CS599 (Spring 2007) - Takehome Final

Due in my office by Monday, 05/07, noon

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 final 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

Give a primal-dual 2-approximation algorithm for the first problem from the midterm. For your convenience, it is restated here:

> You are given a universe U of n elements. For each $i = 1, \ldots, m$, you are given two subsets $S_i, T_i \subseteq U$. Your goal is to select at least one of S_i, T_i for each i, while minimizing the size of the union of the sets selected.

(Hint: Use the LP from the midterm sample solution. You won't need the full-blown dual-raising machinery. I recommend coming up with a simple way to first make the x_e constraints all tight, and taking it from there.)

Problem 2

In class, we talked (briefly) about graph bisections and b-balanced cuts. Here is a somewhat different version: You are given an undirected graph $G = (V, E)$ with non-negative edge capacities $c_e \geq 0$, and a source s and sink t. Your goal is to find a minimum capacity s-t cut $(S, \overline{S})$ such that $|S|$, the number of nodes on the s-side, is at most a given number k.

In the spirit of problem 3 from the midterm, you are asked to give an approximation algorithm comparing yourself against a different OPT. Specifically, let OPT(k) denote the minimum capacity of a cut that has at most k nodes on the s-side. You are to find a cut $(S, \overline{S})$ with at most $2k$ nodes on the s-side, such that $c(S, \overline{S}) \leq 2\text{OPT}(k)$. That is, you get to put twice as many nodes on the s-side, and cut twice as much total capacity.

Use LP-rounding to give such an approximation algorithm. (Hint: (1) think about how to express whether a node is on the s-side or the t-side. (2) in rounding the LP, choose a radius to cut a ball according to some useful distribution.)

Problem 3

Having spent a lot of time on low-distortion embeddings, let's look at some concrete examples. For each of the following two classes of examples, decide whether they can be embedded into a (sufficiently high-dimensional) ℓ_1 space with distortion 1. Either give a proof, or a counter-example with a proof that it is a counter-example.

(a) All cycles with n nodes and unit edge lengths (technically, the shortest path metric induced by a cycle).

(b) All complete bipartite graphs with m nodes on one side, n nodes on the other, and unit edge lengths (again, technically, the induced shortest path metric).

Problem 4

Recall that in Graph Coloring, you are given an undirected graph $G = (V, E)$ (which in our case will be allowed to contain parallel edges), and to assign at most k colors to the vertices such that each pair of adjacent vertices (u, v) has different colors. There are many different objectives, such as minimizing the number of colors. Here, however, we will look at a different objective: you are supposed to use only 3 colors, and the goal is to maximize the number of edges such that the endpoints have different colors. This is still NP-complete, as the graph is 3-colorable iff all m edges have different colors.

(a) [3 points] Give and analyze a 2/3 approximation algorithm for this problem (Hint: Johnson's Algorithm).

(b) [7 points] Prove that there is a constant $\gamma > 0$ such that no polynomial-time algorithm can be better than a $1 - \gamma$ approximation for this problem. (Hint: Find a suitable problem to reduce from. You are explicitly allowed, in fact encouraged, to look up in Kleinberg/Tardos or CLRS the standard reduction proving NP-hardness of 3-Coloring.)