22 SPRING CSCE 629 600: ANALYSIS OF ALGORITHMS - Homework 11

Name: Rohan Chaudhury

UIN: 432001358

Question (1) Textbook page 1117, Exercise 35.2-3.

Consider the following closest-point heuristic for building an approximate traveling-salesman tour whose cost function satisfies the triangle inequality. 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.

ANSWER:

Let us consider the following:

1. The optimal tour at a certain step i is Xi^*, and the entire optimal tour is X^*
2. The tour produced by the heuristic defined in the question at step i is Xi and the entire tour is X

Now, let us suppose that the vertex on the cycle that is nearest to ui is vi when we are adding ui in the ith iteration. Let us also suppose that before the insertion of ui to the tour, vi was connected to wi. So, the insertion of ui to the tour caused the addition of the edges {vi,ui} and {ui, wi} to the tour and the removal of the edge {vi, wi} from the tour. According to the triangle inequality we have:

$$c(ui, wi) \leq c(vi, wi) + c(vi, ui) \text{ [where c() denotes cost]}$$

$$\Rightarrow c(ui, wi) - c(vi, wi) \leq c(vi, ui)$$

Adding c(vi, ui) to both sides we get:

$$\Rightarrow c(vi, ui) + c(ui, wi) - c(vi, wi) \leq 2*c(vi, ui) \text{ ------- (1)}$$

The total increase of the cost of the tour during an iteration i is given by: c(vi, ui) + c(ui, wi) - c(vi, wi). So from equation (1) above we can say that, the total increase of the cost of the tour during an iteration i is at most 2*c(vi, ui). The total cost of the output tour after already choosing the first vertex is at most:

$$2 * \sum_{i=2}^{n} c(vi, ui)$$

$$\text{i.e., } c(X) \leq 2 * \sum_{i=2}^{n} c(vi, ui)$$

(c(X) is the cost of the tour returned by the heuristic defined in the question and $c(X^*)$ is the cost of an optimal tour)

Here n is the total number of vertices. Now, for the closest-point heuristic defined in the question, we can see that the sequence in which the vertices are added to the tour using the heuristic is the same as the sequence in which Prim's algorithm would add the vertices toward constructing a Minimum Spanning Tree (MST) when starting from the same initial vertex but at the total cost of $\sum_{i=2}^{n} c(vi, ui)$, (using Prim's algorithm, an MST can be found by repeatedly finding the vertex closest to the vertices already selected and then adding it to the tree where the new vertex will be added adjacent to the closest vertex to it among the already selected vertices.) so,

$$c(MST) = \sum_{i=2}^{n} c(vi, ui).$$

Now, we also know that the cost of constructing an MST lower bounds the cost of the optimal tour, i.e, $c(MST) \leq c(X^*)$,

So now we have the following:

$$c(X) \leq 2 * \sum_{i=2}^{n} c(vi, ui)$$

$$\Rightarrow c(X) \leq 2 * c(MST) \text{ [as } c(MST) = \sum_{i=2}^{n} c(vi, ui)]$$

$$\Rightarrow c(X) \leq 2 * c(MST) \leq 2 * c(X^*), \text{ [as } c(MST) \leq c(X^*)]$$

$$\Rightarrow c(X) \leq 2 * c(X^*), \text{ this was the required proof.}$$

Hence this heuristic returns a tour whose total cost is not more than twice the cost of an optimal tour.

Question (2) Textbook page 1127, Exercise 35.4-2.

The MAX-CNF satisfiability problem is like the MAX-3-CNF satisfiability problem, except that it does not restrict each clause to have exactly 3 literals. Give a randomized 2-approximation algorithm for the MAX-CNF satisfiability problem.

ANSWER:

Let us assume the following:

1. Each clause contains at most 1 instance of each literal
2. The clause doesn't contain both a literal and its negation

Now we will assign each variable 1 and 0 with an equal probability of 1/2. That is, each variable will be assigned 1 with probability 1/2 and 0 with probability 1/2.

Let us consider a random variable Xi which takes on the value 1 if clause i is satisfied and 0 if clause i is not satisfied. Now, if there are k number of literals (k≥1) in a clause i, then the probability that the clause i fails to be satisfied is $P(\overline{Xi}) = (1/2)^k$ which is ≤1/2. So, the probability that the clause i is satisfied is given by $P(Xi) = 1 - P(\overline{Xi})$ which is ≥ 1/2 as $P(\overline{Xi})$ is ≤1/2. If n is the total number of clauses then,

Expected number of satisfied clauses = C ≥ n/2

Let C^* be the optimal number of clauses satisfied by an optimal solution. Then we have,

$C^* \leq n$

Then we have, $C^*/C \leq n/(n/2)$

$\Rightarrow C^*/C \leq 2$

As $C^*/C \leq 2$, therefore this is a randomized 2-approximation algorithm for the MAX-CNF satisfiability problem.