

Spring 2011, CS 598CSC: Approximation Algorithms

Homework 6

Due: 05/9/2011 to Chandra

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 two for this homework.

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 Let $T = (V, E)$ be a tree on V and let $w : E \rightarrow \mathcal{R}^+$ 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 2 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 3 Given an $n \times n$ matrix A , a *principal submatrix* of A is a square submatrix obtained by picking a set of indices $S \subseteq \{1, 2, \dots, n\}$, 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.)

Problem 4 In this problem you will consider the *node-weighted* Steiner tree problem. The input consists of an undirected graph $G = (V, E)$ and a subset $S \subseteq V$ of terminals. Each node v has a non-negative weight $w(v)$. The goal is to find a minimum weight subset of nodes S' such that the subgraph induced on those nodes $G[S']$ connects all terminals. One can equivalently phrase it as finding a tree T in G that contains all the terminals with the goal of minimizing the weight of the nodes in T . It is useful to assume that the terminals have zero weight since they have to be included anyway.

- Show that an $\alpha(|S|)$ -approximation for the above problem implies an $\alpha(n)$ -approximation for the set cover problem on n elements.
- Derive an $O(\log |S|)$ -approximation as follows. A *spider* is a tree in which at most one node has degree more than 2. For example a path is a spider. If a spider is not a path then let v be the node with degree strictly more than 2. Then the spider consists of paths P_1, \dots, P_k that are node-disjoint except at v . Given a spider let its density be the ratio of the weight of its nodes to the number of terminals it contains.
 - Let T^* be a tree that contains the terminals. Then show that there is a spider in T^* of density at most $w(T^*)/|S|$.
 - Show how one can compute a minimum density spider in polynomial time.
 - Combine the above two to derive the $O(\log n)$ -approximation.