601.435 / 601.635 Approximation Algorithms **Spring 2024**
Homework #5 **Due:** April 18, 2024, 9:00am

Reminder: you may work in groups of up to three people, but must write up solutions entirely on your own. Collaboration is limited to discussing the problems – you may not look at, compare, reuse, etc. any text from anyone else in the class. Please include your list of collaborators on the first page of your submission. Many of these problems have solutions which can be found on the internet – please don't look. You can of course use the internet (including the links provided on the course webpage) as a learning tool, but don't go looking for solutions.

Please include proofs with all of your answers, unless stated otherwise.

Student: Ha Manh Bui (hbui13@jhu.edu)

1 Multiway Cut (50 points)

Consider the following two permutations π_1 and π_2 of $[k]$, where $\pi_1(1) = 1, \pi_1(2) = 2, \dots, \pi_1(k) = k$ and $\pi_2(1) = k, \pi_2(2) = k-1, \dots, \pi_2(k) = 1$.

(a) (25 points) Consider a modification of the 3/2 approximation for Multiway Cut from Lecture 17: instead of choosing π uniformly at random from all permutations of $[k]$, we choose $\pi = \pi_1$ with probability 1/2 and choose $\pi = \pi_2$ with probability 1/2. Prove that this modified algorithm is still a 3/2-approximation for Multiway Cut.

Recall that in multiway cut, given a graph $G = (V, E)$, our LP is

$$\begin{aligned} \min \quad & \frac{1}{2} \sum_{e=\{u,v\} \in E} c(e) ||x_u - x_v||_1 \\ \text{subject to } & x_{s_i} = e_i \quad \forall i \in [k] \\ & x_u \in \Delta_k, \end{aligned}$$

and $||x_{s_i} - x_{s_j}||_1 = ||e_i - e_j||_1 = 2$ for all $i \neq j \in [k]$. Using the same notation and analysis from Lecture 17, then to show the modified algorithm is still a 3/2-approximation for Multiway Cut, we only need to show

$$p(Z_{u,v} = 1) \leq \frac{3}{4} ||x_u - x_v||_1, \text{ for all} \{u, v\} \in E.$$

Proof: We can see that if $x_u = x_v$, then $p(Z_{u,v} = 1) = 0$. So, for a pair of terminal $i \neq j$, assuming $\{u, v\} \in E_{ij} = \{u, v \in E, (x_u^i \neq x_v^i) \wedge (x_u^j \neq x_v^j)\}$, then we have $\sum_{t=1}^k x_u^t = 1 = \sum_{t=1}^k x_v^t$. Since $x_u^t = x_v^t$ for all $t \notin \{i, j\}$, we have $x_u^i + x_u^j = x_v^i + x_v^j$. Assume, without loss of generality, that $x_u^i = \max\{x_u^i, x_u^j, x_v^i, x_v^j\}$, we obtain either $x_u^i \geq x_v^i \geq x_v^j \geq x_u^j$ or $x_u^i \geq x_v^j \geq x_v^i \geq x_u^j$.
For any $l \in \{1, 2, \cdots, k\} \setminus \{i, j\}$, $x_u^l = x_v^l$, so either both $u, v \in B(l, r)$ or both $u, v \notin B(l, r)$. So the only way $Z_{u,v} = 1$ is to have either $r \in I_l = (1 - x_u^i, 1 - x_v^i]$ or $r \in I_r = (1 - x_v^j, 1 - x_u^j]$. By $1 - x_v^i \leq 1 - x_u^j$, both intervals I_l and I_r have the same length $\frac{1}{2}||x_u - x_v||_1$, so $I_l \cup I_r = I_l \cup (I_r \setminus I_l)$, so that for $Z_{u,v} = 1$, either $r \in I_l$ or $r \in I_r \setminus I_L$.

For $i \neq j$, let $i \prec j$ denote i precedes j by π. Suppose $i \prec j$ and i are processed, neither u nor v has been assigned to a vertex. For $\{u, v\}$ to be cut, we must have $r \in I_l \cup (I_r \setminus I_l)$. So, if $r \in I_r \setminus I_l$, then $r > 1 - x_v^i \geq 1 - x_u^i$. Hence, when i is processed, both u and v are assigned to terminal i, ensuring $Z_{u,v} = 0$. Therefore, by choosing $\pi = \pi_1$ with probability 1/2 and choose $\pi = \pi_2$ with probability 1/2, for all $\{u, v\} \in E$, we obtain

$$p(Z_{u,v} = 1) \leq p\left((j \prec i) \wedge (r \in I_L \cup I_r)\right) + p\left((i \prec j) \wedge (r \in I_L)\right) \tag{1}$$

$$\leq \frac{1}{2}||x_u - x_v||_1 + \frac{1}{4}||x_u - x_v||_1 \tag{2}$$

$$= \frac{3}{4}||x_u - x_v||_1. \tag{3}$$

■

(b) (25 points) Using the previous part, design a *deterministic* 3/2-approximation for Multiway Cut. As always, prove the approximation ratio and polynomial running time.

Proof: Using the result from the previous part, we have

$$\mathbb{E}[ALG] = \mathbb{E}\left[\sum_{e=\{u,v\} \in E} c(e) Z_{u,v}\right] = \sum_{e=\{u,v\} \in E} c(e) \mathbb{E}\left[Z_{u,v}\right] \tag{4}$$

$$\leq \sum_{e=\{u,v\} \in E} c(e) \frac{3}{4}||x_u - x_v||_1 \leq \frac{3}{2} OPT. \tag{5}$$

Since there are two possible choices for π which are $\pi = \pi_1$ with probability 1/2 and $\pi = \pi_2$ with probability 1/2, so for a given permutation π, two different values of r, $r_1 < r_2$, produce combinatorially distinct solutions only if there is a terminal i and a node u s.t. $x_u^i \in (1 - r_2, 1 - r_1]$. Thus we can design a deterministic algorithm by enumerating over at most $k|V|$ values of r. We can determine these values by sorting the nodes according to each coordinate separately. The resulting discrete sample space for (π, r) has a size at most $2k|V|$, so we can search it to find a point that solves cost at most the expectation. Therefore, in polynomial time, we can construct a multiway cut of 3/2-approximation. ■

2 Multicut in Trees (50 points)

Consider the *multicut problem in trees*. In this problem, we are given a tree $T = (V, E)$, k pairs (s_i, t_i) of vertices, and edge costs $c : E \to \mathbb{R}^+$. A feasible solution is a set $F \subseteq E$ such that for all $i \in [k]$, s_i and t_i are in different connected components of $T - F$. The objective is to minimize the total edge cost $\sum_{e \in F} c(e)$.

Let P_i be the unique path between s_i and t_i in T. Then we can write an integer linear programming formulation of this problem:

$$\begin{aligned} \min \quad & \sum_{e \in E} c(e) x_e \\ \text{subject to} \quad & \sum_{e \in P_i} x_e \geq 1 \quad \forall i \in [k] \\ & x_e \in \{0, 1\} \quad \forall e \in E \end{aligned}$$

(a) (25 points) Write the dual of the LP relaxation of the above ILP (note: we did this in class for multicut!)

Since we can relax the integrality constraint $x_e \in \{0, 1\}$ to $x_e \geq 0$, then any optimal solution x^* to this linear program will have $x^*_e \leq 1$, $\forall e \in E$. So, the dual of this linear program is

$$\max \quad \sum_{i \in [k]} y_i \tag{6}$$

$$\text{subject to} \quad \sum_{i : e \in P_i} y_i \leq c(e) \qquad \forall e \in E \tag{7}$$

$$y_i \geq 0 \qquad \forall i \in [k]. \tag{8}$$

Suppose that we root the tree at an arbitrary vertex r. Let $depth(v)$ be the number of edges on the path from v to r. Let $lca(s_i, t_i)$ be the vertex v on the path from s_i to t_i whose depth is minimum. Suppose that we use the primal-dual method to solve this problem, where the dual variable that we increase in each iteration corresponds to the violated (primal) constraint that maximized $depth(lca(s_i, t_i))$. After all primal constraints are satisfied, we do a "reverse cleanup" stage like in Steiner Forest, where we look at the edges we added in reverse order and remove them if we can do so while still having a feasible solution.

(b) (25 points) Prove that this is a 2-approximation. Hint: consider a path P_i where the dual variable is nonzero. How many edges in the final solution can be on the path from s_i to $lca(s_i, t_i)$, and how many can be on the path from t_i to $lca(s_i, t_i)$?

Proof: Let Y be the set output by the algorithm as a multicut of T. Since we use the primal-dual method, where the dual variable that we increase in each iteration corresponds to the violated constraint that maximized $depth(lca(s_i, t_i))$ during the algorithm, so (y_i) is still feasible. Additionally, if we remove Y from E, then there is no (s_i, t_i) path, otw, it means that when we considered $lca(s_i, t_i)$ there exists an edge that hasn't been saturated, so y_i could have been increased, this contradicts the algorithm. Therefore, (y_i) is a feasible dual solution.
By the definition of $lca(s_i, t_i)$, let consider two paths $T_1 = s_i \to v$ and $T_2 = v \to t_i$. Suppose two edges e and e' are picked from T_1, and assume, without loss of generality, that e appears before e' on T_1. Consider when e is tested during the reverse cleanup, since e is not discarded, so $\exists j \neq i$, (s_j, t_j) s.t. e is the only edge picked on the path $s_j \to t_j$. Let $u = lca(s_j, t_j)$, since $e' \notin s_j \to t_j$, so $depth(u) > depth(v)$ and u is processed before v. After u is processed, Y contains an edge e^* from $s_j \to t_j$. Moreover, since $y_i > 0$, e must be added during or after the iteration in which v is processed, hence, e must be added in Y after e^*. So e^* must be in Y when e is being tested for deletion. This contradicts the fact that at this moment e is the only edge of Y on the path $s_j \to t_j$. Therefore, for every pair (s_i, t_i) s.t. $y_i > 0$, Y contains at most 2 edges from the path $s_i \to t_i$.
Since each edge of Y is saturated, then for all $e \in Y$, we have $c(e) = \sum_{i:e \in P_i} y_i$. Additionally, by definition, $c(Y) = \sum_{e \in Y} c(e)$, yielding

$$c(Y) = \sum_{e \in Y} \sum_{i \in [k]} y_i \mathbb{I}_{e \in P_i} = \sum_{i \in [k]} y_i \sum_{e \in Y} \mathbb{I}_{e \in P_i} = \sum_{i \in [k]} y_i |Y \cap P_i| \leq 2 \sum_{i \in [k]} y_i. \tag{9}$$

Let (y^*_i) be an optimal solution of the dual. Since (y_i) is a feasible solution, we obtain $\sum_{i \in [k]} y_i \leq \sum_{i \in [k]} y^*_i \leq c(Y^*) \leq c(Y) \leq 2c(Y^*)$, i.e., this is a 2-approximation algorithm. ■