

## Lecture # 10

Lecturer: Debmalya Panigrahi

Scribe: Brandon Fain

## 1 Overview

Last time we saw how Nagamochi and Ibaraki were able to construct a graph sparsifier in  $O(m)$  time which they then used to compute min-cuts in  $O(mn)$  time, still the best deterministic min-cut algorithm. However, the randomized contraction algorithm we saw earlier came after this as an attempt to make a fast algorithm that can be run in parallel. Today we revisit the randomized contraction technique for computing min-cuts.

## 2 Recursive Contraction

**Definition 1.** Given an undirected graph  $G = (V, E)$ , with edge costs  $c_e$ , a min-cut is a partition of the vertices  $S, G/S$  such that  $\emptyset \neq S \neq G$  and the cost of the cut edges is minimized. A cut edge is an edge  $e = (u, v)$  such that  $u \in S$  and  $v \notin S$ .

The problem is to find a fast randomized algorithm to compute min-cuts with high probability. The approach that Karger and Klein take is to just contract edges randomly as before, but rather than repeating the entire process many times to get high probability we repeat the first half of contractions fewer times than the last half. In fact, we do this recursively. This works out since we are less likely to make a mistake in the initial contraction steps.

The way to think about the algorithm is in terms of a branching recursion. You run the algorithm twice in parallel until each has  $n/\sqrt{2}$  vertices remaining. Then, for each of those intermediate contracted graphs, you branch twice and contract in each instance until there are  $n/2$  nodes in each. You continue this until you reach the leaves of the branching tree, and return the best min cut you find. Karger and Klein formalize this intuition as [KS96]:

---

### Algorithm 1 Recursive Contraction

---

```

1: procedure RECURSIVECONTRACT( $G, n$ )
2:   if  $G$  has fewer than 6 vertices then
3:      $G' \leftarrow \text{Contract}(G, 2)$ 
4:     return weight of min-cut in  $G'$ 
5:   else
6:     repeat
7:        $G' \leftarrow \text{Contract}(G, n/\sqrt{2})$ 
8:        $G'' \leftarrow \text{RECURSIVECONTRACTION}(G, n/\sqrt{2})$ 
9:     until done twice
10:    return weight of smaller of two results
11:   end if
12: end procedure

```

---

Where the subroutine  $\text{CONTRACT}(G, n)$  is defined as:

---

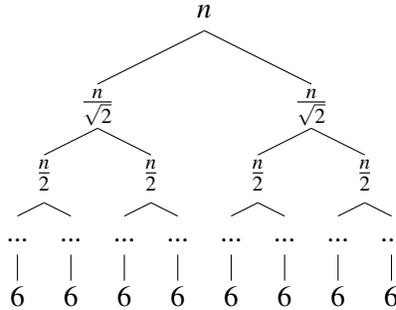
**Algorithm 2** Contraction

---

```
1: procedure CONTRACT( $G, n$ )
2:   repeat
3:      $x \leftarrow e \in G$  uniformly at random
4:      $G \leftarrow G/e$ 
5:     endpoints of  $e$  are collapsed, preserving outgoing edges
6:   until  $G$  has  $n$  vertices
7: end procedure
```

---

Running this algorithm results in a branching tree structure of recursion looking like this:



**Lemma 1.** RECURSIVECONTRACT( $G, n$ ) runs in  $\tilde{O}(n^2)$  time.

*Proof.* The running time is straightforward: one level of recursion involves two invocations of CONTRACT( $G, n/\sqrt{2}$ ). A run of CONTRACT( $G, n/\sqrt{2}$ ) takes  $n^2/\sqrt{2} = O(n^2)$  time. Thus, we have the following recurrence:

$$\begin{aligned} T(n) &= 2 \left( n^2 + T(n/\sqrt{2}) \right) \\ T(n) &= 2 \left( n^2 + 2 \left( (n/\sqrt{2})^2 + T(n/2) \right) \right) \\ T(n) &= O(n^2 + \log n) = \tilde{O}(n^2) \end{aligned}$$

□

**Lemma 2.** RECURSIVECONTRACT( $G, n$ ) returns a min-cut with probability  $1/\log n$ .

*Proof.* When we considered the standard random contraction algorithm previously, we saw that a given cut is preserved by CONTRACT( $G, k$ ) with probability at least  $\Omega((k/n)^2)$ . In each step of the recursive algorithm, we go from  $n$  to  $n/\sqrt{2}$  so this gives us a probability at least  $\Omega(1/2)$  at each step. From this we can get the following recurrence relationship on the success probability  $P(n)$ :

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

Which can be solved to give the lemma statement. □

Given the two lemmas, it is then trivial to complete the argument. Run the algorithm  $\log^2 n$  times and output the smallest cut from among all of the calls. With high probability, this is a min-cut. The formal statement is given in the theorem below:

**Theorem 3.** *A min-cut can be found in  $\tilde{O}(n^2)$  time with high probability.*

*Proof.* The runtime is just given by Lemma 1 and the high probability is just given by boosting the result of Lemma 2 with  $\log^2 n$  calls.  $\square$

### 3 Near-linear time min-cut

Briefly at the end of lecture we discussed the extensions to this work. In 1991, Gabow found an algorithm to compute a global min cut in  $\tilde{O}(m\lambda)$  using the dual notion of global min cut - spanning tree packing (similar to how st-paths are the dual notion of st-min cut) [Gab91]. Given this, it seems that we should be able to use our previous linear time sparsification result to run Gabow's algorithm in time  $O(m\log(n)) = \tilde{O}(m)$ , a near linear algorithm!

However, this turns out not to be so straightforward. Karger successfully made the argument that this is possible in 2000, but he needed to argue that the min cut on the sparsified sample corresponds to the min cut in the original graph. We very briefly noted that he did so by an averaging argument over the packing trees computed in Gabow's algorithm and thus arrived at a near linear time algorithm for min cut [Kar00].

### 4 Summary

In this lecture we covered how to use a randomized edge-contraction scheme to derive high probability (Monte-Carlo)  $\tilde{O}(n^2)$  time algorithm for min-cut that can be executed in parallel, based on the work of Karger and Klein. Then, we briefly noted that Gabow found a  $\tilde{O}(m\lambda)$  algorithm using the dual notion of packing spanning trees, which Karger was able to extend using sparsification to a near linear  $\tilde{O}(m)$  algorithm for min cut.

### References

- [Gab91] Harold N. Gabow. A matroid approach to finding edge connectivity and packing arborescences. In *Proceedings of the Twenty-third Annual ACM Symposium on Theory of Computing, STOC '91*, pages 112–122, New York, NY, USA, 1991. ACM.
- [Kar00] David R. Karger. Minimum cuts in near-linear time. *J. ACM*, 47(1):46–76, January 2000.
- [KS96] David R. Karger and Clifford Stein. A new approach to the minimum cut problem. *J. ACM*, 43(4):601–640, July 1996.