Assignment 3

October 9, 2013

Due date: 23rd October Marks: 20

1. **Unique set cover problem:** We are given a set of n elements and a collection of m subsets of the elements. The goal is to pick out a number of subsets so as to maximize the number of *uniquely covered* elements, those that are contained in exactly one of the picked subsets. (Note that the cost or number of subsets picked is not important.)

   (a) Consider the following algorithm: for some number $p < 1$, the algorithm picks each subset independently with probability p. Assuming that every element is contained in exactly F subsets, compute the expected number of uniquely covered elements. For what value of p is this expectation maximized?

   (b) Assuming that each element is contained in at least $F/2$ subsets and at most F subsets, give a constant factor approximation to unique set cover using the algorithm in part 1a.

   (c) Extend the algorithm to obtain an $O(\log m)$ approximation in general, without assumptions on the frequency of any element. (*Hint: Try reducing this case to the case mentioned in part 1b.*)

   (d) Can you improve the approximation from part 1c to a factor of $O(\log n)$? (*Hint: Can you limit the number of sets under consideration to only n?*)

2. **Multidimensional Knapsack problem:** Our goal is show that the multidimensional knapsack problem (as defined in the midsem paper) is NP-hard even for two dimensions even when all the profit values are 1. We will reduce partition problem (where, given a set of numbers $a_1, \ldots, a_n$, the goal is to determine whether the set can be partitioned into two subsets such that the sum of the numbers in the two sets is equal.) to multidimensional knapsack.

   (a) Reduce the partition problem to *the equipartition problem* (which is similar to the partition problem, but the two partitions are restricted to contain exactly $n/2$ elements each).

   (b) Now reduce the equipartition problem to 2-dimensional knapsack with unit profit values as follows: A vector v_i in the 2-dimensional knapsack instance has first coordinate a_i, which is the ith number in the equipartition instance. Second coordinate is $a_{\max} - a_i$, where $a_{\max} = \max_i a_i$. Complete the reduction by setting appropriate capacity vector.

   (c) Show that an FPTAS for the 2-dimensional knapsack problem can be used to obtain an exact solution to the partition problem.

3. Exercise 12.1 from WS.

4. Exercise 1.4 from WS.

5. Recall the k-center problem: We have an undirected complete graph G with vertex set V, and non-negative edge costs d_{ij} which form a metric. The goal is to find a subset C of vertices with $|C| = k$ that minimizes

$$\max_{v \in V} d(v, C) \tag{1}$$

   Here $d(v, C) = \min_{c \in C} d(v, c)$. Let D^* be this minimum value when $|C| = k$ (i.e. D^* is the value of the optimal solution.)

   Now consider a variant: We insist that the distance of every vertex from its closest center is at most D^* and want to minimize the number of centers needed to achieve this. In other words, we relax the condition on the number of centers instead of relaxing the distance guarantee.

   Design an algorithm that places $|C| = O(k \log |V|)$ centers and ensures that (1) is at most D^*.