Rounding Techniques in Approximation Algorithms

Lecture 3: Independent Randomized Rounding for Connectivity

Lecturer: Nathan Klein *Scribe: Cheng-Hao Fu*

1 Overview of Cuts and Connectivity

The focus of this lecture is finding *cheap, well connected graphs.* Throughout, we will work with multigraphs, in which an edge can appear multiple times.

Definition 1.1 (k-Edge-Connectivity)**.** *A graph $G = (V, E)$ is k-edge-connected if $|\delta(S)| \geq k$ for all $S \subset V$.*

The following theorem is a key fact about k-edge-connected graphs that we will prove in a later lecture.

Theorem 1.2 (Nash-Williams [NW61])**.** *Every k-edge-connected (multi-)graph contains $\frac{k}{2}$ edge disjoint spanning trees.*

We briefly note that this bound is tight for all k, as a cycle with $k/2$ copies of each edge is k-edge-connected but clearly contains no more than $k/2$ edge disjoint spanning trees.

1.1 The Number of α-Near Minimum Cuts

Definition 1.3 (α-Near Minimum Cut)**.** *Given a k-edge-connected graph and $\alpha \geq 1$, a set $S \subset V$ is an α-near minimum cut if $|\delta(S)| \leq \alpha \cdot k$.*

An important theorem of Karger is as follows.

Theorem 1.4 (Number of α-Near Minimum Cuts)**.** *In a graph with n vertices there are at most $n^{2\alpha}$ α-near minimum cuts for all $\alpha \geq 1$.*

This is often proved by analyzing a randomized edge contraction algorithm. Here, we will show a different proof using Theorem 1.2. We will give a simple proof that shows a bound of $2n^{4\alpha}$, but it can be strengthened to obtain $n^{2\alpha}$ with a slightly more careful argument.

Proof. We start with a k-edge connected graph G. Applying Theorem 1.2 gives us a set of $\frac{k}{2}$ edge disjoint spanning trees $\mathcal{T} = T_1, ..., T_{k/2}$.

We now make the following two claims:

Claim 1.5. *Any cut S in G with $|\delta_G(S)| \leq \alpha k$ must have at least $\frac{k}{4}$ trees $T_i \in \mathcal{T}$ such that $|\delta_{T_i}(S)| \leq 4\alpha$. Note that $\frac{k}{4}$ trees is half of the size of $\mathcal{T}$.*

Suppose that the claim is false. That means that fewer than half ($\frac{k}{4}$) of the trees in $\mathcal{T}$ have at most 4α edges. This means that more than half of the trees must have more than 4α edges in the cut. Adding up the edges from all of these trees (keeping in mind each of them is unique because the trees are edge disjoint) gives us a total of $> \frac{k}{4} \cdot 4\alpha = \alpha k$ edges, a contradiction.

Claim 1.6. *A tree T has at most n^k cuts of size at most k.*

The core idea behind this proof is that a cut in a tree is uniquely defined by a selection of edges. Thus, the number of cuts with k edges exactly is the number of ways to select k edges to include in a cut, of which there are $\binom{n-1}{k}$. We can check that each selection of edges admits only one cut by noting that, given a cut set of edges F, and setting u to be on one side of the cut, the location of every vertex is then automatically decided. If the path from u to v crosses an odd number of edges in the cut, then v must be on a different side of a cut as u, and vice versa. That means the number of cuts of size at most k is $\binom{n-1}{k} + \binom{n-1}{k-1} + \cdots \leq n^k$. Thus, the claim is proved.

Given these two claims, we now define the following: $X = \{S \mid |\delta_G(S)| \leq \alpha k\}$. $Y_i = \{S \mid |\delta_{T_i}(S)| \leq 4\alpha\}$. We see that $\sum_i |Y_i| \leq \frac{k}{2} n^{4\alpha}$ via Claim 1.6. We know that $X \subset \bigcup_i Y_i$, because Claim 1.5 tells us that every cut of size at most αk must have some tree where it has size less than 4α. In fact, every $S \in X$ is in at least $\frac{k}{4}$ different Y_i. Thus, we know that $|X| \cdot \frac{k}{4} \leq \sum_i |Y_i| \leq \frac{k}{2} n^{4\alpha}$. Thus, we can conclude that the maximum possible size of X is $\frac{k}{2} n^{4\alpha}$ divided by $\frac{k}{4}$, which gives the bound of $2n^{4\alpha}$. □

On improving the bound: While for this lecture, a bound of $O(n^{4\alpha})$ is sufficient, it is interesting to consider why the stronger bound of $n^{2\alpha}$ is true. Notice that if we look at any cut S that has fewer than $k\alpha$ edges across the $\frac{k}{2}$ spanning trees, the *average* size of the cut in each tree is $\leq 2\alpha$. For simplicity's sake, let's focus on two trees T_1 and T_2. If we want the average size of the cut in these two trees to be 2α, which situation is "typical:" (i) both trees contribute around 2α or (ii) one tree contributes $4\alpha - 1$ and the other contributes only one edge? There are only $n - 1$ cuts of size 1 in a tree (as opposed to $n^{2\alpha}$ cuts of size 2α), so it's certainly (i)! Extending this intuition to $\frac{k}{2}$ trees, we should expect that in the worst case, all cuts of size approximately $k\alpha$ have about $\frac{k}{2}$ edges in every tree. But then the above analysis would work to prove $O(n^{2\alpha})$ instead.

On fractional graphs: In the next section we will apply this bound to graphs where edges have fractional values $x_e \in \mathbb{R}_{\geq 0}$ and we have the guarantee $\sum_{e \in \delta(S)} x_e \geq k$ for all $S \subset V$. We will always obtain this vector x from solving a linear program, so we can assume the values x_e are rational. Therefore, there exists an $\epsilon > 0$ so that every $x_e = r_e \epsilon$ for some $r_e \in \mathbb{Z}$. But then by creating a multigraph with r_e copies of each edge e, we obtain a $\frac{k}{\epsilon}$-edge-connected graph. Applying Karger's bound to this graph, we then can bound the number of cuts of size $\leq \alpha k$ of the original graph.

2 k-ECSS and k-ECSM

The following are two NP-Hard, well-studied problems in approximation algorithms.

Definition 2.1 (k-ECSS)**.** *The k-edge-connected spanning subgraph problem takes as input a graph $G = (V, E)$ and a weight function $c : E \to \mathbb{R}_{\geq 0}$. The output is then the graph $H = (V, F)$ such that H is k-edge-connected, and that F is minimal with respect to the function c.*

Definition 2.2 (k-ECSM)**.** *The k-edge-connected spanning multi-subgraph problem takes as input a graph $G = (V, E)$ and a function $c : E \to \mathbb{R}_{\geq 0}$. The output is then the multigraph $H = (V, F)$ such that H is k-edge-connected, and that multigraph F is minimal with respect to the function c where if an edge is used r times its cost is rc_e.*

So, the k-ECSM problem is the same problem as k-ECSS except we are allowed to take an edge as many times as we want (with each copy at the same cost).

k-ECSS is APX-Hard for all k, i.e. there is a constant $\epsilon > 0$ such that there is no better-than-$(1+\epsilon)$ approximation for any $k \geq 2$ unless P=NP [Pri11]. k-ECSM has no approximation ratio asymptotically better than $1 + O(1/k)$ unless P=NP [HKZoc]. In future lectures, we will explore stronger approximation algorithms for these two problems. Here we show that randomized rounding gives good solutions for unweighted k-ECSS and k-ECSM when $k \gg \log n$.

2.1 LPs and Randomized Rounding

We notice that the only difference between the polytope that contains feasible solutions to k-ECSS and the polyhedron that contains feasible solutions to k-ECSM is how large the edge variables are allowed to be. For that reason, we define a general polyhedron for the non-trivial constraints that we call P_{k-EC}, and distinguish between the LPs for the problem by separately enforcing the bounds on the size of variables.

Definition 2.3 (k-edge-connectivity polytope)**.**

$$P_{k-EC} = \begin{cases} \sum_{e \in \delta_G(S)} x_e \geq k & \forall \emptyset \subsetneq S \subsetneq V \\ x_e \geq 0 & \forall e \in E \end{cases}$$

We use the defined polytope in defining an integer program for the two problems. For k-ECSM, we have:

$$\text{minimize} \quad \textstyle\sum_e x_e c_e \quad \text{s.t.} \quad \begin{array}{l} x \in P_{k-EC} \\ x \in \mathbb{Z}^E_{\geq 0} \end{array}$$

For intuition, we can think of an optimal solution of this LP as being the point in the LP furthest along in the direction of the vector implied by $-c$. Of course, the integer program for the k-ECSS problem is given by replacing $\mathbb{Z}^E_{\geq 0}$ with $\{0, 1\}$.

In accordance with the framework we have discussed extensively in this class, we will now work with the appropriate LP relaxation of the problem, and use randomized rounding to output an approximate solution. We begin with the k-ECSM LP.

$$\text{minimize} \quad \textstyle\sum_e x_e c_e \quad \text{s.t.} \quad \begin{array}{l} x \in P_{k-EC} \\ x \in \mathbb{R}^E_{\geq 0} \end{array}$$

Note that we have simply changed the program to accept any positive real solutions, rather than only integer ones. The k-ECSS LP is the same LP but instead of $x \in \mathbb{R}^E_{\geq 0}$, we have $x \in [0, 1]^E$ We also note that we can construct a polynomial time separation oracle by finding the global minimum cut of the graph, so this LP can be solved in polynomial time.

Lemma 2.4 (Karger [Kar99])**.** *Let $x \in P_{k-ECSS}$. If $k \geq 32 \ln n$, with high probability after sampling every edge independently with probability x_e, the resulting graph G' is $k - 4\sqrt{k \ln n}$ connected.*

Proof. We refer to the graph with edges with weight x_e as G, and G' as in the lemma statement.

We will split the analysis into cases, and combine them in the end. These cases will be based on the size of a cut S in G.

Fix a cut S such that its size in G is between ik and $(i+1)k$ where $i \geq 2$, we want to use concentration bounds to show that the probability that this cut is small after the sampling is quite small. We note that the size of the cut in G corresponds to the expected value of the cut in G'. For these large cuts, we simply bound the probability G' has fewer than k edges.

$$\mathbb{P}\left[|\delta_{G'}(S)| \leq k\right] = \mathbb{P}\left[|\delta_{G'}(S)| \leq \left(1 - \left(1 - \frac{1}{i}\right)\right) ik\right]$$

Now we apply a Chernoff bound with $\delta = 1 - 1/i \geq 1/2$, obtaining

$$\mathbb{P}\left[|\delta_{G'}(S)| \leq k\right] \leq \exp\left(\frac{-(1-\frac{1}{i})^2 ik}{2}\right) \leq \exp\left(ik/8\right) \leq n^{-4i}$$

where we use that $k \geq 32\ln(n)$.

We note that the analysis is only for one particular cut. Thus, we have to union bound over all of the cuts of this size to see the probability that we avoid small cuts for every cut in this case. We know from Theorem 1.4 that there are at most n^{2i+2} such cuts for each i. Thus, taking the union bound gives us a final probability bound of at most $n^{-4i}n^{2i+2} = n^{-2i+2} \leq n^{-i}$ (using $i \geq 2$) for any cut in this size range to have low connectivity. Taking a union bound over all size ranges ($i \geq 2$) then gives us:

$$\mathbb{P}\left[\text{Any cut of size at least } 2k \text{ has fewer than } k \text{ edges in } G'\right] \leq \sum_{i \geq 2}^{\infty} n^{-i} \leq 2n^{-2}$$

All that is left now is to handle cuts of value between k and $2k$. Using $\delta = 4\sqrt{\ln n/k}$ and a Chernoff bound, we obtain (for a fixed cut S):

$$\mathbb{P}\left[|\delta_{G'}(S)| \leq k - 4\sqrt{k\ln n}\right] \leq \mathbb{P}\left[|\delta_{G'}(S)| \leq (1-\delta)k\right] \tag{1}$$

$$\leq \exp\left(\frac{-\delta^2 k}{2}\right) \leq n^{-8} \tag{2}$$

Where we use that k is a lower bound on the expectation of any cut in this class. There are at most n^4 such cuts, so the union bound gives us a final probability of n^{-4} that some cut of size at most $2k$ has connectivity below our bound. Thus, union bounding over the two classes of bad events, with probability at least $1 - 1/n$ the graph is $k - 4\sqrt{k\ln n}$ connected, as desired. □

2.2 Approximation Algorithms for k-ECSS and k-ECSM

We now show how the above lemma leads to a $1 + O(\sqrt{\frac{\log n}{k}})$ approximation for k-ECSM and unweighted k-ECSS for sufficiently large k. While it is possible to make this algorithm work for smaller k with the same guarantee, both problems have easy 2-approximations, so the claimed ratio is only of interest for large enough k.

Theorem 2.5. *There is a randomized $1 + O(\sqrt{\frac{\log n}{k}})$ approximation algorithm for k-ECSM and unweighted k-ECSS for $k \geq 32\ln n$.*

Proof. We solve the corresponding LP, and independently round each edge to 1 with probability x_e. In the k-ECSM case, for any $x_e \geq 1$, then we can define it to be $\lceil x_e \rceil$ different Bernoulllis. All of them except for the last one will have probability 1 of success, and the last one will simply have probability of success equal to the remaining probability mass. This ensures that the expected value of all cuts stays the same and the analysis from the above theorem can still hold.

We're left with a $k - O(\sqrt{k \log n})$ connected graph with high probability. We also note that the expected cost of this is the same as the cost of the optimum solution to the LP problem, which is at most the cost of the optimum solution to the integer problem.. We now need to increase the connectivity back up to k. We do this in one of two ways, depending on which problem we are solving.

For k-ECSM, we find an MST in the graph, and add $O(\sqrt{k \log n})$ copies of this tree into the graph. By Nash-Williams, we know that the optimal solution uses edges that can be partitioned into $\frac{k}{2}$ edge disjoint spanning trees, and so we know that taking an MST in the graph must be at most the value of one of these trees. Thus, we have:

$$c(\text{MST}) \leq \frac{2}{k}(OPT)$$

Which means that these extra edges add weight at most $O(\sqrt{\frac{\log n}{k}})$ of the optimal solution. So our expected cost is at most $(1 + O(\sqrt{\frac{\log n}{k}})) \cdot OPT$, as desired.

For k-ECSS, we would like to do the same thing - adding edges from spanning trees to achieve the desired connectivity. The first problem we run into is that we cannot simply add the same spanning tree over and over again as in the above, as k-ECSS does not allow for edges to be used more than once. Furthermore, we cannot even guarantee the existence of any spanning tree that is disjoint from any edges we had before - we may have used edges such that the graph remaining after our selection of edges from the LP is no longer connected. However, we can get around this issue with the simple observation that if the remaining graph is not connected, the disconnected sections must already have at least k edges across the cut. Thus, it suffices to take a spanning forest F of the remaining graph, and add it to the existing graph. We then take a spanning forest of the remaining graph after the inclusion of F, and so on. Each instance of adding the forest will thus increase the connectivity by at least 1, and so we add at most $O(\sqrt{k \log n})$ forests.

If we are dealing with the *weighted* version of k-ECSS (which is APX-Hard for all k) this strategy cannot work, as we cannot bound the cost of these forests in terms of the cost of OPT. However, if we are solving the unweighted variant (i.e. all edges have cost 1) then we know OPT uses at least $\frac{nk}{2}$ edges. So, we can bound the additional cost by n times the number of additional forests, or $O(n\sqrt{k \log n})$ edges. This gives us the desired guarantee of $1 + O(\sqrt{\frac{\log n}{k}})$. □

References

[HKZoc] D Ellis Hershkowitz, Nathan Klein, and Rico Zenklusen. *Ghost Value Augmentation for k-Edge-Connectivity*. STOC 2024 (cit. on p. 3).

[Kar99] D. Karger. "Random sampling in cut, flow, and network design problems". In: *Math OR* 24 (1999), pp. 383–413 (cit. on p. 3).

[NW61] C. St. J. A. Nash-Williams. "Edge disjoint spanning trees of finite graphs". In: *J. London Math. Soc.* 36 (1961), pp. 445–45 (cit. on p. 1).

[Pri11] David Pritchard. “k-Edge-Connectivity: Approximation and LP Relaxation”. In: *Proceedings of the Ninth International Workshop on Approximation and Online Algorithms (WAOA)*. Ed. by Klaus Jansen and Roberto Solis-Oba. Springer Berlin Heidelberg, 2011, pp. 225–236. DOI: 10.1007/978-3-642-18318-8_20 (cit. on p. 3).