ORIE 634 Approximation Algorithms November 2, 2006

Problem Set 5

Due Date: November 16, 2006

1. In the *Steiner k-cut problem*, we are given an undirected graph $G = (V, E)$, costs $c_e \geq 0$ for all $e \in E$, a set of terminals $T \subseteq V$, and a positive integer $k \leq |T|$. The goal of the problem is to partition the vertices into k sets $S_1, \ldots, S_k$ such that each set contains at least one terminal (that is, $S_i \cap T \neq \emptyset$ for $i = 1, \ldots, k$) and to minimize the weight of the edges with endpoints in different parts. Given a partition $\mathcal{P} = \{S_1, \ldots, S_k\}$, let $c(\mathcal{P})$ be the total cost of the edges that have endpoints in different parts of the partition.

   Consider the following greedy algorithm for the Steiner k-cut problem: We start with $\mathcal{P} = \{V\}$. As long as $\mathcal{P}$ does not have k parts, we consider each set $S \in \mathcal{P}$ with $|S \cap T| \geq 2$, consider each pair of terminals, and compute the minimum-cost cut between that pair of terminals. We then choose the minimum-cost cut found overall by this procedure; note that this breaks some set $S \in \mathcal{P}$ into two pieces. We replace S in $\mathcal{P}$ with these two new parts, and continue.

   (a) Let $\mathcal{P}_i$ be the contents of the partition found by the algorithm when it has i parts. Let $\hat{\mathcal{P}} = \{V_1, V_2, \ldots, V_i\}$ be any valid partition into i parts (that is, $V_j \cap T \neq \emptyset$ for $j = 1, \ldots, i$). Show that

   $$c(\mathcal{P}_i) \leq \sum_{j=1}^{i-1} \sum_{e \in \delta(V_j)} c_e.$$

   (b) Use the above to show that the greedy algorithm is a $\left(2 - \frac{2}{k}\right)$-approximation algorithm for the Steiner k-cut problem.

2. In the sparsest cut problem, we are given an undirected graph $G = (V, E)$, costs $c_e \geq 0$ for all $e \in E$, and k pairs of vertices s_i, t_i with an associated demand d_i. We want to find a set of vertices S that minimizes

   $$\frac{\sum_{e \in \delta(S)} c_e}{\sum_{i:|S \cap \{s_i, t_i\}|=1} d_i}.$$

   That is, the sparsest cut problem finds a cut that minimizes the ratio of cost to separated demands. Let $\mathcal{P}_i$ denote the set of all paths P from s_i to t_i.

   (a) Prove that it is equivalent to find a minimum-cost set of edges F that minimizes

   $$\frac{\sum_{e \in F} c_e}{\sum_{i \in s(F)} d_i},$$

where $s(F)$ is the set of indices i such that s_i and t_i are not connected in the graph $(V, E - F)$.

(b) Prove that the following LP is a linear programming relaxation of the sparsest cut problem.

$$\begin{aligned} \text{Min} \sum_{e \in E} c_e x_e & \\ \sum_{i=1}^{k} d_i y_i &= 1 \\ \sum_{e \in P} x_e &\geq y_i \qquad \forall P \in \mathcal{P}_i, 1 \leq i \leq k \\ y_i &\geq 0 \qquad 1 \leq i \leq k \\ x_e &\geq 0. \end{aligned}$$

(c) Prove that the linear program can be solved in polynomial time.

(d) Let (x^*, y^*) be an optimal solution to the linear program, and suppose that $y_1^* \geq y_2^* \geq \cdots \geq y_k^*$. Let $D_i = \sum_{j=1}^{i} d_j$, and $H_n = 1 + \frac{1}{2} + \cdots + \frac{1}{n}$. Show that there exists i, $1 \leq i \leq k$, such that

$$y_i^* \geq \frac{1}{D_i \cdot H_{D_k}}.$$

(e) Use the preceding discussion to get an $O(\log k \cdot H_{D_k})$-approximation algorithm for the sparsest cut problem. Since $H_n = O(\log n)$, this is a $O(\log k \cdot \log D_k)$-approximation algorithm.

3. Let $C_n = (V, E)$ be a cycle on n vertices, and let d_{uv} be the distance between $u, v \in V$ on C_n. Show that for any tree metric (V, T) on the same set of vertices V, there must exist a pair of vertices $u, v \in V$ such that $d_{uv} = 1$, but $T_{uv} \geq n - 1$. To do this, suppose that of all trees T with optimal distortion, T has the minimum total length. Show that T must be a path of vertices of degree two, then conclude the statement above.

4. In the *minimum cut linear arrangement problem*, we are given an undirected graph $G = (V, E)$ and costs $c_e \geq 0$ for all $e \in E$. As in the minimum linear arrangement problem, a feasible solution is a one-to-one mapping f of the vertices to the numbers $\{1, 2, \ldots, n\}$. In this problem, however, the goal is to minimize the cost of the cut $\{f(1), \ldots, f(i)\}$ over all i; that is, we wish to minimize

$$\max_{1 \leq i < n} \sum_{e=(u,v): f(u) \leq i, f(v) > i} w_e.$$

Show that by using the balanced cut pseudo-approximation algorithm given in class, one can obtain an $O(\log^2 n)$-approximation algorithm for this problem.