Spring 2009, CS 598CSC: Approximation Algorithms
Homework 6
Due: 05/06/2009 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.

Embedding finite metrics into trees: For Problems 1 and 2 we will use probabilistic tree embeddings. You can assume that there is a randomized algorithm that, given a n point finite metric on a set V, gives a tree T on V such that:

- $d_T(uv) \geq d_G(uv)$ for all $u, v \in V$.
- $E[d_T(uv)] \leq \alpha(n) d_G(uv)$ for all $u, v \in V$.

We showed $\alpha(n) = O(\log n)$ in class, but we will leave it as a a parameter in the two problems.

Problem 1 [25 pts] Consider the (uniform) buy-at-bulk problem. In this problem we are given an undirected edge-weighted graph $G = (V, E)$ and k pairs of nodes $s_1t_1, s_2t_2, \ldots, s_kt_k$. Each pair has a non-negative demand d_i. Further we are given some p cable types; cable type i has a capacity u_i and cost c_i. Assume that $u_1 < u_2 < \ldots < u_p$ and $c_1 < c_2 < \ldots < c_p$. These cable types exhibit economies of scale, meaning that the higher capacity cables are cheaper per unit capacity; formally, $c_i/u_i < c_{i-1}/u_{i-1}$ for $1 < i \leq p$. The goal is to select, for each pair s_it_i, a path P_i that connects them and route the pair's demand d_i on P_i. The total flow routed on an edge e is $f(e) = \sum_{i:e \in P_i} d_i$. To support this flow we need to install cables on the edges; on edge e the total capacity of the cables installed should be at least $f(e)$. Note that multiple copies of a cable might need to be installed for this purpose. To install a cable of type i on an edge e it costs $\ell(e)c_i$ where $\ell(e)$ is the length of the edges e. The problem involves finding the paths P_i and installing the necessary cables to minimize total cost.

1. Suppose you are given the paths P_i and the only goal is to install cables of cheapest cost to support the flow on the paths. Give a constant factor approximation to minimize the cost of installing cables. Note that you can do this separately for each edge e. (Why is this problem hard?).
2. Use part 1 and the tree embedding result to obtain a randomized $O(\alpha(n))$ approximation to the problem. Carefully argue why you can use the tree embedding result.

Problem 2 [30 pts] Let $T = (V, E)$ be a tree on V and let $w : E \to \mathcal{R}^+$ denote edge lengths on T. The shortest path distances on T induce a metric d_T on V. Prove that d_T is an ℓ_1 metric. Use this and the tree embedding result to show that any n point metric can be embedded into ℓ_1 with $\alpha(n)$ distortion. Is the resulting embedding a contraction or an expansion? Give a randomized approximation algorithm with ratio $\alpha(n)$ for the sparsest cut problem in general graphs using this observation: Use the LP relaxation we discussed in class and the tree embedding. Why do you not need to compute the embedding explicitly?

Problem 3 [30 pts] Consider MAX-CUT with the additional constraint that specified pairs of vertices be on the same/opposite sides of the cut. Formally, we are given two sets of pairs of vertices, S_1 and S_2. The pairs in S_1 need to be separated, and those in S_2 need to be on the same side of the cut sought. Under these constraints, the problem is to find a maximum-weight cut.

1. Give an efficient algorithm to check if there is a *feasible* solution.
2. Assuming there is a feasible solution, give a strict quadratic program and vector program relaxation for this problem. Show how the algorithm for MAX-CUT we saw in class can be adapted to this problem while maintaining the same approximation ratio.

Problem 4 [15 pts] Given an $n \times n$ matrix A, a *principal submatrix* of A is a square submatrix obtained by picking a set of indices $S \subseteq \{1, 2, \ldots, n\}$, and discarding the rows *and* columns of A indexed by S. For instance, the principal submatrix of A corresponding to $S = \{1, 4, 5\}$ is the $(n-3) \times (n-3)$ matrix obtained from A by discarding rows $1, 4, 5$ and columns $1, 4, 5$.

Prove that *all* the principal submatrices of a positive semidefinite matrix are also positive semidefinite. (Which characterization of positive semidefinite matrices can you use?) Conclude that if a matrix A is positive semidefinite, the determinant of any principal submatrix of A is nonnegative.

(*Note:* The converse is also true, though you do not have to prove it: If the determinants of all principal submatrices of a real symmetric matrix A are nonnegative, A is positive semidefinite.)