Spring 2009, CS 598CSC: Approximation Algorithms
Homework 4
Due: 04/08/2009 in class

Instructions and Policy: You are not allowed to consult any material outside of the textbook and class notes in solving these problems. 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.

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

Problem 1 [30 pts] Consider a variant of the Bandwidth/Resource Allocation Problem from 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 [15 pts] Recall the simple rounding algorithm for Multiway-Cut that gave a 2-approximation using the metric relaxation: Pick a θ at random from $[0, 1/2)$ and output the cuts corresponding to the balls of radius θ around each terminal. Suppose we picked θ at random from $[0, 1)$. Give an example graph and a feasible LP solution (not necessarily optimal) such that the following is true: there is an edge e such that the probability of cutting e is at least cd_e for some $c > 2$; here d_e is the LP value on e.

Problem 3 [30 pts] We consider a problem that can be thought of as adding bells and whistles to the Multiway-Cut problem; let us call it MW-Cut-BW for short. The input is an undirected graph $G = (V, E)$ with non-negative edge costs c_e and k colors $\{1, 2, \ldots, k\}$. The goal is to assign each vertex a color to minimize the cost of the assignment, as defined below. For each vertex v there is a given cost $w(v, i)$ to assign color i to v. For an edge (u, v) there is a cost $c(u, v)$ that is paid if u and v are assigned different colors. Thus the goal is to find an assignment $f : V \to \{1, \ldots, k\}$ so as to minimize

$$\sum_{v \in V} w(v, f(v)) + \sum_{(u,v) \in E} c(u, v)[f(u) \neq f(v)].$$

We think of the first term as the color-assignment cost and the second term as the cut cost.

- Show the Multiway-Cut problem is a special case of MW-Cut-BW.
- Write an LP relaxation for MW-Cut-BW with variables $x(u, i)$ as we did for Multiway-Cut (the geometric relaxation).
- Consider the following randomized rounding algorithm based on the variables $x(u, i)$.
  1. No vertex is colored initially
  2. While there is a v that has not been assigned a color do
     (a) Pick a color i from $\{1, \ldots, k\}$ uniformly at random
     (b) Pick a number θ uniformly at random from $[0, 1]$
     (c) For each v that is unassigned, if $\theta \leq x(v, i)$, assign color i to v

  Note that a vertex is assigned a color only once. Prove the following facts to obtain a randomized 2-approximation for MW-Cut-BW.

  – With high probability, the algorithm terminates in $O(k \log n)$ iterations of the while loop.
  – The probability of u being assigned i is exactly $x(u, i)$.
  – The probability of an edge (u, v) being cut is at most $2d(u, v)$ where $d(u, v)$ is the distance assigned to (u, v) by the LP.

Problem 4 [25pts] Consider the multicut problem when the underlying graph is a tree. That is, we are given an edge-weighted tree $T = (V, E)$ (edge e has weight/cost c_e) and k node pairs $s_1t_1, s_2t_2, \ldots, s_kt_k$. The goal is find a minimum weight set of edges whose removal separates each pair s_it_i. Consider the LP relaxation for this problem that we discussed in class (also see Chapters 18 and 19 in Vazirani's book) which has variables d_e for each edge and constraints that state that the distance between s_i and t_i is at least 1 in the metric imposed by d.

- Suppose the tree is rooted at some vertex r and in this rooting, for each pair s_it_i, either s_i is an ancestor of t_i or t_i is an ancestor of s_i. Prove that the matrix obtained from the LP is a network matrix and hence totally unimodular (see class notes of Lecture 11 from Fall 2006 course website). **You can do the second part even if you cannot figure this part out.**

- Obtain a 2 approximation for the multicut problem on trees as follows. Root the tree arbitrarily. For each pair s_it_i let v_i be the least common ancestor of s_i and t_i (v_i could be one of s_i, t_i). Solve the LP to obtain an optimum feasible solution d^*. Transform the problem to the case discussed in the first part by separating s_i or t_i from v_i. How do you choose which of s_i or t_i to separate from v_i and how do you obtain a feasible LP solution to the transformed instance from d?