CMPUT 675: Approximation Algorithms	**Winter 2005**

Lecture 5: Jan 26

Lecturer: Mohammad R. Salavatipour — *Scribe: Zhipeng Cai*

5.1 Steiner Tree and TSP

Today we will study the Steiner Tree and Traveling Salesman problems in the metric case.

Definition 5.1 *Give a graph $G(V, E)$ with nonnegative edges whose vertices are partitioned into two sets,* terminal $T \subseteq V$*) and* Steiner *nodes $V - T$. We are also given a cost function $c : E \rightarrow Q^+$, and the goal is to find a minimum cost Steiner tree in G, where a Steiner tree is a tree which all the terminals (Steiner nodes are optional).*

One of the special cases of Steiner tree is Spanning tree in which $T = V$. As we all know, minimum Spanning tree problem can be solved in polynomial time. But the Minimum Steiner problem is NP-hard; in fact it is APX-hard even if costs are all in $\{1, 2\}$.

We will consider the restriction of the problem to the metric case. The metric Case is when the edge costs satisfy triangle inequality, i.e. $\forall u, v, w \in V : c(uv) \leq c(uw) + c(wv)$. We show that this restricted version of the problem is in fact as hard as the general version:

Theorem 5.2 *There is an approximation factor preserving reduction from the Steiner tree problem to the metric Steiner tree problem.*

Proof: Given $G(V, E)$ as an instance for the general case, construct the complete graph G' on V by assigning $c_{G'}(uv)$ (cost of uv in G') to be the cost of the shortest uv-path in G. The set of terminals in G' is the same as in G.

Trivially this is a metric instance (follows directly from the definition and property of shortest paths).

Claim 5.3 *The cost of optimal solution to G' is less than or equal to the cost of optimal solution of G.*

Proof: This is because for any edge uv in G, $c(uv)_{G'} \leq c(uv)_G$. ■

Claim 5.4 *The cost of optimal solution to G is less than or equal to the cost of optimal solution to G'.*

Proof: Take any optimal Steiner tree T' for G' and replace each edge vw in T' by the shortest path between v and w in G to obtain a subgraph of G. Remove the extra edges to obtain that create cycles to obtain a tree$T$. Clearly, the cost does not increase during this process. Therefore: $cost(T) \leq cost(T')$. ■

The proof of theorem follows from the above two claims. ■

By Theorem 5.2, it is enough to consider only metric instances of the Steiner tree problem. Now we present a simple 2-approximation for Steiner tree.

Theorem 5.5 *The cost of a MST on T is at most twice the optimum Steiner tree.*

Proof: Consider an optimal Steiner tree F for G. Double every edge and obtain an Eulerian tour τ from a DFS traversal of F. The cost of this tour τ is exactly $cost(\tau) = 2 * cost(F)$. Now shortcut the tour τ by visiting the next unvisited terminal directly. During the process of shortcutting the tour τ, we do not increase the cost because we are in the metric case. Therefore, at the end of this step, we obtain a cycle which contain only the terminals and whose cost is at most the cost of the original tour, which is at most $2 * cost(F)$. Now remove the heaviest edge of this cycle to obtain a path P on the terminals. We obtain that $cost(P) \leq (2 - \frac{2}{|T|}) \cdot cost(F) = (2 - \frac{2}{|T|}) \cdot OPT$, where $|T|$ is the number of terminals (which is also the number of vertices of P). Since P is a (special) spanning tree for terminals (it is a path), the cost of the MST on T is at most $2 - \frac{2}{|T|}$ of the OPT. ■

Currently the best approximation ratio is $1 + \frac{\ln 3}{2} \approx 1.55$. The starting point of these improved algorithms is also a MST. The difference is in the shortcutting procedure.

5.2 Traveling Salesman Problem

This is a very well-known NP-hard problem. There are at least three books written on this problem.

Definition 5.6 Traveling Salesman Problem (TSP): *Given a complete graph $G(V, E)$ on n vertices with edge cost $c : E \longrightarrow Q^+$, find a minimum cost cycle visiting every vertex exactly once, i.e. a minimum cost Hamiltonian cycle.*

Finding a Hamiltonian cycle in a graph is NP-hard. Using this fact, we show that TSP cannot have an approximation algorithm in the general case.

Theorem 5.7 *For any polynomially computable function $f(\cdot)$, TSP does not have an $f(n)$-approximation algorithm unless P=NP.*

Proof: Let G be the instance of Hamiltonian cycle problem and construct G' on the same vertex set in the following way:

- If $e \in G$, then the cost of e in G' is 1.
- If $e \notin G$, the cost of e in G' is $> f(n) \cdot n$, where n is the number of vertices in G.

If G has a Hamiltonian cycle then the TSP tour in G' has cost n. If G does not have a Hamiltonian cycle then every TSP tour in G' must use at least one of those heavy edges and therefore has cost larger than $f(n) \cdot n$. Thus, if we have an algorithm A for TSP with factor $f(n)$, we can decide whether G has a Hamiltonian cycle, which is NP-hard. ■

So let's focus on the metric instances of TSP. This includes the Euclidean metric, for which there is a PTAS.

5.2.1 Metric TSP

The first algorithm we present is a 2-approximation algorithm for metric TSP.

Algorithm Metric TSP - Factor 2
1.Find a MST T on G
2.Double every edge to obtain an Eulerian tour τ.
3.Do the short-cutting procedure to obtain a Hamiltonian cycle c from τ.

Figure 5.1: 2-Approximation algorithm for Metric TSP

From every TSP tour we can obtain a spanning tree of G by removing one edge. Therefore, the cost of a TSP tour is at least as large as the cost of MST of G. This implies that for the Eulerian tour τ obtained in Step 2: $cost(\tau) = 2 * cost(T) \leq 2OPT$. Since the shortcutting procedure does not increase the cost of the tour, at the end we have a TSP tour with cost at most $2OPT$.

We can improve this algorithm to obtain a 1.5-approximation algorithm.

Algorithm Metric TSP - Factor 1.5
1.Find a MST T on G.
2.Find a minimum cost perfect matching M on the graph which includes odd degree vertices.
3.Add M to T, we obtain a graph which is Eulerian.
4.Obtain an Eulerian tour tour τ and do the shortcutting procedure as in the previous algorithm.

Figure 5.2: 1.5-Approximation algorithm for Metric TSP

After step 3 all the vertices have even degrees and the rest of the algorithm is the same as the 2-approximation algorithm. Therefore, we only need to show that $cost(M) \leq OPT/2$, since the cost of Euler tour τ is exactly $cost(T) + cost(M)$. The following lemma completes the proof of this algorithm.

Lemma 5.8 *Let $V' \subseteq V$ s.t $|V'|$ is even and let M be a minimum cost perfect matching on V'. Then the $cost(M) \leq OPT/2$.*

Proof: Consider any optimal TSP tour τ of G and let τ' be the tour obtained from τ by shortcutting on the vertices of $V - V'$, i.e. skip the vertices of $V - V'$. So τ' is a tour on V' only and $cost(\tau') \leq cost(\tau)$ because we have a metric instance. Now, since $|V'|$ is even, τ' can be decomposed to two perfect matchings by choosing the even edges or the odd edges on the tour. Since the cost of a minimum perfect matching on V' is smaller that each of these: $cost(M) \leq \frac{1}{2}cost(\tau') \leq OPT/2$. ■

From the lemma, we can obtain the guarantee ratio for the algorithm to be $\frac{3}{2}$.

This algorithm, called Christofides, is the best known approximation algorithm for TSP for the past 30 years.

Major open problem: Obtain a better approximation algorithm for metric TSP or prove that there is no such algorithm, under some reasonable complexity assumption.

In fact, this algorithm proves that 3/2 is an upper bound on the integrality gap of a well-known IP/LP formulation of TSP, called the subtour elimination LP-relaxation, given below. For every set $S \subseteq V$, let $\delta(S)$ denote the set of edges in the cut between S and $V - S$, i.e. $\delta(S) = \{uv \in E | u \in S, v \in V - S\}$.

$$\begin{array}{ll} \text{minimize} & \sum e \in E c_e x_e \\ \text{subject to} & \forall v \in V : \sum e : v \in e x_e = 2 \\ & \sum_{e: e \in \delta(S)} x_e \geq 2 \\ & x_e \in \{0,1\} \end{array}$$

It is not difficult to show that the optimal solutions to this IP are optimal TSP tours. Consider the LP-relaxation of this IP. Christofides algorithm gives a $\frac{3}{2}$-approximation algorithm for this IP. The best known lower bound for the integrality gap of this LP is $4/3$.

Open problem: Improve the lower or upper bound for the integrality gap of this LP.