CSE 202: Combinatorial Algorithms
Lecture 14: The Multiway Cut Problem

Lecturer: C. Seshadhri — University of California, Santa Cruz
Scribe: Sabyasachi Basu — May 18, 2021

1 The k-multiway cut problem

In the previous lecture, we saw how one could relax integer problems to some corresponding linear program, and use rounding schemes to get approximation algorithms. In the next two lectures, we will do a detailed study of the classic k-way multicut problem in this context.

Problem: Given undirected $G = (V, E)$ with weights c_e on the edges and k terminals $s_1, \ldots, s_k$, remove a set of edges of minimum cost that separates all terminals.

For $k = 2$, this is precisely the $s - t$ mincut problem, which is polynomial time solvable. For $k \geq 3$, the k-multiway cut problem is $\mathbb{NP}$-hard. As we show below, a simple greedy heuristic yields a $2(1 - 1/k)$-approximation. We will also show an LP rounding scheme that yields the same approximation factor. But careful think about the LP leads to deep insights into relating LP solutions to metrics over vertices. These insights form the basis of the Calinescu-Karloff-Rabani (CKR) relaxation and rounding, which leads to a 3/2-approximation. We will cover details on the CKR approach in the next lecture.

1.1 The isolating cut heuristic

The first result on multiway cut was given by Dahlhaus-Johnson-Papadimitriou-Seymour-Yannakakis, 92. A simple approach to multiway cut is to find mincuts between a single terminal and *all of the other terminals*. Such a cut can be found by a single mincut computation (Exercise). From terminal s_i, this process leads to a cut $(\mathcal{S}_i, \overline{\mathcal{S}_i})$, where $s_i \in \mathcal{S}_i$. We use $c(\delta(\mathcal{S}_i))$ to denote the cost of this cut.

We can remove all the $\delta(\mathcal{S}_i)$s to get a valid multiway cut. A nice observation is that it suffices to only remove $k - 1$ of them, so we can improve on the output cost.

`Isolating cut heuristic`$(G, s_1, \ldots, s_k)$
1. For each i, compute $\mathcal{S}_i$ using a mincut procedure.
2. Output the $k - 1$ smallest $\delta(\mathcal{S}_i)$ cuts.

Claim 1.1. *The output is a valid multiway cut.*

Proof. By removing $\delta(\mathcal{S}_i)$ for $k-1$ choices of s_i, each of these $(k-1)$ terminals is separated from all of the other terminals. But the remaining terminal (for which the algorithm did not remove $\delta(\mathcal{S}_i)$) is also disconnected from the other terminals. But that means each of these $k - 1$ terminals is also separated from the k-th terminal, which implies that this is a valid k-multiway cut. □

Now that we've proved that this is a valid multiway cut, we look at how good it is compared to the optimal.

Claim 1.2. *The cost of the output is at most* $2(1 - 1/k)$ *times the optimal multiway cut.*

Proof. The optimal solution removes some set of edges C to separate all of the $\mathcal{S}_i$'s from each other. Hence, removal of C leads to at least k connected components, with at most one terminal in each component. Let $\mathcal{C}_i$ be the component contains s_i. Since $\delta(\mathcal{C}_i)$ is a cut separating s_i from the remaining terminals and $\delta(\mathcal{S}_i)$ is the optimal such cut, $c(\delta(\mathcal{S}_i)) \leq c(\delta(\mathcal{C}_i))$.

Summing over all i, $\sum_{i=1}^{k} c(\delta(\mathcal{S}_i)) \leq \sum_{i=1}^{k} c(\delta(\mathcal{C}_i))$. Observe that in the latter sum, each edge of C is counted at most twice. (Each edge can be part of at most two cuts $\delta(\mathcal{C}_i)$.) Let OPT denote the cost of C and ALG denote the cost of the algorithm's output.

Since ALG is the cost of the low $(k-1)$ cuts, we deduce

$$ALG \leq \frac{k-1}{k} \cdot \sum_{i=1}^{k} c(\delta(\mathcal{S}_i)) \leq (1-1/k) \sum_{i=1}^{k} c(\delta(\mathcal{C}_i)) \leq 2(1-1/k) OPT$$

□

As a direct consequence, we have the following theorem. Note that for $k=2$, we get back the exact algorithm for s-t mincut. (So there is a "conceptual" benefit for taking the smallest $(k-1)$ cuts.)

Theorem 1.3. *There is a polytime $2(1-1/k)$-approximation for undirected k-multiway cut.*

2 The LP approach

Let us start by writing the LP for the mincut problem ($k=2$), and see how to generalize for larger k.

$$\begin{array}{rl} \min & \sum c_e x_e \\ \forall (u,v) \in E, & d(v) \leq d(u) + x_e \\ & d(s) = 0, \quad d(t) = 1 \\ \forall e & x_e \geq 0 \end{array}$$

This LP is integral, as we've seen in the past. Moreover, we have a rounding scheme by Garg-Vazirani-Yannakakis, as discussed in the earlier lecture on the maxflow-mincut theorem. We pick a value $r \sim \mathcal{U}[0,1)$, and choose $\mathcal{S} = \{v | d_v \leq r\}$. This gives an optimal integral solution.

Note that we had two kinds of variables in the LP: distance variables, and edge selection variables. To generalize the LP to larger k, we follow the same strategy of using distance variables. We have variables $d_i(v)$ to be the distance of terminal s_i from vertex v, and retain x_e as earlier. We set the distance between all pairs $(s_i, s_j) \geq 1$ for any two distinct terminals. Our LP is then:

$$\begin{array}{rl} \min & \sum c_e x_e \\ \forall (u,v) \in E, \forall i \in [k], & d_i(v) \leq d_i(u) + x_e \\ \forall i,j \in [k], i \neq j & d_i(s_i) = 0, d_i(s_j) \geq 1 \\ \forall e & x_e \geq 0 \end{array}$$

To see that this a valid relaxation for the multiway cut IP, let x_e be the indicator for a valid multicut. The shortest path distances between all pairs of terminals are at least one, since they all lie in different connected components.

Let us look at randomized rounding methods for this LP. We start with a naive extension of the mincut rounding, and see why it *doesn't* work. All the schemes discussed will produce "isolating" sets $\mathcal{S}_i$, each of which contain a single terminal. Thus, they follow the same high-level strategy of the isolating cut algorithm.

Rounding Scheme, Attempt #1: Let's just follow what we did for standard mincut. We pick a value $r \sim \mathcal{U}[0,1)$, and define $\mathcal{S}_i = \{v | d_i(v) \leq r\}$. Since $\forall i \neq j$, $d_i(s_j) \geq 1$, each $\mathcal{S}_i$ contains only one terminal. This gives us a valid mincut. We now need to bound the expected cost of the cut. As in the mincut analysis, let X_e be the indicator random variable for edge e being cut. We wish to bound $\mathbf{E}[\sum_e X_e] = \sum_e \mathbf{E}[X_e] = \sum_e \Pr[e \text{ is cut}]$. So, if we can upper bound the probability that an edge (u,v) is cut by αx_{uv}, then we get an α-approximation ratio.

We can think of k (or $k-1$) iterations, where each iteration cuts some $\delta(\mathcal{S}_i)$. The edge (u,v) may be cut in any iteration. What is the probability that (u,v) is cut in the iteration corresponding to $\delta(\mathcal{S}_i)$? Wlog, let us assume that $d_i(u) \leq d_i(v)$. The edge (u,v) is cut precisely when $r \in [d_i(u), d_i(v))$.

$$Pr[r \in [d_i(u), d_i(v))] = d_i(v) - d_i(u) \leq x_{uv} \tag{1}$$

The probability that (u,v) is cut in one of $(k-1)$ iterations is at most $1-(1-x_{uv})^{k-1} \approx (k-1)x_{uv}$; this is dreadful, because we already had a 2 approximation from the isolating cut approach!

2.1 The optimal rounding scheme

The problem with the previous rounding scheme is that every iteration had some chance of cutting (u,v), and the overall probability of cutting was too high. By exploiting the fact that graph is undirected, we can ensure that *at most two iterations* will cut (u,v). This will lead to the 2-approximation.

Our new scheme is: pick $r \sim \mathcal{U}[0,1/2)$, and define $\mathcal{S}_i = \{v | d_i(v) \leq r\}$. Since $r < 1$, each $\mathcal{S}_i$ is an isolating set as desired. The following simple claim is key to the analysis.

Claim 2.1. *For any vertex v, there is a unique terminal s_i such that $d_i(v) < 1/2$.*

Proof. If not, there exist indices $i \neq j$ such that $d_i(v) < 1/2$ and $d_j(v) < 1/2$. Now, we use triangle inequality of distances and the fact that the graph is undirected (so $dist(v, s_j) = dist(s_j, v)$).

$$d_i(s_j) = dist(s_i, s_j) \leq dist(s_i, v) + dist(v, s_j) = d_i(v) + d_j(v) < 1. \tag{2}$$

This violates the distance constraint between the terminals. □

Let s_i and s_j be the two unique vertices (if they exist) at distance less than $1/2$ from u and v respectively. In the new rounding scheme, $\mathcal{S}_i$ or $\mathcal{S}_j$ can cut (u,v).

Claim 2.2. $\Pr[\exists b, (u,v) \in \delta(\mathcal{S}_b)] \leq 2x_{uv}$.

Proof. We split the proof into two cases. We call the two unique terminals s_i and s_j as above.
Case 1: $s_i = s_j$: In this case, (u, v) can only be cut by $\delta(\mathcal{S}_i)$. Then $Pr[(u, v) \text{ is cut}] = Pr[r \in [d_i(u), d_i(v))] = (d_i(v) - d_i(u))/0.5 \leq 2x_{uv}$. The calculation here is identical to that in (1), except that r is now in $[0, 1/2)$ instead of $[0, 1)$. The latter fact introduces the factor of 2.
Case 2: $s_i \neq s_j$: In this case, our probability is the union of two events. We will use the bound $\Pr[(u, v) \in \delta(\mathcal{S}_i)] \leq (1/2 - d_i(u))/(0.5) = 2(1/2 - d_i(u))$. By the union bound,

$$
\begin{aligned}
\Pr[\exists b, (u, v) \in \delta(\mathcal{S}_b)] &\leq \Pr[(u, v) \in \delta(\mathcal{S}_i)] + \Pr[(u, v) \in \delta(\mathcal{S}_j)] \\
&\leq 2 \times \left[\frac{1}{2} - d_i(u) + \frac{1}{2} - d_j(v)\right]
\end{aligned}
\tag{3}
$$

Consider a path from s_i to s_j going from s_i to u, taking edge (u, v), and then from v to s_j. By triangle inequality $1 \leq d_i(s_j) \leq d_i(u) + x_{u,v} + d_j(v)$. Rearranging, $1 - d_i(u) - d_j(v) \leq x_{u,v}$. Plugging this bound into the above, $\Pr[\exists b, (u, v) \in \delta(\mathcal{S}_b)] \leq 2x_{u,v}$ as desired. $\square$

These claims lead to the following theorem.

Theorem 2.3. *There is an LP rounding scheme that leads to a $2(1 - 1/k)$-approximation.*

Proof. Just for completeness' sake, let us write out the proof formally. Applying the rounding scheme where $r \sim \mathcal{U}[0, 1/2)$, by Claim 2.2,

$$\mathbf{E}[\sum_i c(\delta(\mathcal{S}_i)] = \sum_e \Pr[\exists i, e \in \delta(S_i)] \leq 2 \sum_e x_e = 2 \cdot cost(x)$$

Thus, $\bigcup_i \delta(\mathcal{S}_i)$ is a 2-approximation for multiway cut. We can again turn this into a $2(1 - 1/k)$ approximation by taking the $k - 1$ cheapest cuts among the $\delta(\mathcal{S}_i)$'s. $\square$

This rounding scheme is optimal. We give a matching integrality gap.

Theorem 2.4. *The integrality gap of the LP is at least $2(1 - 1/k)$.*

Proof. Consider a k-star, where all the leaves are terminals. We need to remove at least $k - 1$ edges to separate the terminals, so the optimal of the IP is at least $k - 1$. (The optimal value is $k - 1$.) For the LP, set $x_e = 1/2$. The shortest path distance between all pairs of terminals is exactly 1, so this solution is feasible. The cost of the LP is at most $k/2$. Thus the integrality gap is at least $(k - 1)/(k/2) = 2(1 - 1/k)$. $\square$

3 Exercises

Easy exercises.

1. Consider a multiway cut instance. Give an efficient algorithm that computes the optimal cut separating s_i from all the remaining terminals.
2. Argue that the integrality gap of the LP is exactly $2(1 - 1/k)$.

Medium exercises.

1. Prove that the optimal multiway cut creates exactly k connected components, each containing a different terminal.

2. In the LP for multiway cut, we set $d_i(s_j) \geq 1$. Prove that if we replace these constraints with equality constraints $d_i(s_j) = 1$, then the LP is not a valid relaxation any longer.
3. For the integrality gap example in Theorem 2.4, prove that the LP optimum has value exactly 2.
4. Consider the directed multiway cut problem where the input is a directed graph, and we have to remove edges so that no terminal can reach any other terminal. Give a k-approximation algorithm for this problem.