/n^2$. Thus, we need to repeat $\Omega(n^2)$ times of the algorithm CONTRACTION, in order to have a good probability that the algorithm returns a correct solution, which has a significant impact on the complexity of the algorithm.

In order to improve the efficiency of the algorithm for MIN-CUT, we can consider how to increase the success probability of the algorithm CONTRACTION. Since the probability (n - h - 1)/(n - h + 1)gets small when h gets large, if we stop the iteration in the algorithm CONTRACTION earlier when the number $n_{h-1} = n - h + 1$ of vertices in the graph G_{h-1} is sufficiently larger, we should end up with a good probability of success. However, then how do we deal with the resulting graph, which still has a significant number of vertices?

Karger and Stein's idea is to work on the small graph G_{h-1} multiple times to increase the success probability. Consider the following algorithm:

Algorithm 5 CONTRACTION-II Input: An undirected graph G; Output: A cut of G;

1. if G has no more than 6 vertices

then construct the min-cut of G by brute-force method; return;

2. $G' = G; \quad G'' = G;$

3. repeat $n(1-1/\sqrt{2}) - 1$ times: randomly pick an edge e' in G'; G' = G'/e';

4. repeat $n(1-1/\sqrt{2}) - 1$ times: randomly pick an edge e'' in G''; G'' = G''/e'';

- 5. $C' = \text{CONTRACTION-II}(G'); \quad C'' = \text{CONTRACTION-II}(G'');$
- 6. return the smaller of C' and C''.

For simplicity, we have assumed that $n(1-1/\sqrt{2})-1$ is an integer, which will simplify the analysis. If we want to be more precise, we may use $\lceil n(1-1/\sqrt{2})-1 \rceil$ instead, which will lead to the same conclusions but will make the analysis more tedious. The algorithm CONTRACTION-II starts with an input graph

G of n vertices, applies the contraction operations $n(1-1/\sqrt{2})-1$ times to construct two graphs G'and G'' of $n/\sqrt{2}+1$ vertices, recursively works on the two smaller graphs G' and G'', and finally picks the smaller cut returned by the two recursive calls. Note that the two smaller graphs G' and G'' are obtained by two different random contraction sequences, instead of a single such a random contraction sequence. This does not increase the order of the complexity of the algorithm (see the analysis below). On the other hand, this simplifies the probability analysis. Thus, besides the contraction operations, the algorithm reduces the problem on the given instance that is a graph of n vertices into the problem on two smaller instances, which are graphs G' and G'' of $n/\sqrt{2}+1$ vertices (in the following, sometime in order to simplify the analysis, we simply take an approximation of this graph size, i.e., assuming that the graphs G' and G'' have $n/\sqrt{2}$ vertices).

The complexity of the algorithm CONTRACTION-II can be derived based on the standard techniques for divide-and-conquer algorithms. For this, let T(n) be the running time of the algorithm CONTRACTION-II on a graph G of n vertices. By Lemma 1.1, randomly picking an edge then contracting it in a graph of n vertices can be done in time O(n). Therefore, besides the recursive calls, the total time of the algorithm CONTRACTION-II is $O(n^2)$ (note that even if we construct a single graph of $n/\sqrt{2}$ vertices by a sequence of random contractions, we still need the amount of time of order $O(n^2)$). This gives the following recurrence relation:

$$T(n) = 2T(n/\sqrt{2}) + O(n^2)$$
 and $T(n) = O(1)$ for $n \le 6$.

This implies that there is a constant c > 0 such that

$$T(n) \le 2T(n/\sqrt{2}) + cn^2$$
 and $T(n) \le c$ for $n \le 6$.

It is easy to verify, using induction, that $T(n) \leq 3cn^2 \log n$. Therefore, the running time of the algorithm CONTRACTION-II is $O(n^2 \log n)$.

It remains to show that the algorithm CONTRACTION-II has a better probability to succeed than that of the algorithm CONTRACTION. As before, let C be a fixed min-cut of the input graph G, and let E_h be the event that the h-th iteration of the loop in step 3 of the algorithm CONTRACTION-II does not pick an edge in the min-cut C, for $1 \le h \le n(1 - 1/\sqrt{2}) - 1$, then

$$\Pr\left[E_h \mid \bigcap_{i=1}^{h-1} E_i\right] \ge \frac{n_{h-1}-2}{n_{h-1}} = \frac{n-h-1}{n-h+1}$$

and

$$\Pr\left[\prod_{i=1}^{n(1-1/\sqrt{2})-1} E_{i}\right]$$

$$= \Pr\left[E_{n(1-1/\sqrt{2})-1} \left|\prod_{i=1}^{n(1-1/\sqrt{2})-2} E_{i}\right] \Pr\left[E_{n(1-1/\sqrt{2})-2} \left|\prod_{i=1}^{n(1-1/\sqrt{2})-3} E_{i}\right] \cdots \Pr[E_{2}|E_{1}] \cdot \Pr[E_{1}]\right]$$

$$\geq \left(\frac{n/\sqrt{2}}{n/\sqrt{2}+2}\right) \left(\frac{n/\sqrt{2}+1}{n/\sqrt{2}+3}\right) \left(\frac{n/\sqrt{2}+2}{n/\sqrt{2}+4}\right) \cdots \left(\frac{n-4}{n-2}\right) \left(\frac{n-3}{n-1}\right) \left(\frac{n-2}{n}\right)$$

$$= \frac{n/\sqrt{2} \cdot (n/\sqrt{2}+1)}{n(n-1)}$$

$$\geq \frac{n^{2}/2}{n^{2}}$$

$$= \frac{1}{2}.$$
(3)

Therefore, the probability that step 3 of the algorithm CONTRACTION-II returns a smaller graph G' that has C as a min-cut is at least 1/2. The same method shows that step 4 of the algorithm CONTRACTION-II returns a smaller graph G'' that has C as a min-cut with a probability at least 1/2. Note that the event that G' has C as a min-cut and the event that G'' has C as a min-cut are independent.

Now let P(n) be the probability that the algorithm CONTRACTION-II returns a min-cut of the input graph G of n vertices. A sufficient condition for the cut C' constructed in step 5 to be a min-cut of the graph G is that the graph G' constructed in step 3 has the min-cut C of G as a min-cut and that the recursive call CONTRACTION-II(G') in step 5 returns a min-cut of the graph G'. Therefore,

 $\Pr[\text{the cut } C' \text{ is a min-cut of the graph } G]$

- $\geq \Pr[G' \text{ has } C \text{ as a min-cut } \& \text{ CONTRACTION-II}(G') \text{ returns a min-cut of } G']$
- = $\Pr[G' \text{ has } C \text{ as a min-cut}] \cdot \Pr[\text{CONTRACTION-II}(G') \text{ returns a min-cut of } G']$

$$\geq \frac{P(n/\sqrt{2})}{2}$$

where $\Pr[G' \text{ has } C \text{ as a min-cut}] \geq 1/2$ comes from (3). Note that G' is a graph of $n/\sqrt{2}$ vertices, and that the equality in the derivation above is based on the fact that the event "G' has C as a min-cut" and the event "CONTRACTION-II(G') returns a min-cut of G'" are independent. Therefore, the probability that the cut C' constructed in step 5 is not a min-cut of the graph G is bounded by $1 - P(n/\sqrt{2})/2$. A similar reasoning concludes that the probability that the cut C'' constructed in step 5 is not a min-cut of the graph G is also bounded by $1 - P(n/\sqrt{2})/2$

Note that the algorithm CONTRACTION-II fails in returning a min-cut of G if and only if neither of C' and C'' is a min-cut of G. Therefore,

 $\Pr[\text{the algorithm CONTRACTION-II fails in returning a min-cut of } G]$

- = $\Pr[C' \text{ is not a min-cut of } G \& C'' \text{ is not a min-cut of } G]$
- = $\Pr[C' \text{ is not a min-cut of } G] \cdot \Pr[C'' \text{ is not a min-cut of } G]$

$$\leq \left(1 - \frac{P(n/\sqrt{2})}{2}\right)^2,$$

This derives that the probability P(n) that CONTRACTION-II returns a min-cut of the graph G with n vertices is at least $1 - (1 - P(n/\sqrt{2})/2)^2$, i.e.,

$$P(n) \ge 1 - \left(1 - \frac{P(n/\sqrt{2})}{2}\right)^2 = P(n/\sqrt{2}) - \frac{1}{4} \left(P(n/\sqrt{2})\right)^2 \tag{4}$$

It is perhaps difficult to solve the recurrence relation (4), but it is rather easy to verify that $P(n) \ge 1/\log n$. First of all, for $n \le 6$, we surely have $P(n) = 1 \ge 1/\log n$ (here we assume $n \ge 2$). Now assume by induction that $P(k) \ge 1/\log k$ for k < n. Then

$$\begin{split} P(n) &\geq P(n/\sqrt{2}) - \frac{1}{4} \left(P(n/\sqrt{2}) \right)^2 \\ &\geq \frac{1}{\log(n/\sqrt{2})} - \frac{1}{4\log^2(n/\sqrt{2})} \\ &= \frac{1}{\log n - 1/2} - \frac{1}{4(\log n - 1/2)^2} \\ &= \frac{4\log n - 3}{4\log^2 n - 4\log n + 1} \\ &= \frac{1}{\log n} + \frac{1 - 1/\log n}{4\log^2 n - 4\log n + 1} \\ &\geq \frac{1}{\log n} \end{split}$$

This derives

Lemma 2.1 The probability that the algorithm CONTRACTION-II returns a min-cut of the input graph G is at least $1/\log n$.

Now we can play the same trick by repeating the algorithm CONTRACTION-II sufficiently many times to achieve a good success probability.

Algorithm 6 KARGER-STEIN Input: An undirected graph G; Output: A cut of G;

1. run the algorithm CONTRACTION-II $t \log n$ times;

2. return the cut that is the smallest among those constructed in step 1.

Theorem 2.2 The algorithm KARGER-STEIN returns a min-cut of the input graph G with a probability at least $1 - 1/e^t$, where $e = 2.718 \cdots$ is the base of the natural logarithm.

PROOF. The algorithm KARGER-STEIN fails in returning a min-cut of the graph G if and only if all calls to CONTRACTION-II in step 1 of the algorithm fail in returning a min-cut of G. By Lemma 2.1, each call to CONTRACTION-II in step 1 of the algorithm KARGER-STEIN fails in returning a min-cut of G with a probability bounded by $1-1/\log n$. Therefore, the probability that all these calls in step 1 fail in returning a min-cut of G, i.e., the probability that the algorithm KARGER-STEIN fails in returning a min-cut of G, is bounded by

$$\left(1 - \frac{1}{\log n}\right)^{t\log n} < e^{-t}.$$

where we have used Lemma 1.6. In conclusion, the probability that the algorithm KARGER-STEIN returns a min-cut of G is at least $1 - e^{-t} = 1 - 1/e^t$.

Thus, if we let t = 10, then the algorithm KARGER-STEIN runs in time $O(n^2 \log^2 n)$ and returns a min-cut of G with a probability at least $1 - 1/e^{10} > 0.9999$.

Theorem 2.3 For any fixed constant $\epsilon > 0$, the algorithm KARGER-STEIN can be implemented to run in time $O(n^2 \log^2 n)$ and returns a min-cut of the input graph G with a probability larger than $1 - \epsilon$.

PROOF. For a given $\epsilon > 0$, take a positive integer t_0 such that $1/e^{t_0} < \epsilon$: for instance, pick the smallest positive integer t_0 such that $t_0 > \ln(1/\epsilon)$. Note that t_0 is a constant that depends on ϵ but is independent of the input size n. Now replace t in the algorithm Karger-Stein with t_0 . By Theorem 2.2, the algorithm returns a min-cut of the input graph G with a probability at least $1 - 1/e^{t_0} > 1 - \epsilon$.

Since picking a random edge and contracting it in a graph of n vertices can be done is time O(n), the algorithm CONTRACTION-II runs in time $O(n^2 \log n)$. Thus, the algorithm KARGER-STEIN with the above selected t_0 runs in time $O(t_0n^2 \log^2 n) = O(n^2 \log^2 n)$, the equality is because that t_0 is a constant independent of n. \Box