COMPSCI 638: Graph Algorithms September 13, 2019

Lecture 6

Lecturer: Debmalya Panigrahi *Scribe: Kevin Sun*

1 Overview

In this lecture, we give another rounding scheme for the multicut problem and bound its performance. We then study the sparsest cut problem and give an algorithm that reduces the problem to the multicut problem.

2 The Multicut Problem

Recall that the multicut problem is the following: we are given a graph $G = (V, E)$ with edge costs $c(e)$ and a set of source-sink pairs (s_i, t_i) for $i \in \{1, \ldots, k\}$. Our goal is to find a minimum-cost subset of edges whose removal separates s_i from t_i for every i.

In the linear program relaxation, we seek to find edge lengths to minimize the total volume while ensuring that the distance from each s_i to t_i is at least 1. Letting P denote the set of all s_i-t_i paths, this is the following:

$$\text{(LP): } \min \sum_{e \in E} c(e)\ell_e$$
$$\sum_{e \in p} \ell_e \geq 1 \quad \forall p \in P$$
$$\ell_e \geq 0 \quad \forall e \in E$$

Recall that we interpret the objective of this linear program as a volume minimization by considering each edge e as a pipe with width $c(e)$ and length ℓ_e.

2.1 The GVY Algorithm

Before we describe the "region-growing" algorithm of Garg et al. [GVY96], we must first establish some notation. Given a solution ℓ to (LP), let $d(x, y)$ denote the shortest distance from x to y under ℓ. Furthermore, for any $r \geq 0$, we let $S_i(r) = \{v : d(s_i, v) \leq r\}$ and for any $S \subseteq V$, we set

$$\delta(S) = \{(u, v) \in E : u \in S, v \notin S\}$$
$$I(S) = \{(u, v) \in E : u, v \in S\}.$$

For any subset F of edges, we let $c(F)$ denote the total cost of edges in F. Finally, we let V^* denote the optimal objective value achieved by (LP) and for any $i \in \{1, \ldots, k\}$ and $r \in [0, 1/2)$, we define $V_i(r)$ as the total volume of pipes within $S_i(r)$, as well as $1/k$ of the optimal volume. More specifically, we set

$$V_i(r) = \frac{V^*}{k} + \sum_{e \in I(S_i(r))} c(e)\ell_e + \sum_{\substack{e=(x,y) \in \delta(S_i(r)) \\ x \in S_i(r), y \notin S_i(r)}} c(e)(r - d(s_i, x)).$$

We will later give the reasoning behind the V^*/k term. The second term adds the volume of pipes entirely contained in $S_i(r)$, and the final term adds the portion of the volume of pipes partially contained in $S_i(r)$. We now state the key lemma behind the GVY algorithm.

Lemma 1. *For every i, there exists $r_i < 1/2$ such that $c(\delta(S_i(r_i))) \leq 2\ln(2k)V_i(r_i)$.*

And with this lemma, we are now ready to state the GVY algorithm.

Algorithm 1 The GVY Algorithm for the Multicut problem

```
Solve (LP) to obtain an optimal edge lengths
F ← ∅
for i = 1, ..., k do
    if s_i and t_i are connected in (V, E \ F) then
        Choose r_i as in Lemma 1
        F ← F ∪ δ(S_i(r_i))
        Remove S_i(r) and its incident edges from the graph
return F
```

Lemma 2. *The output of Algorithm 1 is a feasible multicut.*

Proof. For contradiction, suppose s_j is connected to t_j for some j upon the removal of F. This implies s_j and t_j were both in some ball $S_i(r)$ when the vertices of this ball were removed from the graph. But this violates the triangle inequality and feasibility of the edge lengths. □

Now before proving Lemma 1, we use it to bound the approximation ratio of Algorithm 1.

Theorem 3. *Algorithm 1 is an $O(\log k)$-approximation for the multicut problem.*

Proof. Let F_i denote the set of edges added to F in the i-th iteration of the algorithm, and let M_i denote the total volume of edges removed when we remove $S_i(r)$ and its incident edges from the graph. Then notice that

$$c(F_i) \leq 2\ln(2k)V_i(r_i) \leq 2\ln(2k)\left(M_i + \frac{V^*}{k}\right), \tag{1}$$

where the first inequality follows from Lemma 1 and the second inequality holds because M_i considers the entire volume of its edges while $V_i(r_i)$ only considers a portion of some edges. Now notice that an edge contributes to at most one M_i because an edge can only be removed from the graph at most once; this implies $\sum_{i=1}^{k} M_i \leq V^*$. With this observation, we can sum (1) across all i and conclude

$$c(F) = \sum_{i=1}^{k} c(F_i) \leq 2\ln(2k)\sum_{i=1}^{k}\left(M_i + \frac{V^*}{k}\right) \leq 4\ln(2k)V^* = O(\log k)V^*. \qquad \square$$

We end this section with a proof of Lemma 1.

Proof of Lemma 1. The key observation is that the change in the volume of $S_i(r)$, with respect to r, is precisely the capacity of $\delta(S_i(r))$. More formally, we have

$$\frac{\mathrm{d}V_i(r)}{\mathrm{d}r} = c(\delta(S_i(r))).$$

For contradiction, suppose there exists an i such that for every $r \in [0, 1/2)$, we have

$$\frac{c(\delta(S_i(r_i)))}{V_i(r_i)} = \frac{\mathrm{d}V_i(r)}{\mathrm{d}r} \cdot \frac{1}{V_i(r_i)} > \alpha,$$

where $\alpha = 2\ln(2k)$ denotes our approximation ratio. Rearranging and integrating both sides yields

$$\int_{V_0}^{V_{1/2}} \frac{1}{V_i(r_i)}\,\mathrm{d}V_i(r) > \int_0^{1/2} \alpha\,\mathrm{d}r\,,$$

where V_0 denotes the initial volume (i.e., $V_0 = V^*/k$) at each s_i, and $V_{1/2}$ denotes the volume of $S_i(r)$ when $r = 1/2$. Computing the integrals gives us $\ln(V_{1/2}/V_0) > \alpha/2$, and substituting $V_{1/2} \leq kV_0 + V^*$ gives us

$$\alpha < 2\ln\left(\frac{kV_0 + V^*}{V_0}\right) = 2\ln\left(k + \frac{V^*}{V_0}\right). \tag{2}$$

Looking at (2), we see why we had set $V_0 = V^*/k$: this is done in order to minimize the competitive ratio α. Substituting $\alpha = 2\ln(2k)$ and $V_0 = V^*/k$ into (2) yields the desired contradiction. □

3 Sparsest Cut

We transition towards studying a new problem known as the sparsest cut problem. Before formally defining this problem, let us first recall the maximum concurrent flow problem: given a graph $G = (V, E)$ with edge costs $c(e)$ and source-sink pairs (s_i, t_i) for $i = 1, \ldots, k$, find a flow that maximizes *minimum* amount of flow between every source-sink pair.

Recall that we let P_i denote the set of s_i-t_i paths, and P denote the set of all source-sink paths. The corresponding linear program is the following:

$$\text{(P): } \max \lambda$$
$$\sum_{p \in P_i} f_p \geq \lambda \quad \forall i \in \{1, \ldots, k\}$$
$$\sum_{i=1}^{k} \sum_{\substack{p \in P_i \\ p \ni e}} f_p \leq c(e) \quad \forall e \in E$$
$$f_p \geq 0 \quad \forall p \in P$$

And the dual is the following:

$$\text{(D): } \min \sum_{e \in E} c(e)\ell_e$$
$$\sum_{e \in p} \ell_e \geq \lambda_i \quad \forall i, \forall p \in P_i$$
$$\sum_{i=1}^{k} \lambda_i \geq 1$$
$$\lambda_i, \ell_e \geq 0 \quad \forall i, \forall e \in E$$

Notice that in (D), we have increased the flexibility of the constraint compared to the multicut linear program. Now, the distance between s_i and t_i must be at least $\lambda_i \geq 0$, but the sum of the λ_i must be at least one.

Now instead of viewing the problem (D) as obtaining an edge length function ℓ, we can equivalently view the variables as $d(x, y)$ for every $x, y \in V$. The variable d must be a metric, and in this case, λ_i would be the shortest s_i-t_i distance. Thus, (D) is equivalent to

$$\begin{aligned} \text{(D-SC):} \ \min & \sum_{e=(x,y)\in E} c(e)d(x,y) \\ & \sum_{i=1}^{k} d(s_i, t_i) \geq 1 \\ & d \text{ is a metric} \end{aligned}$$

The Sparsest Cut Problem: We are given a graph $G = (V, E)$ with edge costs $c(e)$. Our goal is to find $S \subset V$ that minimizes the *sparsity* of S, defined as $c(\delta(S))/p(S)$, where $p(S) = |S| \cdot |V \setminus S|$. Notice that this is a "normalized" version of (D-SC): set every $x \neq y \in V$ as a source-sink pair; given S, set $d(x, y) = 1$ if $|S \cap \{x, y\}| = 1$ and 0 otherwise; and notice $p(S) = \sum_i d(s_i, t_i)$ is the number of pairs separated by S. In general, however, (D-SC) searches over all metrics while the sparsest cut problem is restricted to metrics of the form described above.

We remark that this problem is also known as the *uniform* sparsest cut problems since all pairs are treated equally. Furthermore, in some contexts, the denominator is $\min\{|S|, |V \setminus S|\}$, but it is not hard to show that $|V|$ times this is always within a factor 2 of $|S| \cdot |V \setminus S|$, and for our purposes, losing a factor of 2 is not critical.

We now show that the optimum value of (D-SC) provides a lower bound on the sparsity of the sparsest cut. In other words, (D-SC) is indeed a relaxation of the sparsest cut problem.

Fact 4. *Let D^* denote the optimum value of (D-SC) where every $x \neq y \in V$ is a source-sink pair. Then*

$$D^* \leq \min_{S \subset V} \frac{c(\delta(S))}{p(S)},$$

where $p(S)$ denotes the number of (s_i, t_i) pairs separated by S.

Proof. Let $S \subset V$ be any cut, and consider the following solution to (D-SC): set $d(x, y) = 1/p(S)$ if $|S \cap \{x, y\}| = 1$ and 0 otherwise. Then d is a feasible solution to (D-SC), and the objective value obtained by d is precisely $c(\delta(S))/p(S)$, i.e., the sparsity of S. □

3.1 Reduction to Multicut

In this section, we give an algorithm for the sparsest cut problem by reducing it to the multicut problem. Intuitively, we will find a subset of source-sink pairs that are sufficiently far apart. We then invoke a multicut rounding algorithm on this subset to obtain a multicut solution for these pairs. Finally, we return the sparsest cut generated by this solution. Recall that for any $k \geq 1$, we let $H_k = 1 + 1/2 + \cdots + 1/k$ denote the k-th Harmonic number.

Lemma 5. *Let d^* be an optimal solution to (D-SC). Then there exists $S \subseteq \{(s_i, t_i)\}_{i=1}^k$ such that*

$$d^*(s_i, t_i) \geq \frac{1}{H_{\binom{n}{2}} \cdot |S|}$$

for every $(s_i, t_i) \in S$.

Before proving this lemma, we state the reduction algorithm and bound its performance.

Algorithm 2 Reducing Sparsest Cut to Multicut

1: $d^* \leftarrow$ optimal solution to (D-SC)
2: $S \leftarrow$ subset of pairs as in Lemma 5
3: $d' \leftarrow d \cdot H_{\binom{n}{2}} \cdot |S|$
4: Run GKY (or CKR) rounding on d' to obtain a subset of edges F
5: $X \leftarrow$ the minimum-sparsity connected component produced by F
6: **return** X

Theorem 6. *Algorithm 2 is an $O(\log^2 n)$-approximation for the sparsest cut problem.*

Proof. From Lemma 5, we know that every pair in S has distance at least 1 with respect to d', which means F is a multicut solution that separates the pairs in S. Although F is not necessarily a cut, from the performance guarantee of the rounding, we have

$$\begin{aligned} c(F) &= \sum_{e \in F} c(e) \\ &= O(\log k) \cdot \sum_{e=(x,y) \in E} c(e) d'(x, y) \\ &= O(\log k) H_{\binom{n}{2}} \cdot |S| \cdot \sum_{e=(x,y) \in E} c(e) d^*(x, y) \\ &= O(\log^2 n) \cdot |S| \cdot D^*, \end{aligned}$$

where D^* denotes the optimal value of (D-SC). Since F separates at least the pairs in $|S|$, the "sparsity" of F (recall that F is not necessarily a cut) is

$$\frac{c(F)}{p(F)} = \frac{O(\log^2 n) \cdot |S| \cdot D^*}{|S|} = O(\log^2 n) \cdot D^*.$$

Our final step is to bound the sparsity of X, and we do this by an averaging argument. Let $X_1, \ldots, X_r$ denote the connected components produced by F, and notice that

$$\frac{c(\delta(X_1)) + \cdots + c(\delta(X_r))}{p(X_1) + \cdots + p(X_r)} = \frac{2c(F)}{2p(F)} = O(\log^2 n) \cdot D^*.$$

The first equality holds in both the numerator and denominator because every edge/pair cut/separated by F is also cut/separated by exactly two connected components. By a standard averaging argument, this implies that the sparsity of X is $O(\log^2 n) \cdot D^*$, as desired. □

We now prove Lemma 5.

Proof of Lemma 5. Rename the source-sink pairs so that

$$d^*(s_1, t_1) \geq d^*(s_2, t_2) \geq \cdots \geq d^*(s_k, t_k),$$

and let $S_j = \{(s_i, t_i)\}_{j=1}^{i}$. For contradiction, assume that the lemma is false for every subset of pairs. In particular, S_1 is not satisfying, which implies

$$d^*(s_1, t_1) < \frac{1}{H_{\binom{n}{2}}}.$$

Similarly, S_2 is not a solution, which implies

$$d^*(s_2, t_2) < \frac{1}{H_{\binom{n}{2}}} \cdot \frac{1}{2}.$$

Extending this reasoning to all $k = \binom{n}{2}$ source-sink pairs, we get

$$\sum_{i=1}^{k} d^*(s_i, t_i) < \frac{1}{H_{\binom{n}{2}}} \left(1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{\binom{n}{2}}\right) = 1,$$

violating the constraint of (D-SC), so in fact, some S_j must satisfy the lemma. □

4 Summary

In this lecture, we saw the GVY region-growing technique applied to the multicut problem. We also showed the relationship between the maximum concurrent flow problem to the sparsest cut problem and gave a reduction for this problem to the multicut problem.

References

[GVY96] Naveen Garg, Vijay V Vazirani, and Mihalis Yannakakis. Approximate max-flow min-(multi) cut theorems and their applications. *SIAM Journal on Computing*, 25(2):235–251, 1996.