601.435/635 Approximation Algorithms **Lecturer:** Michael Dinitz
Topic: LP Solutions as Metrics: Multiway Cut **Date:** 3/18/21
Scribe: Michael Dinitz

16.1 Introduction

The last few weeks we've done a lot of work on metric spaces, i.e., on problems where WLOG the input is a metric space. Then we focused on embedding these metrics, allowing us to solve problems on trees rather than on general graphs. We've also done a lot of work on LP-based approximation algorithms. Now we're going to consider some interesting settings in which metrics and LPs are combined in a different way: the *solution* to the LP is a metric, rather than the input being a metric. So then after solving the LP we'll have a metric space that we can manipulate algorithmically.

We're going to see this first in an exact algorithm (not an approximation algorithm) for the classical min $s-t$ cut problem. Then we'll see it used in an approximation algorithm for Multiway Cut.

16.2 Min $s-t$ Cut

We recall the basic definition of the Min Cut Problem

$$\begin{aligned} \text{Input:} \quad & \text{Graph } G = (V, E) \\ & \text{Costs } c : E \to \mathbb{R}^+ \\ & \text{Source } s \in V \text{ and sink } t \in V \\ \text{Feasible:} \quad & A \subseteq E \text{ s.t. } G - A \text{ has no } s-t \text{ path} \\ \text{Objective:} \quad & \min \sum_{e \in A} c(e) \end{aligned}$$

We note that this definition of Min Cut Problem can be written in an equivalent form:

$$\begin{aligned} \text{Input:} \quad & \text{Graph } G = (V, E) \\ & \text{Costs } c : E \to \mathbb{R}^+ \\ & \text{Source } s \in V \text{ and sink } t \in T \\ \text{Feasible:} \quad & S \subseteq V \text{ s.t. } s \in S \text{ and } t \notin S \\ \text{Objective:} \quad & \min \sum_{e \in E(S, \bar{S})} c(e) \end{aligned}$$

Definition 16.2.1 $\mathcal{P}_{s,t} = \{all\ s-t\ paths\}$

We define the following LP, whose integer solutions are solutions to the Min Cut Problem

$$\begin{aligned} \min \quad & \sum_{e \in E} c(e) x_e \\ \text{subject to} \quad & \sum_{e \in P} x_e \geq 1 \quad \forall P \in \mathcal{P}_{s,t} \\ & 0 \leq x_e \leq 1 \qquad \forall e \in E \end{aligned}$$

Intuitively, this states that along each path at least one edge must be in the cut.

Theorem 16.2.2 *This LP can be solved in polytime, even though it has an exponential number of constraints.*

Proof: Suppose $\vec{x}$ is not a feasible solution to the LP. There must thus be a path $P \in \mathcal{P}_{s,t}$ s.t. $\sum_{e \in P} x_e < 1$. We now think of x_e as a length assigned to edge $e \in E$. We can easily find a shortest $s - t$ path (e.g., using Dijkstra's algorithm). Because it is the shortest path, and there is some $P \in \mathcal{P}_{s,t}$ with $\sum_{e \in P} x_e < 1$, the shortest $s - t$ path P^* also has $\sum_{e \in P^*} x_e < 1$, and thus P^* corresponds to a violated constraint. Thus we can use Dijkstra's algorithm as a separation oracle for the ellipsoid method, allowing us to solve the LP in polynomial time. ■

Now we can solve the LP, so let's define a rounding algorithm. We don't want to use *independent* randomized rounding, though – it's not hard to see that this will not do well. Let's instead use the interpretation of x as edge lengths, and round in a way that corresponds to the metric that this gives.

We begin by defining a few variables.

Definition 16.2.3 *Let $d(u)$ denote the shortest path distance from s to u under the edge lengths $x_e \in \vec{x}$.*

Definition 16.2.4 *Let $B(s, r) = \{v \in V : d(v) \leq r\}$.*

Definition 16.2.5 *If $S \subset V$, let $\delta(S) = E(S, \bar{S}) =$ set of edges with one endpoint in S and one endpoint in $\bar{S}$.*

Algorithm 1 LP rounding for Min Cut

Input: Graph $G = (V, E)$ and solution to the LP $\vec{x}$
Output: $S \subseteq E$

Choose r uniformly at random in $(0, 1)$
$S \leftarrow B(s, r)$
return $A \leftarrow \delta(S)$

Claim 16.2.6 *With probability* 1*, the algorithm returns a feasible solution.*

Proof: Since x is a feasible solution, every path from s to t has length at least 1. So $t \notin B(s, r)$ for any choice of r, and thus A cuts t from s. ■

Claim 16.2.7 $\mathbf{Pr}[e \in A] \leq x_e$ *for all* $e \in E$.

Proof: Let $e = \{u, v\}$. Without loss of generality, let $d(u) \leq d(v)$. In order for e to be in A, we must have $r \in [d(u), d(v))$. So $\mathbf{Pr}[e \in A] = \mathbf{Pr}[r \in [d(u), d(v))] \leq d(v) - d(u) \leq d(u, v) \leq x_e$, where $d(u, v)$ denotes the length of the shortest $u - v$ path. ■

So now by linearity of expectations we know that $\mathbf{E}[c(A)] = \sum_{e \in E} c(e)\mathbf{Pr}[e \in A] = \sum_{e \in E} c(e)x_e = LP$. So this algorithm performs as well as the LP in expectation.

We can take this seemingly randomized algorithm and make it deterministic by simply trying all possibilities. Suppose there is no node w such that $d(u) \leq d(w) \leq d(v)$. Thus any $r \in [d(u), d(v)]$ will yield the same cut. Thus there are a linear number of possible cuts, which can each be tested in polynomial time. So our algorithm will be to solve the LP and try each of these cuts, taking whichever is best. Clearly this runs in polynomial time. Since the randomized algorithm does no worse than the LP in expectation, at least one of these cuts must do as well as the LP, so we will return a cut of cost at most $\sum_{e \in E} c(e)x_e$. Thus this algorithm returns the optimal solution, since $\sum_{e \in E} c(e)x_e \geq OPT$. So we have yet another polynomial-time algorithm for min $s - t$ cut.

16.3 Multiway Cut

Let's move to a problem that's actually NP-hard: Multiway Cut.

$$\begin{aligned}
\text{Input:} \quad & \text{Graph } G = (V, E) \\
& \text{Costs } c : E \to \mathbb{R}^+ \\
& T = \{s_1, s_2, ..., s_k\} \subseteq V \\
\text{Feasible:} \quad & A \subseteq E \text{ s.t. } G - A \text{ has no } s_i - s_j \text{ path } \forall i, j \in \{1, 2, ..., k\} \\
\text{Objective:} \quad & \min \sum_{e \in A} c(e)
\end{aligned}$$

In other words, rather than disconnecting s and t, we need to disconnect every terminal from every other terminal (nodes in T are usually called terminals). We define the following LP, whose integer solutions are solutions to Multiway Cut.

$$\begin{aligned}
\min \quad & \sum_{e \in E} c(e)x_e & & \\
\text{subject to} \quad & \sum_{e \in P} x_e \geq 1 & \forall i, j \in \{1, 2, ..., k\}, \forall P \in \mathcal{P}_{s_i, s_j} \\
& 0 \leq x_e \leq 1 & \forall e \in E
\end{aligned}$$

Note that the LP solution $\vec{x}$ induces a metric d on the nodes through the shortest-path distances, and the constraints guarantee that $d(s_i, s_j) \geq 1$ for all $i, j \in [k]$. We use the same separation oracle as we did for Min Cut example, but simply use it to check for each pair of $\{i, j\}$. Thus this LP can also be solved in polynomial time. As earlier, the LP solution $\vec{x}$, when interpreted as edge lengths, gives a metric d by shortest-paths.

Now consider the following rounding algorithm, which is the obvious generalization of the algorithm we designed for Min $s-t$ Cut.

Algorithm 2 LP rounding for Multiway Cut

Input: Graph $G=(V,E)$ and solution to the LP $\vec{x}$
Output: $S \subseteq E$

$A \leftarrow \emptyset$
Choose r uniformly at random from $[0, \frac{1}{2})$
for all $i \in \{1,2,..,k\}$ **do**
  $A_i \leftarrow \delta(B(s_i, r))$
  $A \leftarrow A \cup A_i$
end for
return A

Claim 16.3.1 *This algorithm always returns a feasible solution.*

Proof: By the LP constraints on x, we know that $d(s_i, s_j) \geq 1$ for all terminals s_i, s_j. Since $r < 1/2$, this implies that A_i is enough to cut s_i from s_j (so is A_j, in fact). ■

Now let's analyze the cost.

Claim 16.3.2 $\mathbf{Pr}[e \in A] \leq 2x_e$ *for all* $e \in E$.

Proof: Let $w \in V$. By the triangle inequality, $d(s_i, w) + d(w, s_j) \geq d(s_i, s_j) \geq 1$ for all $s_i, s_j \in T$. This implies that for every $w \in V$, there is at most one $s_i \in T$ such that $d(s_i, u) < 1/2$. So we can define the following disjoint subsets:

Definition 16.3.3 *Let* $C_i = \{w \in V : d(s_i, w) < \frac{1}{2}\}$.

Let $e = \{u, v\} \in E$. To prove the claim, w break into two cases.

Case 1: $u, v \in C_i$ for some $i \in 1, 2, ..., k$. WLOG we assume $d(s_i, u) \leq d(s_i, v)$.

$$\mathbf{Pr}[e \in A] = \mathbf{Pr}[e \in A_i] = \mathbf{Pr}[r \in [d(s_i,u), d(s_i,v))] = \frac{d(s_i,v) - d(s_i,u)}{\frac{1}{2}} \leq 2d(u,v) \leq 2x_e$$

Case 2: There is no i such that $u, v \in C_i$. If neither u nor v is in *any* of the C_i's, then $\mathbf{Pr}[e \in A] = 0$ so we're finished. Otherwise, we know that $u \in C_i$, and either v is not in any C_j or $v \in C_j$ for $j \neq i$. WLOG, let's assume that $d(s_i, u) \leq d(s_j, v)$. Then e will be in A if and only if $r \geq d(s_i, u)$. So we get that

$$\mathbf{Pr}[e \in A] = \mathbf{Pr}[r \in [d(s_i,u), 1/2)] \leq \frac{\frac{1}{2} - d(s_i,u)}{\frac{1}{2}} = 2\left(\frac{1}{2} - d(s_i,u)\right) \leq 2d(u,v) \leq 2x_e$$

So in all cases $\mathbf{Pr}[e \in A] \leq 2x_e$. ■

By linearity of expectations $E[c(A)] = \sum_{e \in E} c(e)\mathbf{Pr}[e \in A] \leq \sum_{e \in E} c(e)(2x_e) \leq 2\sum_{e \in E} c(e)x_e = 2 \cdot LP$. So this is a 2-approximation in expectation, and we can derandomize as we did with min $s-t$ cut by trying out all relevant values of r.

16.3.1 Integrality Gap

Is our analysis tight? We consider the star with k nodes around the outside connected by a single node v. We choose as our k terminal nodes the outside nodes.

The optimal solution OPT is clearly given by cutting all but one edge. So the cost is $k-1$.

The best LP solution is given by assigning all edges $\frac{1}{2}$. So the cost is $\frac{k}{2}$

Thus the gap is given by $\frac{OPT}{LP} = \frac{k-1}{\frac{k}{2}} = 2(1-\frac{1}{k})$. So our analysis is tight.

16.3.2 A Better Algorithm

Despite this integrality gap, it turns out that we can do better by using a better LP relaxation! To construct a better relaxation, we first need to change our viewpoint a bit. Consider some optimal solution F. Then let $C_i = \{v \in V : v \text{ reachable from } s_i \text{ in } G \setminus F\}$. Clearly since F is a feasible multiway cut, $C_i \cap C_j = \emptyset$ for all $i \neq j$. But it turns out that they also form a partition: every node is in some C_i. To see this, suppose that it is false, and let S be the set of nodes that unreachable from any terminal in $G \setminus F$. Add S to C_1 to get C_1', which together with the other C_i's now form a partition. Then any edge which is cut under the new sets was also cut by the old sets, so this new solution is just as good as the old solution.

Thus WLOG, we can assume that the optimal solution is actually a partition $C_1, C_2, \ldots C_k$ where $s_i \in C_i$ for all $i \in [k]$, and the edges we cut are the edges between the parts of the partition.

This point of view suggests the following, different LP relaxation. We'll have the following variables.

$$x_u^i = \begin{cases} 1 & \text{if } u \in C_i \\ 0 & \text{else} \end{cases}$$

$$z_e^i = \begin{cases} 1 & \text{if } e \in \delta(C_i) \\ 0 & \text{else} \end{cases}$$

We now define an LP using these indicator variables

$$\begin{aligned}
\min \quad & \frac{1}{2}\sum_{e\in E}\sum_{i=1}^{k} c(e) z_e^i & & \\
\text{subject to} \quad & \sum_{i=1}^{k} x_u^i = 1 & & \forall u \in V \\
& z_e^i \geq x_u^i - x_v^i & \forall e = \{u,v\} \in E, & \forall i \in 1,2,...,k \\
& z_e^i \geq x_v^i - x_u^i & \forall e = \{u,v\} \in E, & \forall i \in 1,2,...,k \\
& x_{s_i}^i = 1 & & \forall i \in 1,2,...,k \\
& 0 \leq x_u^i \leq 1 & \forall u \in V, & \forall i \in [k] \\
& 0 \leq z_e^i \leq 1 & \forall e \in E, & \forall i \in [k]
\end{aligned}$$