Approximation Algorithms: Traveling Salesman Problem

In this recitation, we will be studying the ***Traveling Salesman Problem (TSP)***: Given an undirected graph $G(V, E)$ with non-negative integer cost $c(u, v)$ for each edge $(u, v) \in E$, find the Hamiltonian cycle with minimum cost.

1 Metric TSP

TSP is an NP-complete problem, and therefore there is no known efficient solution. In fact, for the general TSP problem, there is no good approximation algorithm unless $P = NP$. There is, however, a known 2-approximation for the ***metric TSP***. In metric TSP, the cost function satisfies the triangular inequality:

$$c(u, w) \leq c(u, v) + c(v, w) \forall u, v, w \in V.$$

This also implies that any shortest paths satisfy the triangular inequality as well: $d(u, w) \leq d(u, v) + d(v, w)$. The metric TSP is still an NP-complete problem, even with this constraint.

2 MST Approximation Algorithm

When you remove an edge from a Hamiltonian cycle, you get a spanning tree. We know how to find minimum spanning trees efficiently. Using this idea, we create an approximation algorithm for minimum weight Hamiltonian cycle.

The algorithm is as follows: Find the minimum spanning tree T of G rooted at some node r. Let H be the list of vertices visited in pre-order tree walk of T starting at r. Return the cycle that visits the vertices in the order of H.

2.1 Approximation Ratio

We will now show that the MST-based approximation is a 2-approximation for the metric TSP problem. Let H^* be the optimal Hamiltonian cycle of graph G, and let $c(R)$ be the total weight of all edges in R. Furthermore, let $c(S)$ for a list of vertices S be the total weight of the edges needed to visit all vertices in S in the order they appear in S.

Lemma 1 $c(T)$ *is a lower bound of* $c(H^*)$.

Proof. Removing any edge from H^* results in a spanning tree. Thus the weight of MST must be smaller than that of H^*.

Lemma 2 $c(S') \leq c(S)$ *for all* $S' \subset S$.

Proof. Consider $S' = S - \{v\}$. WLOG, assume that vertex v was removed from a subsequence u, v, w of S. Then in S', we have $u \to w$ rather than $u \to v \to w$. By triangular inequality, we know that $c(u, w) \leq c(u, v) + c(v, w)$. Therefore $c(S)$ is non-increasing, and $c(S') \leq c(S)$ for all $S' \subset S$.

Consider the walk W performed by traversing the tree in pre-order. This walk traverses each edge exactly twice, meaning $c(W) = 2c(T)$. We also know that removing duplicates from W results in H. By Lemma 1, we know that $c(T) \leq c(H^*)$. By Lemma 2, we know that $c(H) \leq c(W)$. Putting it all together, we have $c(H) \leq c(W) = 2c(T) \leq 2c(H^*)$.

3 Christofides Algorithm

We can improve on the MST algorithm by slightly modifying the MST. Define an ***Euler tour*** of a graph to be a tour that visits every edge in the graph exactly once.

As before, find the minimum spanning tree T of G rooted at some node r. Compute the minimum cost perfect matching M of all the odd degree vertices, and add M to T to create T'. Let H be the list of vertices of Euler tour of T' with duplicate vertices removed. Return the cycle that visits vertices in the order of H.

3.1 Approximation Ratio

We will show that the Christofies algorithm is a $\frac{3}{2}$-approximation algorithm for the metric TSP problem. We first note that an Euler tour of $T' = T \cup M$ exists because all vertices are of even degree. We now bound the cost of the matching M.

Lemma 3 $c(M) \leq \frac{1}{2}c(H^*)$.

Proof. Consider the optimal solution H' to the TSP of just the odd degree vertices of T. We can break H' to two perfect matchings M_1 and M_2 by taking every other edge. Because M is the minimum cost perfect matching, we know that $c(M) \leq \min(c(M_1), c(M_2))$. Furthermore, because H' only visits a subset of the graph, $c(H') \leq c(H^*)$. Therefore, $2c(M) \leq c(H') \leq c(H^*) \Rightarrow c(M) \leq \frac{1}{2}c(H^*)$.

The cost of Euler tour of T' is $c(T) + c(M)$ since it visits all edges exactly once. We know that $c(T) \leq c(H^*)$ as before (Lemma 1). Using Lemma 3 along with Lemma 1, we get $c(T) + c(M) \leq c(H^*) + \frac{1}{2}c(H^*) = \frac{3}{2}c(H^*)$. Finally, removing duplicates further reduces the cost by triangular inequality. Therefore, $c(H) \leq c(T') = c(T) + c(M) \leq \frac{3}{2}c(H^*)$.

6.046J / 18.410J Design and Analysis of Algorithms
Spring 2015