Cmpt-405/705

March 11, 2016

Assignment 8

Problem 1

The TSP algorithm is to find a complete hamilton cycle with minimum cost in a weighted undirected graph $G = (V, E)$. In the following we assume the edge weights $c(u, v)$ are nonnegative integers, and the weight function satisfies the following inequality: $c(u, w) < c(u, v) + 2 * c(v, w)$ for all $u, v, w \in V(G)$ (NOTE: it's no longer the triangle inequality). Use the following polynomial approximation algorithm to solve this problem. Give the approximation ratio of the algorithm (or find the best approximation ratio as you can).

`Approx-TSP-TOUR(G,c)`

1. `Select any vertex` $v \in V(G)$ `as "root" vertex`
2. `Compute a Minimum Spanning Tree (by Kruskal or Prim)` T `from` G `from root` v
3. `Let` L `be the list of vertices visited in a preorder tree walk of` T
4. **`return`** `the hamilton cycle` H `that visits the vertices in the order` L

Problem 2

Show that if $P = NP$ then every language in NP other than two of them are NP-complete. Which two are not NP-complete?

Problem 3

Suppose you are given a set of positive integers $A = \{a_1, a_2, \ldots, a_n\}$ and a positive integer B. A subset $S \subseteq A$ is called *feasible* if the sum of the numbers in S does not exceed B:

$$\sum_{a_i \in S} a_i \leq B.$$

The sum of the numbers in S will be called the *total sum* of S. You would like to select a feasible subset S of A whose total sum is as large as possible. **Example.** If $A = \{8, 2, 4\}$ and $B = 11$, then the optimal solution is the subset $S = \{8, 2\}$.

Here is an algorithm for this problem.

- Initially $S = \phi$.
- Define $T = 0$.

- For $i = 1, 2, \ldots, n$
  - If $T + a_i \leq B$ then $S \leftarrow S \cup a_i$ and $T \leftarrow T + a_i$.

1. Give an instance in which the total sum of the set S returned by this algorithm is less than half the total sum of some other feasible subset of A.
2. Give and analyse a polynomial-time approximation for Problem 6 with the following guarantee: It returns a feasible set $S \subseteq A$ whose total sum is at least half as large as the maximum total sum of any feasible set $S' \subseteq A$. Your algorithm should have a running time of at most $O(n \log n)$.

Problem 4

In the Knapsack problem a kapsack of capacity C and n items are given. Every item i ($1 \leq i \leq n$) has weight w_i and value v_i. The problem is now to choose items for the knapsack such that their value is maximixed but the total weight of the chosen items is smaller than C.

Consider the following algorithm for the Knapsack problem. The value-weight ration of item i is defined as v_i/w_i. Order the items using their value/weight ratio. Consider the items in decreasing order an put every item into the knapsack it it still fits.

What is the approximation ratio of this algorithm? Show the result formally.