Scheduling Theory **CO 454, Spring 2009**

Lecture 14: The Traveling Salesman Problem

June 18th, 2009

In the traveling salesman problem (TSP), we have n-cities which we think as the vertices of a complete graph, G, and there are costs $d_{i,j}$ on edge (i,j). We saw in HW3 that approximating the TSP to any β-factor is NP-hard. In this lecture, we give an approximation algorithm when the costs on the edges induce a metric.

Definition 0.1. The costs d_{ij} induce a metric if for any three cities i,j,k, we have $d(i,k) \leq d(i,j) + d(j,k)$.

Claim 0.2. *If d induces a metric, then for any two cities i and j, and any path $i, i_1, \ldots, i_k, j$ from i to j, we have $d(i,j) \leq d(i,i_1) \sum_{\ell=1}^{k-1} d(i_\ell, i_{\ell+1}) + d(i_k, j)$.*

Proof. We will prove by induction on the length of the path. For a path as in the claim, note that the definition of metric property implies that this is true for paths of length 2. Suppose the lemma is true for all paths of length $\leq k$, and now consider a path from i to j, $(i, i_1, \ldots, i_k, j)$ of length $k+1$.

By induction, we have $d(i_1, j) \leq \sum_{\ell=1}^{k-1} d(i_\ell, i_{\ell+1}) + d(i_k, j)$. By metric property, we have $d(i,j) \leq d(i,i_1) + d(i_1, j)$ Adding the two proves the claim. $\square$

If the cost d is a metric, then the TSP instance is called a metric TSP instance. We will give a constant factor approximation for the metric TSP problem.

1 Factor 2 approximation for Metric TSP

Before going into the algorithm, let us discuss a lower bound on the cost of the optimum tour. Let us denote the cost of the optimum tour as OPT. Let C^* be the optimum tour. In particular, let C^* be the edges in the tour. This will be our convention throughout. We will use the following short hand. Given a set of edges $F \subseteq E$,

$$d(F) = \sum_{(i,j) \in F} d(i,j)$$

Thus, $OPT = d(C^*)$.

Let (i,j) be any edge in C^*. Note that $C^* \setminus (i,j)$ is a tree (in fact it is a path) which spans all vertices of G.Therefore,

$$OPT \geq d(C^* \setminus (i,j)) \geq MST(G,d)$$

that is, the total cost of the minimum cost spanning tree is a lower bound on the cost of the optimum tour.

Before going on to describe the algorithm, we will need another definition.

Definition 1.1. Given a graph $H(V, F)$, an *Eulerian tour* of H is a closed *walk* from a vertex v back to itself which covers every edge of H exactly once. Recall that a walk is a sequence $v_1, v_2, \ldots, v_k$ such that each (v_i, v_{i+1}) is an edge, and v_i and v_j might be the same for distinct i and j.

Eulerian tours are well behaved as the following theorem from graph theory shows us.

Theorem 1.2. *A graph $H(V, F)$ has an Eulerian tour if and only if all the vertices in H have even degree. In the latter case, an Eulerian tour can be found in polynomial time.*

Now we are ready to state the algorithm

1. Let T be the minimum spanning tree of (G, d).
2. Obtain graph $H(V, F)$ by taking two parallel copies of each edge in T. Observe that in H, each vertex has even degree.
3. Find an Eulerian tour in $H(V, F)$. Let the tour be $v_1, v_2, \ldots, v_k, v_1$.
4. From the Eulerian tour obtain a traveling salesman tour by *shortcutting.* That is, the tour is $v_1, v_2, \ldots, v_n, v_1$ where duplicate entries of all vertices (except the last entry of v_1) in the Eulerian tour have been removed.

Theorem 1.3. *Let C be the tour obtained from the above algorithm. $d(C) \leq 2OPT$.*

Proof. Note that if F is the set of edges in H, then $d(F) = 2d(T) \leq 2OPT$. Furthermore, the total cost of the Eulerian tour is precisely $d(F)$. Therefore, to prove the theorem, it suffices to prove that shortcutting can only decrease the cost.

To see that, consider a short cutting operation. Suppose some portion of the Eulerian tour, $v_i, v_{i+1}, \ldots, v_j$ is shortcutted to v_i, v_j in the tour because all the intermediate vertices were visited before. But we know from Claim 0.2 that

$$d(v_i, v_j) \leq d(v_i, v_{i+1}) + d(v_{i+1}, v_{i+2}) + \cdots + d(v_{j-1}, v_j)$$

Thus, shortcutting from v_i to v_j only leads to a tour of smaller cost. □

2 An improvement: A factor 3/2-approximation

To show this, we describe another lower bound on OPT. Let $U \subseteq V$ be any subset of vertices such that $|U|$ is even. Now consider the optimal tour C^* and do the following shortcutting operation. Start the tour C^* from a vertex in U, and directly go to the next U-vertex in C^*, and so on, till you get back to the start vertex. Note that, as in the proof of Theorem 1.1, we get a smaller tour,C^*_U, of the vertices of U. Since U is of even size, the tour C^*_U can be written as $M_1 \cup M_2$ where both M_1 and M_2 are perfect matchings of U. Let M^* be the minimum cost perfect matching of the vertices of U. Therefore, we get

$$OPT = d(C^*) \geq d(C^*_U) = d(M_1) + d(M_2) \geq 2d(M^*)$$

Now, we can state the algorithm.

1. Let T be the minimum spanning tree of (G, d).

2. Let U be the set of odd degree vertices of T. Note that this must be even, since the set of odd degree vertices in *any* graph is of even size.

3. Let M^* be the minimum cost perfect matching of U in G, d.

4. Obtain graph $H(V, F)$ by taking the set of edges in T and the set of edges in M^*. Observe that in H, each vertex has even degree.

5. Find an Eulerian tour in $H(V, F)$. Let the tour be $v_1, v_2, \ldots, v_k, v_1$.

6. From the Eulerian tour obtain a traveling salesman tour by *shortcutting*. That is, the tour is $v_1, v_2, \ldots, v_n, v_1$ where duplicate entries of all vertices (except the last entry of v_1) in the Eulerian tour have been removed.

Theorem 2.1. *Let C be the tour returned by the above algorithm. Then, we get, $d(C) \leq 3/2OPT$.*

Proof. Since $d(C) \leq d(F)$, it suffices to show $d(F) \leq 3/2OPT$. But $d(F) = d(T)+d(M^*) \leq OPT + OPT/2$. □