COMPSCI 590.1: Graph Algorithms March 25, 2015

Lecture 21

Lecturer: Debmalya Panigrahi *Scribe: Rupert Freeman*

1 Overview

In the last lecture we looked at the online Set Cover problem which we solved by LP rounding. We then extended this to an algorithm for the online Facility Location problem (which we observed was equivalent to Set Cover), obtaining a competitive ratio of $\log^2(\max\{m,n\})$. In this lecture we extend this to an approximation algorithm for the online node-weighted Steiner tree problem.

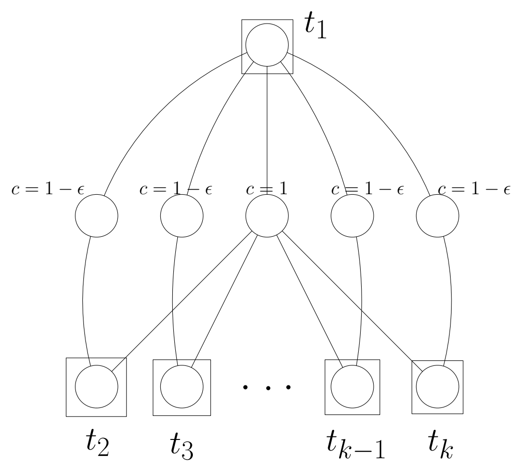
2 Online Node-Weighted Steiner Tree

We are given a graph $G = (V,E)$, weight function $w : V \to \mathbb{R}_+$ and set of terminals $T = \{t_1,t_2,\ldots,t_k\}$ that arrive one at a time. We wish to select a minimum cost subgraph $H \subseteq G$ such that all terminal are connected.

2.1 Failure of Greedy Algorithm

We first observe that a greedy approach can fail badly for this problem.

Example 1. Here the greedy algorithm at step i chooses the direct path from t_1 to t_i, for at total cost of



$k(1-\varepsilon)$. The optimal solution is to buy only the central vertex at cost 1. We have assumed without loss of generality that the terminals have no cost, since they are bought even in the optimal solution.

2.2 Reduction to Facility Location

We will use the following lemma to reduce node-weighted Steiner tree to the Facility Location problem.

Lemma 1. *For any Steiner tree T and any sequence of terminals $t_1,t_2,\ldots,t_k$, there exist paths $p_2,p_3,\ldots,p_k$ and vertices $v_2,v_3,\ldots,v_k$ (with v_i lying on path p_i) such that*

(1) $\sum_{i=2}^{k}(cost(p_i) - cost(v_i)) = O(\log n) cost(T)$

(2) $\sum_{v: v=v_i \text{ for some } i} cost(v) \leq cost(T)$

(3) p_i *is from* t_i *to some terminal* $t_j,\ j < i$

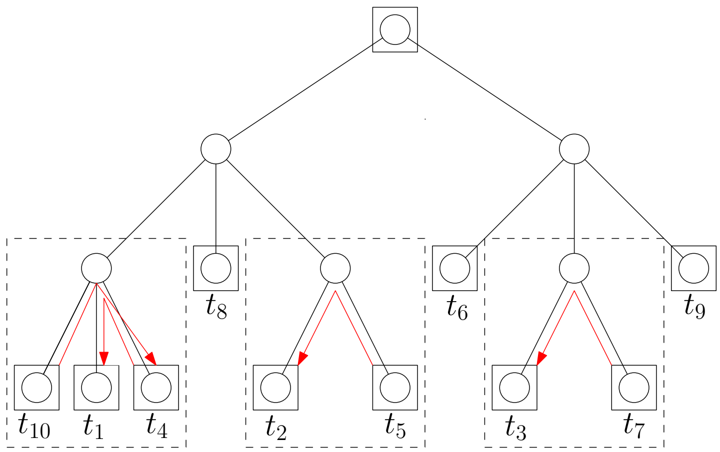


Figure 1: Steiner tree used to illustrate the proof of Lemma 1, with lowest level spiders in dashed boxes and paths p_i illustrated with red arrows.

Proof. Consider a Steiner tree. Without loss of generality, we can assume that a node is a terminal if and only if it is a leaf of the tree. Consider the spider decomposition of T, first looking at only the lowest level spiders (contained in dotted line boxes in Figure 2.2). For each terminal t_i, declare p_i to be the path shortest path to the previous terminal within each spider (if it exists). For each p_i, let v_i be the head of the spider on path p_i.

Now, remove each terminal for which p_i has been defined, along with the associated spider leg. The terminal with lowest index in each spider will remain. Now recurse on higher level spiders until each p_i and v_i ($i \geq 2$) has been defined.

Observe that (3) is automatically satisfied by the definition of the paths. (2) is satisfied because $v_i \in T$ for all i. Property (1) is satisfied because every vertex not removed appears on at most two paths at each level of recursion, and there are $\log n$ levels of recursion. Therefore then total cost over all paths (with distinguished vertices removed) is $2 \log n \text{cost}(T)$. □

Given an instance of node-weighted Steiner tree, we construct an instance of Facility Location as follows: Each terminal is a client, and each node is a facility with the cost of the facility equal to the cost of the node. The cost c_{ij} of an edge from a terminal t_i to a node v_j is the length of the shortest path from t_i to some previous terminal with the cost of v_j set to zero. The setup is shown in Figure 2.2.

Note that any solution to the constructed Facility Location problem corresponds to a solution to the Steiner tree problem, since by definition all terminals are connected to a previous terminal. And by Lemma 1 we know that the total cost of the edges in the Facility Location solution is $O(\log n)\text{cost}(T)$ (moreover, this is the bound on the cost of the full solution since the nodes chosen as "facilities" have total cost at most $\text{cost}(T)$), where T is the minimal Steiner tree.

From the previous lecture, we know that we can achieve a competitive ratio of $\log^2 k$ for online Facility Location. Therefore the overall competitive ratio for online node-weighted Steiner tree is $\log^2 k \log n$.

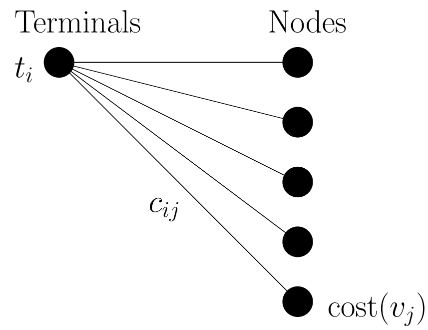


Figure 2: The consturction of an instance of Facility Location from an instance of node-weighted Steiner tree.

3 Summary

We have given a summary of an algorithm for the online node-weighted Steiner tree problem with competitive ratio $\log^2 k \log n$ as given in [NPS11].

References

[NPS11] Joseph Naor, Debmalya Panigrahi, and Mohit Singh. Online node-weighted steiner tree and related problems. In *Foundations of Computer Science*, pages 210–219, 2011.