Fall 2006, CS 598CC: Approximation Algorithms
Homework 1
Due: 9/26/2006 in class

Instructions and Policy: You are not allowed to consult any material outside of the textbook and class notes in solving these problems. Unless otherwise noted you are to solve the problems on your own. Even when collaboration is allowed, each person 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 me that you know the solution, as quickly as possible.

For this homework collaboration is allowed for problems 6 to 10.

Problem 1 [5 pts] Given an undirected graph G with n vertices and m edges, let $d = 2m/n$ be the average degree of a vertex. Give an $O(d)$ approximation for the maximum independent set problem. What does this imply for planar graphs?

Problem 2 [10 pts] Exercise 3.4 from Vazirani's book on TSP-Path.

Problem 3 [10 pts] Let $G = (V, A)$ be a directed graph with arc weights $c : A \to \mathcal{R}^+$. Define the density of a directed cycle C as $\sum_{a \in C} c(a)/|V(C)|$ where $V(C)$ is set of vertices in C. A cycle with the minimum density is called a minimum mean cycle and such a cycle can be computed in polynomial time (how?). Consider the following algorithm for ATSP. Given G (with c satisfying asymmetric triangle inequality) compute a minimum mean cycle C. Pick an arbitrary vertex v from C and recurse on the graph $G' = G[V - C \cup \{v\}]$. A solution to the problem on G can be computed by patching C with a tour in the graph G'. Prove that the approximation ratio for this heuristic is at most $2H_n$ where $H_n = 1 + 1/2 + \ldots + 1/n$ is the nth harmonic number.

Problem 4 [10 pts] Considier the k-dimensional knapsack problem. We are given n non-negative k-dimensional vectors $\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_n$. Each vector has a non-negative weight, w_i for $\bar{v}_i$. We are also given a k-dimensional knapsack $\bar{V}$ and the goal is to find a maximum weight subset of vectors that pack in to V. A subset of vectors pack into $\bar{V}$ if their vector addition is less than $\bar{V}$ (co-ordinate wise). Prove that there is a pseudo-polynomial time algorithm for this problem when k is a fixed constant independent of n. What is the running time of your algorithm? Prove that even for $k = 2$ and *unit weights* the problem does not admit an FPTAS. (Hint: use a reduction from the Partition problem. For an item a_i consider a vector $(a_i, A - a_i)$ for some large enough A.) Extra credit: obtain a PTAS for this problem when k is a fixed constant independent of n (might need to use LP techniques).

Problem 5 [10 pts] Exercise 10.2 in Vazirani's book.

Problem 6 [15 pts] Consider a budgeted version of the maximum coverage problem. We are given m sets $S_1, S_2, \ldots, S_m$, each a subset of a set $\mathcal{U}$. Each set S_i has a non-negative cost c_i and we are also given a budget B. The goal is to pick sets of total cost at most B so as to maximize the number of elements covered. Show that if $c_i \leq \epsilon B$ for $1 \leq i \leq m$ then the Greedy algorithm yields a $1 - 1/e - \epsilon$ approximation (for sufficiently small ϵ). For any fixed $\epsilon > 0$ obtain a $1 - 1/e - \epsilon$ approximation. (Hint: consider the PTAS for knapsack from the lectures. You might find the inequality $1 - x \leq e^{-x}$ useful.) Extra credit: Obtain a $1 - 1/e$ approximation for this problem.

Problem 7 [20 pts] We are given a subsets $S_1, S_2, \ldots, S_m$ of a set U and an integer k. Our goal is to pick k sets to maximize some function of the elements that are covered by the chosen sets. In maximum coverage the objective was simply the number of elements covered. Consider now a more general objective function. Each element $e \in U$ has a weight function $w_e : \{1, 2, \ldots, m\} \rightarrow \mathcal{R}^+$. In other words $w_e(i)$ is the weight obtained by covering e by set S_i. The objective is now to pick k sets to maximize the weight of the elements covered. Note that once we pick k sets, an element e that is covered by more than one set will be assigned to the set that gives it the most weight. Let us formalize this. Define a function f on subsets of $\{1, 2, \ldots, m\}$ as follows. For $X \subseteq \{1, 2, \ldots, m\}$ we let $U_X = \cup_{i \in X} S_i$. We then define $f(X) = \sum_{e \in U_X} \max_{i \in X} w_e(i)$. Note that the problem we have defined is $\max_{X \subseteq \{1,2,\ldots,m\}, |X|=k} f(X)$. Prove that f is a submodular function. What is the greedy algorithm for this problem? Conclude that the greedy algorithm yields a $1 - 1/e$ approximation.

Problem 8 [20 pts] Exercise 5.4 in Vazirani's book.

Problem 9 [20 pts] For Metric-TSP consider the nearest neighbour heuristic discussed in class. Prove that the heuristic yields an $O(\log n)$ approximation. (Hint: use the basic idea in the online greedy algorithm for the Steiner tree problem). Also give examples to show that there is no constant c such that the heuristic is a c-approximation algorithm.

Problem 10 [20 pts] Multi-processor scheduling: given n jobs $J_1, \ldots, J_n$ with processing times $p_1, p_2, \ldots, p_n$ and m machines $M_1, M_2, \ldots, M_m$.

- For identical machines show that greedy list scheduling that orders the jobs in non-increasing sizes has an approximation ratio of $3/2$. Extra credit for a bound of $4/3$.
- Now consider the problem where the machines are not identical. Machines M_j has a speed s_j. Job J_i with processing time p_i takes p_i/s_j time to complete on machine M_j. Give a constant factor approximation for scheduling in this setting to minimize makespan (the maximum completion time). (Hint: consider jobs in decreasing sizes. Assuming $p_1 \geq p_2 \geq \ldots \geq p_n$ and $s_1 \geq s_2 \geq \ldots s_m$, show that $OPT \geq \max_{i \leq m}(\sum_{j \leq i} p_i / \sum_{j \leq i} s_i$.)