Randomized Algorithms 2013A
Lecture 7 – Global Minimum Cuts and Edge Sparsification*

Robert Krauthgamer

We move on to study graph algorithms, and may possibly return later in the course to sketching and streaming algorithms.

1 Global Minimum Cut

Input: an undirected and connected graph $G = (V, E)$ with positive edge capacities (weights) $\mathrm{cap} : E \to \mathbb{R}_+$.

Goal: find S that minimizes $\mathrm{cap}(S, \bar{S}) := \sum_{(u,v)\in(S,\bar{S})} \mathrm{cap}(u, v)$.

One method is to compute the minimum st-cut for two vertices $s, t \in V$ on opposite sides of the minimum cut. Since we do now know s, t, we can try at most $n - 1$ different choices (rather than n^2) for them.

We will see now a different approach that is randomized and can be made to work faster. The idea is that each edge gives a "weak signal" that its endpoints should be on same side of the cut. The problem is to combine these signals.

Algorithm A:

1. while G has more than 2 vertices
2. pick a random edge with probabilities proportional edge capacities
3. contract the chosen edge // it's not necessary to remove self-loops
4. report the edges between the two final vertices

Illustration: two cliques connected by one (or a few) edges.

Theorem 1 [Karger 1993]: Algorithm A reports the minimum cut in an n-vertex graph with probability at least $2/n^2$.

The proof was seen in class, and is based calculating the relevant probability directly.

*These notes summarize the material covered in class, usually skipping proofs, details, examples and so forth, and possibly adding some remarks, or pointers. The exercises are for self-practice and need not be handed in. In the interest of brevity, most references and credits were omitted.

Theorem 2: Reporting the minimum cut among n^2 independent repetitios of algorithm A outputs the minimum cut in the graph with probability at least 0.8.

The proof is straightforward, as seen in class.

Corollary 3: For every connected graph, the number of distinct cuts attaining the minimum value is $\leq n(n-1)/2$.

As discussed in class, this is because the proof above actually lower bounds the probability of reporting any particular minimmum cut, and these are disjoint events.

Exer: Show that this bound is tight.

Exer: Speed up the overall algorithm by more "careful" repetitions; for simplicitly, assume a basic operation here like picking and contracting an edge takes $O(1)$ time. (Hint: notice that we need more repetitions for the later iterations, because the probability of making errors there is much larger.)

2 Graph Sparsification for Cuts I

Input: an unweighted graph $G = (V, E)$ (i.e., with unit capacities).

Goal: we want to approximate all cuts within factor $1 \pm \varepsilon$, by creating a sparser graph G' that has (approximately) the same cut values.

We shall actually allow parallel edges, i.e., let G be a multi-graph, and we can thus actually handle "small" weights. For two graphs on the same vertex-set, we write $G \leq G'$ if

$$\forall S \subset V, \qquad \mathrm{cap}_G(S, \bar{S}) \leq \mathrm{cap}_{G'}(S, \bar{S}).$$

Applications: Smaller storage and potentially faster computation of min/max cut problems.

First attempt – subsampling:

Let's sample (i.e. keep) every edge independently with probability $p \in [0, 1]$. Denote the resulting graph $G' = (V, E')$. Consider a cut $(S, \bar{S})$, and suppose it's capacity in G is $c := \mathrm{cap}_G(S, \bar{S})$. Denote the capacity of the corresponding cut in G' by a random variable $c' := \mathrm{cap}(S, \bar{S})$. Then

$$\mathbb{E}[c'] = pc.$$

So in expectation, cuts are preserved up to scaling by a factor of $1/p$. This can be "corrected" by giving every sampled edge capacity $1/p$. But is c' likely to be close to its expectation?

Analysis of subsampling: Using the Chernoff concentration bound,

$$\Pr[c' > (1 + \varepsilon)\mathbb{E}c'] \leq e^{-\varepsilon^2 pc/3}.$$

Suppose we make sure that $p \geq \frac{3d \ln n}{\varepsilon^2 \hat{c}} \geq \frac{3d \ln n}{\varepsilon^2 c}$ for some fixed $d > 0$ (say $d = 5$); then the RHS is $\leq 1/n^d$. And since a similar bound applies to deviation in the other direction, we get

$$\Pr[c' \notin (1 \pm \varepsilon)\mathbb{E}c'] \leq 2/n^d.$$

But is it possible to guarantee this approximation to *all* cuts (recall there are about 2^n cuts)? Turns out the answer is yes...

Theorem 5 [Karger]: Let G be a graph on n vertices and minimum cut capacity $\hat{c}$. Construct G' by including every edge from G with probability $1 \geq p \geq \frac{6(d+2)\ln n}{\varepsilon^2 \hat{c}}$. Then with probability $\geq 1 - O(1/n^d)$, every cut in G' has capacity within $1 \pm \varepsilon$ factor from its expectation.

Illustration: consider applying to a clique where $\hat{c} = \Theta(n^2)$, and to a graph with two cliques connected by one edge where $\hat{c} = 1$.

The proof was seen in class. The main idea is that by Corollary 4, the number of small cuts is not too large. We can then apply several "smaller" union bounds, each with number of events (cuts) that is inversely proportional to their probabilities.

Remark: The expected number of edges in G' is $p|E|$, and thus WHP it is $\Theta(p|E|)$.

Exer: Let c_{max} denote the value of a maximum cut in the graph G. Prove that the expected number of edges in G' is $\Theta(\frac{1}{\varepsilon^2}\frac{c_{max}}{\hat{c}} \cdot \log n)$. (Hint: prove that $|E|/2 \leq c_{max} \leq |E|$ by considering a random bipartition.)

Theorem 6 [Karger]: Let H be an n-vertex graph and $X_e \in [0, M]$ for $e \in E(H)$ be independent random variables. Let $H(X_e)$ be a random graph obtained from G by placing edge weights equal to X_e, and denote by $\hat{c}$ the minimum expected capacity over all cuts in $H(X_e)$. Then with probability $\geq 1 - O(1/n^d)$, we have every cut in $G(X_e)$ has capacity within $1 \pm \tilde{\varepsilon}$ factor of its expectation, for $\tilde{\varepsilon} = \sqrt{2(d+2)(M/\hat{c})\ln n}$.

This is a slightly more general version of our Theorem 5, that will be used below.

Exer: Prove this theorem (similarly to the previous one).

3 Graph Sparsification for Cuts II

The downside of the above result is that the number of edges might not decrease at all. For instance, if the initial graph is two cliques connected by a single edge, we actually need to "sample down" each clique separately (perhaps at different rates, if they have different sizes), but not the entire graph at the same rate.

We now aim to overcome this.

Theorem 7 [Benczur-Karger, 1996]:

For every weighted graph $G = (V, E)$ on n vertices and error parameter $\varepsilon > 0$, there is a weighted subgraph $G' = (V, E')$ with $O(\varepsilon^{-2} n \log n)$ edges such that $G' \in (1 \pm \varepsilon)G$. Moreover, G' can be constructed in $O(|E| \log^2 n)$ time.

Such a graph G' is called a $(1 + \varepsilon)$-cut sparsifier.

Remark: The number of edges was later improved to $|E'| \leq O(\varepsilon^{-2} n)$ by [Batson-Spielman-Srivastava, 2009], but without near-linear time computation.

Exer: Suppose that for an input graph $G = (V, E)$ with edge capacities and k vertices $t_1, \ldots, t_k \in V$, we want to find k-way partition $V = V_1 \cup \ldots \cup V_k$ that separates all the t_i's (meaning $t_i \in V_i$ for all $i \in [k]$) and has minimum total capacity (defined as $\sum_{i<j} \mathrm{cap}(V_i, V_j)$). Prove or Disprove: If G' is a $(1+\varepsilon)$-cut sparsifier of G, then solving this problem on G' gives a $(1 \pm \varepsilon)$-approximation for its value on G.

Next week we will prove a slightly weaker version, for unweighted graphs, with another $\log^2 n$ factor, and without the near-linear time algoritm.