Fall 2013, CS 583: Approximation Algorithms

Homework 5
Due: 11/18/2013

Instructions and Policy: Each student should write up their own solutions independently. You need to indicate the names of the people you discussed a problem with; ideally you should discuss with no more than two other people.

Solve as many problems as you can. I expect at least four.

Please write clearly and concisely - clarity and brevity will be rewarded. Refer to known facts as necessary. Your job is to convince me that you know the solution, as quickly as possible.

Problem 1 Consider a variant of the Bandwidth/Resource Allocation Problem from Spring 2009, Homework set 3. Suppose we are given a path P, with an *integer* capacity c_e for each edge $e \in P$. We are also given a set of demand requests $\mathcal{R} = \{R_1, R_2, \ldots R_m\}$. Each request R_i consists of a pair of vertices u_i, v_i (interpreted as a request for 1 unit of capacity from u_i to v_i), and a profit p_i. The goal is to route a maximum-profit set of feasible requests; a set of requests is feasible if, for each edge e, the total number of requests that pass through e is at most c_e.

1. Write a (natural) LP for the Resource Allocation Problem above with a variable x_i for each request R_i.

2. It is known that the matrix corresponding to the LP is totally unimodular and hence one can obtain an optimum integral solution via the LP. In this problem we will use a different approach to prove the optimality of the LP via iterated rounding. The goal is to prove that any basic feasible solution x to the LP there is some i such that $x_i = 1$. Prove this using the following hints.

   Hint 1: Let x be a basic feasible solution and $\mathcal{R}'$ be the set of requests with $x_i \in (0, 1)$, and $n = |\mathcal{R}'|$. Consider the set of tight constraints (edges) which determine x. There must be n of them and the rows associated with them must be linearly independent. Obtain a contradiction by using a counting argument to show that there can be at most $n - 1$ of them using the next hint.

   Hint 2: Let $e_1, \ldots, e_n$ be the tight edges from left to right where $e_i = (\ell_i, r_i)$. First, suppose that the entire path consists of tight edges. Prove that for each e_i there must be at least 2 demands from $\mathcal{R}'$ that start or end at ℓ_i and 2 demands that start or end at r_i; otherwise e_i will not be linearly independent from e_{i-1} and e_{i+1}, or e_i will not be tight; use the fact that capacities are integer valued and x_i are in $(0, 1)$.

   If not all path edges are tight, try to contract some edges on the path.

Problem 2 In the k-tree problem you are given an undirected edge-weighted graph $G = (V, E)$ with edge weights $c : E \to \mathbb{R}^+$ and an integer k. The goal is to find a tree $T = (V_T, E_T)$ in G of smallest edge weight ($\sum_{e \in E_T} c(e)$) such that $|V_T| \geq k$. Show that if there is an α-approximation for k-tree then there is an α-approximation for the Steiner tree problem. Recall that in the Steiner tree problem, the input is an edge-weighted graph $G = (V, E)$ and a set of terminals $S \subseteq V$; the goal is to find a tree T of minimum edge-weight that connects (contains) all the terminals S.

Problem 3 In this problem you will derive an $O(\log k \cdot \log n)$ approximation for the rooted k-Steiner-tree problem which is related to the previous problem. The input consists of an edge-weighted undirected graph $G = (V, E)$, a specified root vertex r and a set $S \subset V$ of terminals. The goal is to find a min-cost tree (V_T, E_T), a sub-graph of G, such that $r \in V_T$ and $|S \cap V_T| \geq k$. Obtain an approximation for this problem following the outline below.

- Consider the density variant of the problem where the goal is to find a tree $T = (V_T, E_T)$ rooted at r that minimizes the ratio $c(E_T)/|V_T \cap S|$. Write an LP relaxation for this problem using ideas similar to the one for Sparsest Cut (and Steiner tree). Using the scaling idea used to reduce Sparsest Cut to Multicut, reduce the density variant of k-Steiner-tree to the Steiner tree problem and obtain an $O(\log n)$ approximation. Recall that the Steiner tree LP has an integrality gap of 2.

  See Lecture 19 from spring 2009, and Prob 8.6 from the Shmoys-Williamson book to learn about the scaling idea mentioned.

- Use an approximation algorithm for the density variant above in an iterative greedy fashion to create a tree rooted at r with at least k terminals. What is the density of this tree when compared to the density of the optimal solution to the original k-Steiner-tree problem?

- If the tree you have in the previous step has many more than k terminals, *prune* it to have k' terminals where $k \leq k' \leq 2k$ such that the density of the resulting tree is not much worse than the tree you started with.

- How can you connect the pruned tree to the original root r without incurring too much cost? Assuming you know the optimal cost, can you preprocess the instance to ensure that this connection cost is not too much?

Problem 4 Prove that any ring metric isometrically embeds into ℓ_1.

Problem 5 Given a graph $G = (V, E)$ with edge-weights $c : E \to \mathbb{R}^+$, you wish to partition G into $G_1 = G[V_1], G_2 = G[V_2], G_3 = G[V_3]$ such that $\lfloor |V|/3 \rfloor \leq |V_i| \leq \lceil |V|/3 \rceil$ for $1 \leq i \leq 3$, and the cost of the edges between the partitions is minimized. Using an α-approximation for the sparsest cut problem, give a pseudo-approximation for this problem where you partition the graph into 3 pieces $G[V'_1], G[V'_2], G[V'_3]$ such that $|V|/c_2 \leq |V'_i| \leq |V|/c_1$ for some constants $1 < c_1 < c_2$ and the cost of the edges between the partitions is $O(\alpha)$OPT. What constants c_1, c_2 can you guarantee? Note that c_1 and c_2 should be *constants*, independent of the graph size. (Hint: this problem is similar to the one on partitioning into two pieces that is in Vazirani's book on applications of sparsest cut (Section 21.6.3).)

Problem 6 Problem 8.9 from Shmoys-Williamson book. This is to make you read Section 8.6 on buy-at-bulk network design.

Problem 7 Problem 11.6 from Shmoys-Williamson book.