15-854: Approximations Algorithms **Lecturer:** Anupam Gupta
Topic: LP solutions as Metrics: MultiCut, and Region Growing **Date:** 11/16/2005
Scribe: Jonathan Derryberry

20.1 Introduction

Last time we covered the s-t MinCut problem and the Multiway Cut problem, for which we gave a $2(1-\frac{1}{k})$-approximation via an LP relaxation and showed that the integrality gap was $2(1-\frac{1}{k})$. Today we will cover the *MultiCut* problem, and approximation algorithms for the problem using a techniques called low-diameter random decomposition (LDRD) and region growing. The MultiCut problem is related to the *Sparsest Cut* problem in which a roughly balanced separator is sought in a graph. Being able to find a sparse cut can help with divide and conquer algorithms because it implies we can divide the graph into two roughly equal-sized pieces with little "interaction" (i.e., edges crossing between the two pieces). Ideas similar to the ones in today's lecture come from [1, 2, 3, 4, 5].

20.2 The MultiCut Problem

The MultiCut problem is formally defined as follows. As input, we have an undirected graph $G = (V, E)$ where each edge e has capacity c_e and $|V| = n$ and $|E| = m$. We also have a set $T = \{(s_i, t_i) \mid s_i, t_i \in V\}$ of k terminal pairs. The goal is to find a set $E' \subseteq E$ of minimum cost such that $G - E'$ has no s_i-t_i path for $1 \le i \le k$. The cost of E' is defined as $c(E') = \sum_{e \in E'} c_e$.

20.2.1 The MultiCut Problem is NP-hard

To get some intuition for why this is a hard problem, we give a simple reduction from Vertex Cover to the MultiCut problem, even when the MultiCut problem has its input graph restricted to being a star and capacities are all set to 1 (implying that the MultiCut problem on general graphs and capacities is also hard).

Given an instance $G_{\mathrm{VC}} = (V_{\mathrm{VC}}, E_{\mathrm{VC}})$ of the Vertex Cover problem, create an instance $(G_{\mathrm{MC}}, T_{\mathrm{MC}})$ of the MultiCut problem as follows. Let $G_{\mathrm{MC}} = (V_{\mathrm{MC}}, E_{\mathrm{MC}})$, where $V_{\mathrm{MC}} = V_{\mathrm{VC}} + \chi$ where χ is disjoint from V_{VC} and will serve as the center of a star. Let $E_{\mathrm{MC}} = \{(\chi, v) \mid v \in V_{\mathrm{VC}}\}$, and $c_e = 1\ \forall e \in E_{\mathrm{MC}}$. Finally, let $T_{\mathrm{VC}} = \{(u, v) \mid (u, v) \in E_{\mathrm{VC}}\}$.

To see that minimizing the cost of the solution to this MultiCut problem is the same as minimizing the size of a vertex cover of G_{VC}, note that for each edge $(u, v) \in E_{\mathrm{VC}}$, either the edge (χ, u) or (χ, v) must be cut at a cost of 1; this is, in different notation, precisely the vertex cover problem, where each edge must be covered by vertex u or v at a cost of 1 to the size of the vertex cover.

20.2.2 LP Formulation of the MultiCut Problem

Now that we know that the MultiCut problem is NP-hard, we try to devise an approximation algorithm for it. We begin by formulating the MultiCut problem as an LP as follows. Given an instance (G, T) of the MultiCut problem with edge capacities c_e, minimize $\sum_{e \in E} c_e \ell_e$ subject to the constraint that $\ell(p) \geq 1$ for all paths p from s_i to t_i for all $(s_i, t_i) \in T$. We also require $\ell_e \geq 0 \; \forall e$. Essentially, ℓ_e can be thought of as the length of edge e.

The constraints can be thought of as ensuring that all paths from an s_i to its corresponding t_i have length at least 1. If edge lengths were restricted to $\{0, 1\}$ then this ensures that s_i is "cut" from t_i in the sense that there is at least one long edge that must be crossed on the way to t_i. Thus, the cost of this LP is at most the cost of an optimal solution to the MultiCut problem.

Notice that minimizing $\sum_{e \in E} c_e \ell_e$ can be thought of as minimizing the total volume of the graph (subject to the above constraints) if c_e is considered to be the cross-sectional area of edge e.

Also notice that the number of constraints in this LP is exponential, but if we wanted we could devise a polynomial-sized LP by creating distance variables $d_{(v,i)}$ for each $v \in V$ and $1 \leq i \leq k$, setting $d_{(s_i,i)} = 0$, requiring $d_{(t_i,i)} \geq 1$, and requiring $d_{(v,i)} \leq d_{(u,i)} + \ell_{(u,v)}$ for all $(u, v) \in E$.

20.2.3 Using a Low-Diameter Random Decomposition to Round LPMC

To make use of the optimal solution ℓ^* to the linear program for MultiCut (LPMC), we have the following theorem.

Theorem 20.2.1 *Given any solution ℓ to LPMC with fractional cost z, there exists a feasible integer solution E_ℓ with cost $c(E_\ell) \leq O(\log n) \cdot z$.*

To describe how to achieve such an integer solution, we use the concept of a low-diameter random decomposition (LDRD) of a graph, which was previously introduced in Lecture 12, where Claim 12.3.1 states the key result. We re-iterate a few aspects of LDRD's below.

Given any graph $G = (V, E)$ with metric edge lengths ℓ_e and a parameter $\delta \geq 0$, we can find a partition $(C_1, \ldots, C_{k'})$ of V (note that k' does not have to be the same as k above) such that the following two properties hold:

1. $G[C_i]$ has diameter at most δ (recall that $G[C_i]$ is the subgraph of G induced by C_i)
2. $\mathbf{Pr}[\text{edge } e \text{ is "cut"}] \leq O(\log n)\ell_e/\delta$, where "cut" means that the endpoints of e lie in different C_i's.

Such a partition $(C_1, \ldots, C_{k'})$ is called a *low-diameter decomposition* of the graph, and we can use randomness to produce such a decomposition in which each edge is cut with the specified probability.

Notice that if $\delta = 1 - \varepsilon$ for any positive ε, then no s_i lies in the same cluster C_j as its corresponding t_i because s_i is at least distance 1 from t_i. The following lemma suffices to prove Theorem 20.2.1:

Lemma 20.2.2 *If E_ℓ is the set of edges cut by an LDRD with $\delta = 1 - \varepsilon$, then $\mathbf{E}[cost(E_\ell)] \leq \frac{O(\log n)}{1-\varepsilon} \cdot \sum_{e \in E} c_e \ell_e \leq O(\log n) \cdot z$.*

The proof of Lemma 20.2.2 is immediate from the definition of an LDRD, so to round LPMC solution ℓ, we simply find an LDRD for G using edge lengths ℓ and parameter $\delta = 1 - \varepsilon$ for some small ε, and we cut the edges that are cut in the LDRD. For details on how to actually find an LDRD, see Lecture 12, Section 12.3.

20.2.4 Deterministically Rounding LPMC Using the Region Growing Theorem

In this section, we present a different technique for rounding LPMC. In doing so, we remove the randomness inherent in the LDRD rounding scheme in the previous section, and improve the approximation ratio from $O(\log n)$ to $O(\log k)$, where k is the number of terminal pairs.

First, recall the alternative view of LPMC in which each edge cost c_e is thought of as the cross-section area of edge e and ℓ_e represents the length of e, so that $c_e\ell_e$ represents the "volume" of e and the volume of G is $\sum_{e\in E} c_e\ell_e$. In this view, our goal will be to grow a region around each s_i where each edge leaving a region is cut, and the sum of the cross-sectional areas of cut edges is only a factor of $O(\log k)$ more than the total volume of G.

To aid our argument, we introduce some new notation. Let $\mathrm{Vol}(G)$ represent the volume of graph G (i.e., $\sum_{e\in E} c_e\ell_e$). Let $f(x,(u,v),\delta_1,\delta_2)$ for $0 < \delta_1 < \delta_2$ represent the "fraction" of edge (u,v) that is between distance δ_1 and δ_2 of v in G. In defining f we also assume $d_\ell(x,u) \le d_\ell(x,v)$, where $d_\ell(x,y)$ is the distance from x to y in graph G using edge lengths ℓ. More formally,

$$f(x,(u,v),\delta_1,\delta_2) = \begin{cases} 1 & \text{if } \delta_1 \le d_\ell(x,u) \le d_\ell(x,v) \le \delta_2 \text{ and } \delta_1 < d_\ell(x,v) \\ 0 & \text{if } d_\ell(x,u) \le d_\ell(x,v) \le \delta_1 \le \delta_2 \\ 0 & \text{if } \delta_1 \le \delta_2 \le d_\ell(x,u) \le d_\ell(x,v) \text{ and } \delta_2 < d_\ell(x,v) \\ \frac{d_\ell(x,v)-\delta_1}{d_\ell(x,v)-d_\ell(x,u)} & \text{if } d_\ell(x,u) < \delta_1 \le d_\ell(x,v) \le \delta_2 \\ \frac{\delta_2-d_\ell(u,v)}{d_\ell(x,v)-d_\ell(x,u)} & \text{if } \delta_1 \le d_\ell(x,u) \le \delta_2 < d_\ell(x,v) \\ \frac{\delta_2-\delta_1}{d_\ell(x,v)-d_\ell(x,u)} & \text{if } d_\ell(x,u) < \delta_1 \le \delta_2 < d_\ell(x,v). \end{cases}$$

Now, let $\mathrm{Vol}(v,[\delta_1,\delta_2])$ be defined as

$$\mathrm{Vol}(v,[\delta_1,\delta_2]) = \sum_{e\in E} c_e\ell_e f(v,e,\delta_1,\delta_2).$$

Note that $\mathrm{Vol}(v,[\delta_1,\delta_2])$ represents the total volume of edges that exists between distances δ_1 and δ_2 from node v. Next, we define $B_x(r)$ as

$$B_x(r) = \{y \in V \mid d_\ell(x,y) \le r\},$$

and define $\partial(S)$ to be the set of edges in E that have one end point in S. Finally, for $E' \subseteq E$ let $c(E') = \sum_{e\in E'} c_e$ represent the total capacity of edge-set E'. See Figure 20.2.1 for a depiction of some of this terminology as well as some of the terminology used in the proof of Theorem 20.2.3.

With these definitions in hand, we have the following theorem, which will enable us to create a rounding scheme for LPMC.

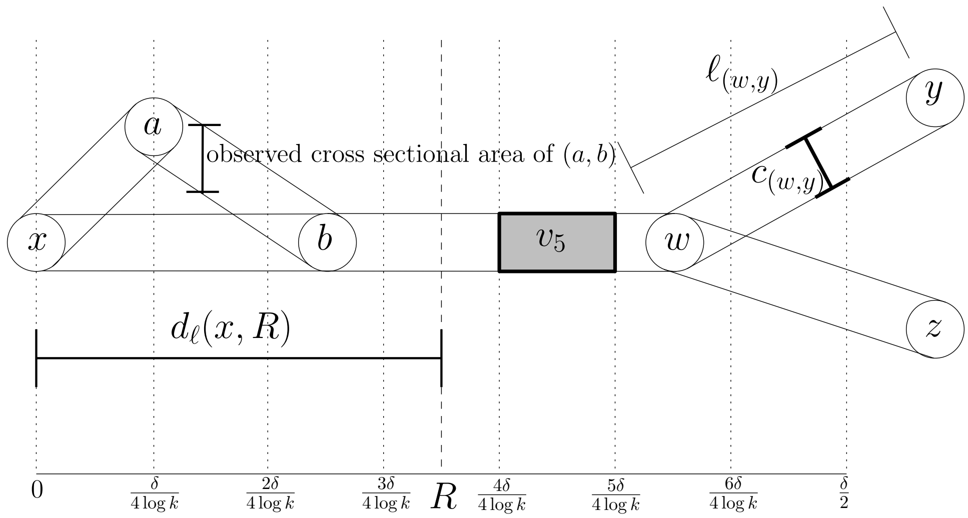

Figure 20.2.1: A visual explanation of various terminology used in the Region Growing Theorem and its proof. Notice that edge (a,b) is "slanted" so that its "observed cross-sectional area" is greater than $c_{(a,b)}$.

Theorem 20.2.3 (Region Growing Theorem) *Given graph $G = (V, E)$, metric edge lengths ℓ_e, edge capacities $c_e > 0$, vertex $x \in V$, diameter bound δ, and parameter k*[1]*, there exists a value $R \leq \delta/2$ such that $c(\partial(B_x(R)) \leq \frac{\mathrm{Vol}(G)}{k} + \frac{O(\log k)\mathrm{Vol}(x,[0,R])}{\delta}$.*

Proof: For $v_1, \ldots, v_{2\log k}$, let $v_i = \mathrm{Vol}(x, [\frac{(i-1)\cdot\delta}{4\log k}, \frac{i\cdot\delta}{4\log k}])$. Intuitively, the v_i's partition the volume of the ball of radius $\delta/2$ around x into concentric shells.

Suppose there exists $R \leq \delta/2$ such that $c(\partial(B_x(R)) \leq \frac{\mathrm{Vol}(G)}{k(\delta/2)}$. If so, we can use this R to satisfy the claim of the theorem. Else, we know that $v_1 \geq \frac{\mathrm{Vol}(G)}{k(\delta/2)} \cdot \frac{\delta}{4\log k} = \frac{\mathrm{Vol}(G)}{2k\log k}$ because the width of volume v_1 is $\frac{\delta}{4\log k}$ and the "observed cross-sectional area"[2] at any radius $R \in (0, \frac{\delta}{4\log k}]$ is more than $\frac{\mathrm{Vol}(G)}{k(\delta/2)}$ because we know $c(\partial(B_x(R)) > \frac{\mathrm{Vol}(G)}{k(\delta/2)}$ for all such R.

Notice that there exists v_i such that $v_i \leq v_1 + \cdots + v_{i-1}$. To see this, suppose there does not exist such a v_i. Then $\sum_{1\leq j\leq i'} v_j \geq v_1 \cdot 2^{i'-1}$ so that $\sum_{1\leq j\leq 2\log k} v_j \geq 2^{2\log k - 1} v_1 \geq \frac{k^2}{2} \cdot \frac{\mathrm{Vol}(G)}{2k\log k} > \mathrm{Vol}(G)$, which is a contradiction.

Now, if we divide this v_i by its width, we get its average observed cross-sectional area, averaging over $R \in [\frac{(i-1)\cdot\delta}{4\log k}, \frac{i\cdot\delta}{4\log k}]$. The average observed cross-sectional area is an upper bound on

[1]Note that k does *not* have to be the same as the number of terminal pairs, but we re-use the character k here to foster intuition for how this theorem will be used because we will later set the parameter k equal to the number of terminal pairs.

[2]Here, by "observed cross-sectional area" we mean the rate at which $\mathrm{Vol}(x, [0, R])$ increases as R increases. This is at least $c(\partial(B_x(R)))$, and is even larger if some of the edges in $\partial(B_x(R))$ are "slanted," where edge (u, v) is slanted if $\ell_{(u,v)} > d_\ell(x, v) - d_\ell(x, u) \geq 0$.

$\min_{R \in [\frac{(i-1)\cdot\delta}{4\log k}, \frac{i\cdot\delta}{4\log k}]} c(\partial(B_x(R)))$, so we can find a cut of value at most

$$\frac{v_i}{\delta/(4\log k)} \leq \frac{v_1 + \cdots + v_{i-1}}{\delta/(4\log k)} \leq \frac{\mathrm{Vol}(x,R)}{\delta/(4\log k)} = \frac{O(\log k)\mathrm{Vol}(x,[0,R])}{\delta},$$

satisfying the claim of the theorem. ∎

The high level idea of the proof of Theorem 20.2.3 is that either there is a small cut or there is a fairly large amount of volume close to x, so either we get a really cheap cut, or at least a cheap cut relative to the large amount of volume near to x. Theorem 20.2.3 of course leads immediately to a deterministic $O(\log k)$-approximation to the MultiCut problem. We simply set $\delta = 1 - \varepsilon$, k equal to the number of terminal pairs, and find a cut of the smallness implied by Theorem 20.2.3 around each s_i. (Note that, as explained below, we do not necessarily grow k regions. We let $a \leq k$ denote the number of regions that are actually grown.)

The diameter bound ensures that no t_i will be in the same region as its corresponding s_i, and the upper-bound on cut size implies that the total cost of all of the cuts is at most

$$a \cdot \frac{\mathrm{Vol}(G)}{k} + \frac{O(\log k)(\mathrm{Vol}(s_{i_1},[0,R_1]) + \cdots + \mathrm{Vol}(s_{i_a},[0,R_a]))}{1-\varepsilon},$$

where $a \leq k$ is the number of cuts made. Note that we must be careful not to perform unnecessary cuts when the current s_i whose region we are growing is already contained within another grown region. It is not necessary to grow this region and make the corresponding cut because s_i already lies in a separate component from t_i because of the diameter bound. This ensures that $\mathrm{Vol}(s_{i_1},[0,R_1]) + \cdots + \mathrm{Vol}(s_{i_a},[0,R_a]) \leq \mathrm{Vol}(G)$ so that it follows that the implicit approximation algorithm is an $O(\log k)$-approximation for the MultiCut problem using the fact that $\mathrm{Vol}(G)$ is a lower bound on the cost of the optimal MultiCut.

References

[1] David B. Shmoys. *Cut problems and their application to divide-and-conquer*, pages 192–235. PWS Publishing Co., Boston, MA, USA, 1997.

[2] Nathan Linial and Michael E. Saks. Low diameter graph decompositions. *Combinatorica*, 13(4):441–454, 1993.

[3] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. In *ACM Symposium on Theory of Computing*, pages 698–707, 1993.

[4] Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, 1999.

[5] Baruch Awerbuch. Complexity of network synchronization. *J. ACM*, 32(4):804–823, 1985.