600.363/463 Solution to Homework 11
Posted: 12/2/03
Due: Wednesday, 12/5/03

1. Weighted set-covering problem

In the original set-covering problem, it assumes the cost of adding each set into the set cover is 1. Now let's consider another case. For each set S, the cost of adding it into $\mathcal{C}$ is $n = 2^k$. The cost of each set is known. The optimal cover is the cover that has minimum cost. Provide an approximation algorithm of this weighted set-coving problem. How well does this algorithm work?

Solution:

We can adapt the problem as follows:
For each vertex v, we find the maxium cost of all sets: $n = \max_{S\in(F)}(weightofS)$. We then divide every vertex v into n subvertices. For a set S that has a weight of i, it contains $2^{\lg n - \lg i}$ subvertices of every vertex it contains. So the revised set S' has $|S| \cdot 2^{\lg n - \lg i}$ vertices. When executing the GREEDY-SET-COVER algorithm, whenever a subvertex of a vertex is included, all the subvertices of this vertex is markered as "'contained"'.
With the similar proof in the text, this algorithm is a $\rho(n) = H(\max\{|S| \cdot weightofS : S \in (F)\}$ approximation algorithm.

2. Broadcast in Radio

A radio network is composed of many radio stations. Each station is connected to some other stations. When a station delievers a message, only the stations connected to it can get this message. Now we want to broadcast a message to all the stations. We want to minimize the number of devileries required to make all stations get this message. Give an approximation algorithm that can solve this problem. (*Hint:* use the greedy-set-cover algorithm.)

Solution:

Think of this problem as a set-covering problem. Let X be the set of all the stations. For each station v_i, the set $S_i = \{v_i\} + \{neighborsofv_i\}$. We call v_i the center of set S_i. $\mathcal{F}$ is composed of

all these sets. So $\mathcal{F}$ has $|X|$ sets.
So we can use the greedy-set-cover algorithm to find a set cover. For any two sets S_i and S_j in the resulting cover such that $S_i \cap S_j \neq \emptyset$, we can deliver the message from a vertex in S_i to the center of S_i, from the center of S_i to any vertex $v \in S_i \cap S_j$, from this vertex to the center of set S_j, and finally from center of S_j to all the other vertices in set S_j. So the number of deliveries is $2\cdot$ *size of set cover*.

3. 2-approximation Steiner Tree (35.2-3 in text)

Consider the following ***closest-point heuristic*** for building an approximate traveling-salesman tour. Begin with a trivial cycle consisting of a single arbitrarily chosen vertex. At each step, identify the vertex u that is not on the cycle but whose distance to any vertex on the cycle is minimum. Suppose that the vertex on the cycle that is nearest u is vertex v. Extend the cycle to include u by inserting u just after v. Repeat until all vertices are on the cycle. Prove that this heuristic returns a tour whose total cost is not more than twice the cost of an optimal tour.

Solution:

Let's denote the optimal tour at the step i as H_i^*, and the tour produced by the heuristic as H_i. Suppose the vertex on the cycle that is nearest u_i is vertex v_i, when adding vertex u_i. Since the cost function satisfies the triangle inequality, it is easy to get: $c(H_i) \leq c(H_{i-1} + 2c$. So $c(H_i) \leq 2\sum_i c(u_i, v_i)$.

Besides, we may notice that the way nodes and edges are added in closest-point heuristic is exactly the same as Prim's algorithm. So the cost of the MST produced by Prim's algorithm is equal to $\sum_i c(u_i, v_i)$.

In the text it is proved that: $c(MST) \leq c(H^*)$, so we have: $c(H) \leq 2c(MST) \leq 2c(H^*)$. So it is proved.