CMPUT 675: Approximation Algorithms **Fall 2015**

Lecture 11-12 (Oct 6 & 8, 2015): Multiway Cut (Continuation), Multi-Cut

Lecturer: Mohammad R. Salavatipour *Scribe: Yao Xu*

11.1 An LP Rounding Algorithm for the Multiway Cut Problem

In the last lecture, we introduced an LP relaxation and the corresponding LP rounding algorithm for the Multiway Cut problem.

11.1.1 Recall: Definition and the Linear Program

Definition 1 Multiway Cut Problem: *Given an undirected graph $G = (V, E)$, a cost function $c : E \to \mathbb{Q}^+$ on edges, and k distinguished terminals, $s_1, s_2, \ldots, s_k$, where $s_i \in V$, for all $i = 1, 2, \ldots, k$, the goal is to find a minimum-cost set of edges, $E' \subseteq E$, whose removal disconnects all terminals from each other.*

The *linear program* (LP) of the Multiway Cut problem we talked about in the last lecture is as follows:

$$\begin{aligned} \text{minimize} \quad & \sum_{e=(u,v)\in E} c_e \cdot \|x_u - x_v\|_1 & & \quad (11.1) \\ \text{subject to} \quad & x_{s_i} = e_i & & i = 1, 2, \ldots, k, \\ & x_u \in \Delta_k & & \forall u \in V. \end{aligned}$$

where $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ is the vector with 1 in the ith coordinate and zeros elsewhere, and Δ_k is the k-simplex, *i.e.*, $\Delta_k = \{x \in \mathbb{R}^k | \sum_{i=1}^{k} x^i = 1\}$.

11.1.2 Recall: The Randomized Rounding Algorithm

For any $r \geq 0$ and $1 \leq i \leq k$, let $B(s_i, r)$ be the set of vertices in a ball of radius r in the ℓ_1-metric around s_i, that is, $B(s_i, r) = \{u \in V | \frac{1}{2}\|x_{s_i} - x_u\|_1 \leq r\}$. Note that $B(s_i, 1) = V$ for all i. Then, as we introduced in the last lecture, the following algorithm MWC2 is a randomized rounding algorithm for the Multiway Cut problem.

Algorithm MWC2: LP Rounding Algorithm for the Multiway Cut Problem

1. Let x^* be an optimal fractional solution to (11.1)
2. $C_i \leftarrow \emptyset$ for all $1 \leq i \leq k$
3. Pick $r \in (0, 1)$ uniformly at random
4. Pick a random permutation π of $\{1, 2, \ldots, k\}$
5. **for** $i \leftarrow 1$ to $k - 1$ **do**
6. $\quad C_{\pi(i)} \leftarrow B(s_{\pi(i)}, r) - \bigcup_{j<i} C_{\pi(j)}$
7. $C_{\pi(k)} \leftarrow V - \bigcup_{j<k} C_{\pi(j)}$
8. **return** $F = \bigcup_{i=1}^{k} \delta(C_i)$

11.1.3 Analysis of the Randomized Rounding Algorithm

Lemma 1 *For each $e = (u, v)$, the probability of e belonging to the cut, i.e.,* $\Pr[e \textit{ is in cut}] \le \frac{3}{4}\|x_u - x_v\|_1$.

Lemma 1 implies the following theorem and we will prove the lemma later.

Theorem 1 *Algorithm MWC2 is a randomized $\frac{3}{2}$-approximation algorithm for the multiway cut problem.*

Proof. Let W be a random variable denoting the value of the cut, and Z_e be a 0-1 variable which is 1 if e is in the cut, so that $W = \sum_{e \in E} c_e Z_e$. Let OPT be the optimum solution of the LP. Then, we have

$$\begin{aligned} E[W] = E\left[\sum_{e\in E} c_e Z_e\right] = \sum_{e \in E} c_e E[Z_e] &= \sum_{e\in E} c_e \Pr[e \text{ is in cut}] \\ &\le \sum_{e=(u,v)\in E} c_e \frac{3}{4}\|x_u - x_v\|_1 && \triangleleft \text{ by Lemma 1} \\ &= \frac{3}{2}\cdot\frac{1}{2}\sum_{e=(u,v)\in E} c_e \|x_u - x_v\|_1 \\ &= \frac{3}{2}\cdot \text{OPT}. \end{aligned}$$

■

Before proving Lemma 1, we first prove the following two lemmas.

Lemma 2 *For any index ℓ and any two vertices $u, v \in V$, $|x_u^\ell - x_v^\ell| \le \frac{1}{2}\|x_u - x_v\|_1$.*

Proof. Without loss of generality, assume that $x_u^\ell \ge x_v^\ell$. Then

$$|x_u^\ell - x_v^\ell| = x_u^\ell - x_v^\ell = \left(1 - \sum_{j\ne\ell} x_u^j\right) - \left(1 - \sum_{j \ne \ell} x_v^j\right) = \sum_{j\ne\ell}(x_v^j - x_u^j) \le \sum_{j \ne \ell}|x_u^j - x_v^j|.$$

Add $|x_u^\ell - x_v^\ell|$ to both sides, we have

$$2|x_u^\ell - x_v^\ell| \le \|x_u - x_v\|_1 \Rightarrow |x_u^\ell - x_v^\ell| \le \frac{1}{2}\|x_u - x_v\|_1.$$

■

Lemma 3 $u \in B(s_i, r) \Leftrightarrow 1 - x_u^i \le r$.

Proof.

$$\begin{aligned} u \in B(s_i, r) &\Leftrightarrow \frac{1}{2}\|x_{s_i} - x_u\|_1 \le r \\ &\equiv \frac{1}{2}\sum_{j=1}^{k}|x_{s_i}^j - x_u^j| \le r \\ &\equiv \frac{1}{2}\sum_{j=i} x_u^j + \frac{1}{2}(1 - x_u^i) \le r \\ &\equiv 1 - x_u^i \le r. && \triangleleft \text{ since } \sum_{j=i} x_u^j = 1 - x_u^i \end{aligned}$$

■

Now we can prove Lemma 1 based on the above two lemmas.

Proof. Consider an edge $e = (u, v)$, define the following two events:

- Event S_i: we say that index i *settles* e if i is the first index such that at least one of $u, v \in B(s_{\pi(i)}, r)$;
- Event X_i: we say that index i *cuts* e if exactly one of $u, v \in B(s_{\pi(i)}, r)$.

Then, we have $\Pr[e \text{ is in cut}] = \sum_{i=1}^{k} \Pr[S_i \wedge X_i]$. By Lemma 3, we get

$$\Pr[X_i] = \Pr[\min\{1 - x_u^i, 1 - x_v^i\} \leq r < \max\{1 - x_v^i, 1 - x_u^i\}] = |x_u^i - x_v^i|.$$

Let $\ell = \arg\min_i\{1 - x_u^i, 1 - x_v^i\}$, that is, s_ℓ is the nearest terminal to either u or v. Then we can claim that index $i \neq \ell$ cannot settle $e = (u, v)$ if ℓ comes before i in π, since by Lemma 3, if at least one of $u, v \in B(s_{\pi(i)}, r)$, then at least one of $u, v \in B(s_{\pi(\ell)}, r)$. Note that $\Pr[\ell \text{ comes after } i] = \frac{1}{2}$. Thus,

- for $\ell \neq i$, we have

$$\begin{aligned}
\Pr[S_i \wedge X_i] &= \frac{1}{2}\Pr[S_i \wedge X_i | \ell \text{ comes after } i] + \frac{1}{2}\Pr[S_i \wedge X_i | \ell \text{ comes before } i] \\
&\leq \frac{1}{2}\Pr[X_i | \ell \text{ comes after } i] + 0 \\
&= \frac{1}{2}\Pr[X_i] && \triangleleft\, X_i \text{ is independent of } \pi \\
&= \frac{1}{2}|x_u^i - x_v^i|.
\end{aligned}$$

- for $\ell = i$, we have

$$\Pr[S_\ell \wedge X_\ell] \leq \Pr[X_\ell] = |x_u^\ell - x_v^\ell|.$$

Therefore,

$$\begin{aligned}
\Pr[e \text{ is in cut}] = \sum_{i=1}^{k} \Pr[S_i \wedge X_i] &\leq |x_u^\ell - x_v^\ell| + \frac{1}{2}\sum_{i \neq \ell} |x_u^i - x_v^i| \\
&= \frac{1}{2}|x_u^\ell - x_v^\ell| + \frac{1}{2}\|x_u - x_v\|_1 \\
&\leq \frac{1}{4}\|x_u - x_v\|_1 + \frac{1}{2}\|x_u - x_v\|_1 && \triangleleft \text{ by Lemma 2} \\
&= \frac{3}{4}\|x_u - x_v\|_1.
\end{aligned}$$

■

11.1.4 Best Known Results

Theorem 2 *There is a Multiway Cut randomized approximation algorithm with an approximation guarantee of* 1.3438*.* [K04]

Theorem 3 *There exists a* $(1.32388 - \frac{1}{2k})$*-approximation algorithm for the Multiway Cut problem.* [BNS13]

Theorem 4 *There is an algorithm that provides a* 1.2965*-approximation for the Multiway Cut problem.* [SV14]

11.2 The Multi-Cut Problem

Definition 2 Multi-Cut Problem: *Given an undirected graph $G = (V, E)$, a cost function $c : E \to \mathbb{Q}^+$ on edges, and k distinguished source-sink pairs of vertices, $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$, where $s_i, t_i \in V$, for all $i = 1, 2, \ldots, k$, the goal is to find a minimum-cost set of edges, $E' \subseteq E$, whose removal disconnects all pairs of s_i, t_i, for every $i = 1, 2, \ldots, k$. Note that there can be paths connecting s_i and s_j or s_i and t_j for $i \neq j$.*

Let $\mathcal{P}_i$ be the set of all paths from s_i to t_i. Then an LP of this problem is as follows:

$$\begin{aligned} \text{minimize} \quad & \sum_{e \in E} c_e x_e \\ \text{subject to} \quad & \sum_{e \in P} x_e \geq 1, \quad \forall P \in \mathcal{P}_i, 1 \leq i \leq k, \\ & x_e \geq 0, \qquad \forall e \in E. \end{aligned} \tag{11.2}$$

Although this LP has exponentially many constraints, we can solve it in polynomial time by considering a polynomial-time *separation oracle*, which is defined as follows:

Separation oracle: Given a solution of x_e values, either say it is indeed a feasible solution to the LP or, if it is infeasible, find a violating constraint.

The separation oracle for this LP works as follows: Consider x_e as the length of each edge in G, compute the length of the shortest $s_i - t_i$ path for each i, $1 \leq i \leq k$. If for each i, the length of the shortest $s_i - t_i$ path is at least 1, then the length of every path $P \in \mathcal{P}_i$ is at least 1, indicating that the solution is feasible; if for some i, the length of the shortest $s_i - t_i$ path P is less than 1, we return it as a violated constraint, since we have $\sum_{e \in P} x_e < 1$ for $P \in \mathcal{P}_i$.

11.2.1 The Region Growing Algorithm

Now we introduce an approximation algorithm based on a *region growing* method presented by Garg, Vazirani, and Yannakakis (GVY) for solving this problem. First, we restate this problem as a pipe system with some denotations as follows:

- x_e: length of a pipe
- c_e: cross-sectional area of a pipe
- $c_e x_e$: volume of a pipe
- $d_x(u, v)$: length of the shortest $u - v$ path with edge length x_e
- $B_x(v, r) = \{u | d_x(v, u) \leq r\}$: ball of radius r around vertex v

The LP objective is then the minimum-volume pipe system such that for every $s_i - t_i$ path, s_i and t_i are at least 1 unit apart, *i.e.*, $d_x(s_i, t_i) \geq 1$. See Figure 11.1 for an illustration of a pipe system.

Let V^* be the optimum total volume of the pipes to the LP, we define the volume of pipes within distance r of s_i plus an extra term $\frac{V^*}{k}$ as follows:

$$V_x(s_i, r) = \frac{V^*}{k} + \sum_{e=(u,v),\ u,v \in B_x(s_i,r)} c_e x_e + \sum_{e=(u,v),\ u \in B_x(s_i,r),\ v \notin B_x(s_i,r)} c_e (r - d_x(s_i, u)).$$

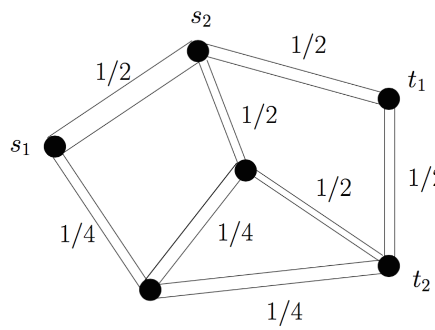

Figure 11.1: An illustration of a pipe system.

Let $\delta(S)$ be the set of edges between S and $V \setminus S$ for all $S \subset V$. The following algorithm GVY is a region growing algorithm for the Multi-Cut problem.

Algorithm GVY: The Region Growing Algorithm for the Multi-Cut Problem

1. $C \leftarrow \emptyset$
2. Let x be an optimal fractional solution to (11.2)
3. **while** there is a connected s_i, t_i **do**
4. $\quad S \leftarrow B_x(s_i, r)$ for some $r < \frac{1}{2}$
5. $\quad C \leftarrow C \cup \delta(S)$ $\qquad \triangleleft\ \delta(S)$ cuts S from the rest
6. $\quad V \leftarrow V \setminus S$ $\qquad \triangleleft\ \delta(S)$ Remove the ball from the G
7. **return** C

11.2.2 Analysis of the GVY Region Growing Algorithm

Lemma 4 *Algorithm GVY terminates in polynomial time.*

Proof. In each iteration of the while loop, lines 4 and 5 indicate that $\delta(S)$ will separate at least one pair of (s_i, t_i), thus there are at most k iterations. Therefore, algorithm GVY terminates in polynomial time. ∎

Lemma 5 *Algorithm GVY returns a Multi-Cut.*

Proof. If algorithm GVY does not return a Multi-Cut, then there must be some $s_j - t_j$ pair in a removed ball. Thus, we show that no $s_j - t_j$ pair remains connected within a ball that is removed by contradiction. If $\exists s_j, t_j \in B_x(s_i, r)$ for $r < \frac{1}{2}$, then $d_x(s_j, t_j) \leq 2r < 1$, which contradicts the constraints for s_j, t_j. ∎

Let V^* be the optimum total volume of the pipes to the LP, then as we introduced in the last lecture, define:

$$V_x(s_i, r) = \frac{V^*}{k} + \sum_{e=(u,v),\ u,v \in B_x(s_i,r)} c_e x_e + \sum_{e=(u,v),\ u \in B_x(s_i,r),\ v \notin B_x(s_i,r)} c_e(r - d_x(s_i, u)),$$

$$C_x(s_i, r) = \sum_{e=(u,v) \in \delta(B_x(s_i,r))} c_e.$$

Observation: $V_x(s_i, r)$ is an increasing function of r. It is also piece-wise linear with possible discontinuity at values of r when the ball includes a new vertex (see Figure 11.2 for an example of the discontinuity) and differentiable between values r in which vertices are added to the ball.

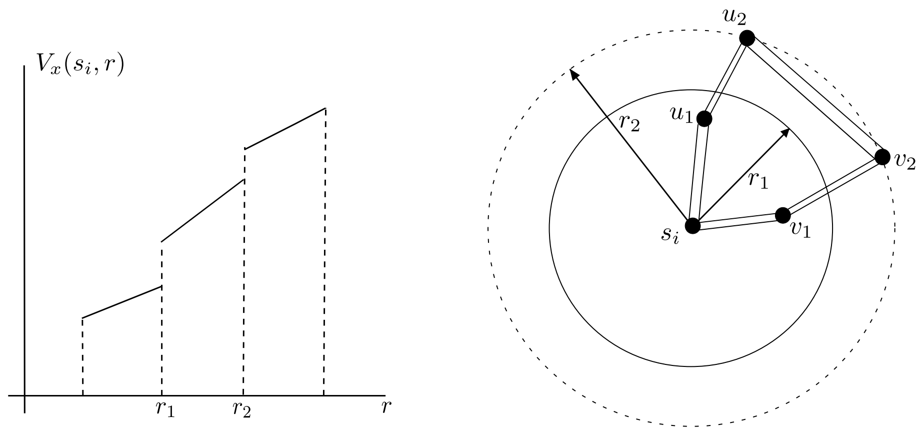

Figure 11.2: An example of when the function $V_x(s_i, r)$ of r is discontinuous. The value of $V_x(s_i, r)$ can jump when a ball is growing with a radius from r_1 to r_2 and there is an edge between u_2 and v_2 which have the same distance (r_2) to s_i, since we will also need to add the volume of pipe (u_2, v_2) at the moment when r reaches r_2.

So, we have

$$\frac{\mathrm{d}V_x(s_i, r)}{\mathrm{d}r} = C_x(s_i, r).$$

Lemma 6 *There is some $r < \frac{1}{2}$ (and we can find it in polynomial time) such that $\frac{C_x(s_i,r)}{V_x(s_i,r)} \leq 2\ln(k+1)$.*

Lemma 6 implies the following theorem and we will prove the lemma later.

Theorem 5 *Algorithm GVY is a $4\ln(k+1)$-approximation algorithm for the Multi-Cut problem.*

Proof. When we cut a ball $B_x(s_i, r)$, charging the cost of $\delta(B_x(s_i, r))$ to the volume of $B_x(s_i, r)$, by Lemma 6, we have

$$\begin{aligned}
\sum_{e \in C} c_e = \sum_{i=1}^{k} \sum_{e \in C_i} c_e &\leq 2\ln(k+1) \sum_{s_i, r \text{ selected}} V_x(s_i, r) \\
&\leq 2\ln(k+1)(V^* + k \cdot \frac{V^*}{k}) \\
&= 4\ln(k+1)V^*.
\end{aligned}$$

■

Now we prove Lemma 6.

Proof. Say we choose $r \in [0, \frac{1}{2})$ uniformly at random. Recall the *mean-value theorem*: for a function $f(\cdot)$ continuous on an interval $[a, b]$ and differentiable on (a, b), $\exists c \in (a, b)$ such that $f'(c) = \frac{f(b)-f(a)}{b-a}$ (see Figure 11.3 for the proof).

Let $f(r) = \ln V(r), f'(r) = \frac{\mathrm{d}\ln V(r)}{\mathrm{d}r} = \frac{C(r)}{V(r)}$, where $V(r) = V_x(s_i, r), C(r) = C_x(s_i, r)$. Note that $V(\frac{1}{2}) \leq V^* + \frac{V^*}{k}$ and $V(0) = \frac{V^*}{k}$. Thus, $\exists r_0$ such that

$$f'(r_0) \leq \frac{\ln V(\frac{1}{2}) - \ln V(0)}{\frac{1}{2} - 0} \leq 2\left(\ln(V^* + \frac{V^*}{k}) - \ln\frac{V^*}{k}\right) = 2\left(\frac{\ln(V^* + \frac{V^*}{k})}{\ln\frac{V^*}{k}}\right) = 2\ln(k+1).$$

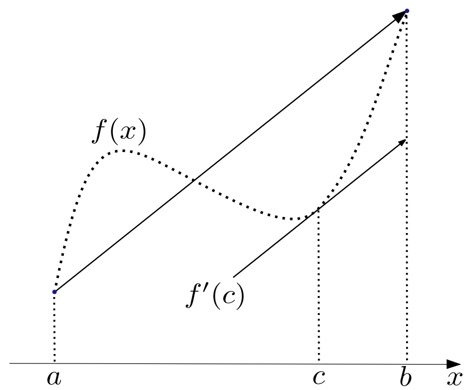

Figure 11.3: Mean-value theorem

Consider vertices based on their increasing distances from s_i: $s_i = v_1, v_2, \ldots, v_p$, $0 = r_0 \leq r_1 \leq \cdots \leq r_p = \frac{1}{2}$. By contradiction, suppose for all $r \in [r_j, r_{j+1})$, $\frac{C(r)}{V(r)} > 2\ln(k+1)$. Then, we have

$$\int_{r_j}^{r_{j+1}^-} \frac{\mathrm{d}V(r)}{\mathrm{d}r} \cdot \frac{1}{V(r)} \mathrm{d}x > \int_{r_j}^{r_{j+1}^-} 2\ln(k+1)\mathrm{d}r$$
$$\Rightarrow \ln V(r_{j+1}^-) - \ln V(r_j) > 2\ln(k+1)(r_{j+1}^- - r_j).$$

For all $j = 0, 1, \ldots, p-1$, we have

$$\ln V(r_1) - \ln V(r_0) > 2\ln(k+1)(r_1 - r_0),$$
$$\vdots$$
$$\ln V(r_p) - \ln V(r_{p-1}) > 2\ln(k+1)(r_p - r_{p-1}).$$

Sum over all j, we get

$$\begin{aligned}
\ln V(r_p) - \ln V(r_0) &> 2\ln(k+1)(r_p - r_0) \\
\Rightarrow \ln V(\frac{1}{2}) - \ln V(0) &> 2\ln(k+1)(\frac{1}{2} - 0) \\
\Rightarrow \ln V(\frac{1}{2}) &> \ln(k+1) + \ln\frac{V^*}{k} \\
&= \ln\frac{(k+1)V^*}{k} \\
&= \ln(V^* + \frac{V^*}{k}) \\
\Rightarrow V(\frac{1}{2}) &> V^* + \frac{V^*}{k}
\end{aligned}$$

which cannot happen. Therefore, there must be an r such that $\frac{C(r)}{V(r)} \leq 2\ln(k+1)$.
To find such r, note that between r_j and r_{j+1}^-, $C(r)$ is constant while $V(r)$ is non-decreasing. So, the minimum value of $\frac{C(r)}{V(r)}$ occurs when $r = r_{j+1}^-$. So, it is enough to check the ratio $\frac{C(r)}{V(r)}$ for $r = r_{j+1} - \epsilon$. So, we only need to check $p \leq n$ vertices and their distances from s_i. Thus, we can find such r in polynomial time. ■

Therefore, algorithm GVY is an $O(\log k)$-approximation algorithm for the Multi-Cut problem.

References

WS11 D. P. Williamson and D. B. Shmoys, The design of approximation algorithms, *Cambridge University Press*, 2011.

K04 D. R. Karger, P. Klein, C. Stein, M. Thorup and N. E. Young, Rounding algorithms for a geometric embedding of minimum multiway cut, *Mathematics of Operations Research*, 2004, 29(3):436–461.

BNS13 N. Buchbinder, J. S. Naor and R. Schwartz, Simplex partitioning via exponential clocks and the multiway cut problem, *In Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, 2013, pp. 535-544, ACM.

SV14 A. Sharma and J. Vondrák, Multiway cut, pairwise realizable distributions, and descending thresholds, *In Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, 2014, pp. 724–733, ACM