CS599 (Spring 2010) - Takehome Midterm
Due in class on Tuesday, 03/23

No late submission will be granted, so you may want to start early. Contrary to the policy on homeworks, you cannot discuss problems, ideas, or solutions to this midterm with anyone (in or outside of class) except myself. Also, as a reminder, you are not allowed to seek solutions to these problems online or elsewhere. If you are stuck on a problem and need hints, or you need help understanding a problem, you are welcome to e-mail me, or talk to me in person.

Problem 1

You are given a binary tree T with root r in which each edge e has a cost $c_e \geq 0$. In addition, each node v has a value $b_v \geq 0$. You are also given a total budget C. Your goal is to select a subtree $T' \subseteq T$ including the root (T' need not be binary, of course), such that $\sum_{e \in T'} c_e \leq C$. Your goal is to maximize the total value thus spanned, i.e., $\sum_{v \in T'} b_c$. Give and analyze an FPTAS for this problem.

Problem 2

The following type of graph is very useful in modeling wireless networks. All nodes are embedded in $\mathbb{R}^2$ (i.e., the plane), and two nodes u, v have an edge between them if and only if they are at distance at most 1 in the plane.

Give and analyze a constant factor approximation algorithm for the weighted independent set problem (i.e., maximizing $\sum_{v \in S} w_v$ over independent sets S) for this type of graph. Your approximation guarantee c should satisfy $c \geq 0.1$. (Hint: a fairly simple greedy algorithm works.)

Problem 3

In our approximation algorithm for Minimum Multi-Cut, we chose — for each ball — a radius $r \in [0, \frac{1}{2})$ minimizing the surface-to-volume ratio. Instead, we could have chosen $r \in [0, \frac{1}{2})$ independently and uniformly at random.

(a) Prove that this algorithm would give an $O(k)$-approximation in expectation, where k is the number of source-sink pairs.

(b) Give an instance and a fractional LP solution where this type of rounding would increase the cost by a factor $\Omega(k)$. (Hint: your fractional solution for the instance need not be optimal, and in fact, it is probably much easier to prove it for a suboptimal — but feasible — solution.)

Problem 4

Here is a variant of the Set Cover problem. As before, you have a universe U (of size n), and subsets $S_1, \ldots, S_m \subseteq U$. Your goal is to select some of these subsets to cover as many of the elements of the universe *exactly once*. In other words, covering an element twice or more is just as bad as not covering it at all. There is no constraint on how many or how few sets you select. Formally, we define

$$c_e(\mathcal{C}) \quad := \quad \begin{cases} 1 & |\{S \in \mathcal{C} \mid e \in S\}| = 1 \\ 0 & \text{otherwise} \end{cases}$$

The objective is now to find a collection $\mathcal{C}$ of sets maximizing $\sum_e c_e(\mathcal{C})$. The following subproblems should guide you toward an $O(\log m)$ approximation for this problem.

(a) Show that if for each element $e \in U$, the number of sets S_j containing e is the same — say, f — then by including each set S_j independently with some probability p, you cover a constant fraction of all elements.

(b) Show that this analysis extends to the case when each element is contained in a number of sets between $f/2$ and f, for some f.

(c) Give an $O(\log m)$ approximation for the general problem. (Hint: some insights from the previous subproblems may be useful.)