CMPUT 675: Approximation Algorithms

Lecture 3 (Sep 8): STEINER TREE and TSP

Lecturer: Zachary Friggstad

Scribe: Zachary Friggstad

Fall 2014

Most of this material is covered in the course notes from the previous lecture. Those notes are borrowed from a different class but we did not cover them all until this lecture. I'm including the details again here so you can see an example of how I would record the notes.

3.1 Steiner Tree Problem

The STEINER TREE and the METRIC STEINER TREE problems were defined in the previous lecture. Equivalence of the two problems was shown in the sense that an α -approximation for one immediately yields an α -approximation for the other. We saw an approximation for the METRIC STEINER TREE problem in this lecture.

Recall that in an instance of the METRIC STEINER TREE problem we are given a set of nodes V with nonnegative costs $c(u, v), u, v \in V$ satisfying the metric properties:

- c(v, v) = 0 for all $v \in V$
- c(u, v) = c(v, u) for all $u, v \in V$ (symmetry)
- $c(u,v) \le c(u,w) + c(w,v)$ for all $u, v, w \in V$ (triangle inequality)

The goal is to find the cheapest subtree F that includes all terminals T and, perhaps, some non-terminals (*Steiner nodes*) in V - T. Some interesting aspects of this problem:

- When T = V this is the MINIMUM SPANNING TREE problem.
- When $T = \{s, t\}$ this is the SHORTEST PATH problem.
- The general problem can be solved in polynomial time when |T| can be regarded as a constant (a simple but illuminating exercise), or even $|T| = O(\log |V|)$ (takes a bit more thought).

Consider the very simple algorithm described in Algorithm 1.

Algorithm 1 METRIC STEINER TREE Approximation

Input: A metric on nodes V with costs $c(u, v), u, v \in V$ and a nonempty subset $T \subseteq V$ of terminals. **Output**: A tree that spans all terminals.

 $F \leftarrow$ minimum spanning tree of the nodes Treturn F

Theorem 1 Algorithm 1 is a $\left(2-\frac{2}{|T|}\right)$ -approximation for the METRIC STEINER TREE problem.

Before proving this, there is one important concept that will be used here and in our discussion of the TRAVELING SALESMAN problem.

Definition 1 A graph G = (V, E) with, perhaps, parallel edges is called Eulerian if it is connected and every vertex v has even degree.

Recall that a tour of a graph is a walk that starts and ends at the same vertex.

Theorem 2 A graph G is Eulerian if and only if it has a tour that uses each edge exactly once. Such a tour can be found in polynomial time, if it exists.

Proof of Theorem 1. The idea is that we will show how to transform an optimum Steiner tree into a spanning tree of T while increasing its cost only by a small amount.

To that end, let F^* be an optimum Steiner tree. Let $2F^*$ be the multiset of edges that contains two copies of each edge in F^* . Let S' denote the set of Steiner nodes that lie on the Steiner tree F^* .

The graph $(T \cup S', 2F^*)$ is Eulerian because it contains a spanning tree F^* (so it is connected) and every vertex has even degree (twice the degree under F^*). Therefore, by Theorem 2 there is some Eulerian tour C of the graph $(T \cup S', 2F^*)$. We will transform this tour into a Hamiltonian cycle H on T with no greater cost using a process known as *shortcutting*.

Iteratively modify C in the following way. If the tour visits some $v \in S'$, say by following the sequence of edges (u, v), (v, w), then delete these edges and replace them with (u, w). Note that we may have to do such shortcutting multiple times for some $v \in S'$. Next, if some $v \in T$ is visited multiple times by the tour then pick any such occurrence of v on the tour. Say the tour visits v by following (u, v) into v and then (v, w) out of v. Replace (u, v) and (v, w) with (u, w). Again, note that we may have to do this shortcutting multiple times for some $v \in T$ until it is visited exactly once by the tour.

Let H be the resulting tour. Now, H does not visit any vertex in S' (they were all removed) and it visits each vertex in T exactly once (since duplicates were removed). Furthermore, every we replaced edges (u, v), (v, w) with a single edge (u, w) we have $c(u, v) \leq c(u, w) + c(w, v)$ so the cost of the tour never increases. Therefore, $c(H) \leq c(\mathcal{C})$.

Finally, delete the most expensive edge on H to get a path \overline{F} . Note that \overline{F} is a spanning tree of T. Since H has exactly |T| edges, then the cost of the deleted edge is at least c(H)/|T|.

We conclude by bounding the cost of F, the solution found by Algorithm 1.

 $\begin{array}{lll} c(F) &\leq & c(\overline{F}) & (F \text{ is the cheapest spanning tree of } T) \\ &\leq & \left(1 - \frac{1}{|T|}\right) \cdot c(H) & (\text{previous paragraph}) \\ &\leq & \left(1 - \frac{1}{|T|}\right) \cdot c(2F^*) & (H \text{ is obtained by shortcutting an Eulerian tour of } 2F^*) \\ &= & \left(1 - \frac{1}{|T|}\right) \cdot 2 \cdot c(F^*) \end{array}$

This is what we needed to show.

This analysis cannot be improved. Consider a metric with $V = \{v_1, \ldots, v_n\}$ and terminals $T = \{v_1, \ldots, v_{n-1}\}$. The costs are given by $c(v_i, v_j) = 2$ for all $1 \le i, j \le n-1$ and $c(v_i, v_n) = 1$ for all $1 \le i \le n-1$. It is easy to see that this is a metric.

The cheapest Steiner tree F^* consists of all cost 1 edges $\{(v_1, v_n), (v_2, v_n), \dots, (v_{n-1}, v_n)\}$ so c(F) = n - 1. Any spanning tree of T uses n - 2 edges and each edge has cost 2, so c(F) = 2(n - 2). Note that $c(F) = \left(2 - \frac{2}{|T|}\right) \cdot c(F^*)$.

3.2 The Traveling Salesman Problem

We only briefly discussed this problem in class.

Definition 2 In the TRAVELING SALESMAN problem (TSP), we are given a metric with nodes V and distances $c(u, v), u, v \in V$. The goal is to find the cheapest Hamiltonian cycle.

Algorithm 2 describes a very simple 2-approximation. The shortcutting step is done exactly the same way as in the proof of Theorem 1.

 Algorithm 2 TRAVELING SALESMAN Approximation

 Input: A metric on nodes V with costs $c(u, v), u, v \in V$.

 Output: A Hamiltonian cycle.

 $T \leftarrow$ a minimum spanning tree of the metric shortcut an Eulerian circuit of the graph (V, 2T) to get a Hamiltonian cycle H return H

Theorem 3 Algorithm 2 is a $(2-\frac{2}{n})$ -approximation.

Proof. Let H^* denote an optimum Hamiltonian cycle. Deleting the heaviest edge of H^* leaves a spanning tree F with $c(F) \leq \left(1 - \frac{1}{n}\right) \cdot c(H^*)$.

Let H be the Hamiltonian cycle returned by Algorithm 2. We bound the cost of H as follows.

 $\begin{array}{rcl} c(H) & \leq & c(2T) & (H \text{ is obtained by shortcutting an Eulerian tour of } 2T) \\ & = & 2 \cdot c(T) \\ & \leq & 2 \cdot c(F) & (T \text{ is the cheapest spanning tree}) \\ & \leq & 2 \cdot \left(1 - \frac{1}{n}\right) \cdot c(H^*) & (\text{above discussion}) \end{array}$

We will see an improved $\frac{3}{2}$ -approximation next lecture.