COMPSCI 638: Graph Algorithms	October 18, 2019

Lecture 16

Lecturer: Debmalya Panigrahi — *Scribe: Kevin Sun*

1 Overview

In the last lecture, we analyzed the greedy algorithm for the online Steiner tree problem. In this lecture, we will use a similar approach to analyze the greedy algorithm for the online Steiner forest problem, and also see an algorithm with provably optimal competitive ratio.

2 Online Steiner Forest

We are given a graph $G = (V, E)$ where each edge e has cost $c(e) \geq 0$. At each time i, the algorithm is given one of k terminal pairs $(s_i, t_i) \in V \times V$ and must connect them by adding a subset of edges to the solution $F \subseteq E$ (initially empty). The goal is to minimize the total cost of the solution, and the performance of an algorithm is measured by its competitive ratio (see Lecture 15).

2.1 The Greedy Algorithm

Notice that at time i, edges already in our solution F have already been "purchased," so we can think of their modified cost as zero. The greedy algorithm, at each time i, connects s_i to t_i by adding the edges of the shortest s_i-t_i path according to the modified costs. This is equivalent to contracting the entire s_i-t_i path to a single vertex at the end of each time step.

Theorem 1. *The greedy algorithm for Steiner forest is $O(\log^2 k)$-competitive.*

Remark: Theorem 1 was originally proven by Azerbach, Azar, and Bartal [AAB04], though our proof today will be slightly different. Furthermore, there is no known lower bound showing that the greedy algorithm is $\omega(\log k)$-competitive, so the analysis of the greedy algorithm is not known to be tight. However, we will analyze an algorithm in Section 2.2 that is $O(\log k)$-competitive. This is optimal, since Imase and Waxman [IW91] showed that any algorithm for online Steiner tree is $\Omega(\log k)$-competitive.

Overview of proof: We first recall the analysis of the greedy algorithm for the Steiner tree problem (see Lecture 15). At each step i, we incurred cost $C_i \in (2^\ell, 2^{\ell+1}]$ for some ℓ, and in the ℓ-th dual, we placed a ball around t_i with radius $2^{\ell-1}$. This resulted in a set of dual solutions, and in each one, every ball had the same radius. We then argued that each dual solution is feasible, so their average is feasible, and the $\log k$ factor resulted from the number of dual solutions we constructed.

Our proof of Theorem 1 will proceed similarly. However, due to the nature of terminal *pairs* (rather than vertices), it may not be possible to place a feasible dual ball at every time step. In these cases, we will add an "accounting" edge in the corresponding dual. Our final (primal) cost will be covered by the usual dual balls, as well as these accounting edges.

To bound the number of accounting edges, we will use the following result without proof; it is sometimes known as the Erdős Girth Theorem. Recall that the *girth* of a graph is the length of the shortest cycle in the graph.

Theorem 2. *Let $G = (V, E)$ be a graph on n vertices with girth g. Then $|E| \approx n^{1+O(1/g)}$; in particular, if $g = \Omega(\log n)$, then $|E| = O(n)$.*

Proof of Theorem 1. We will construct a set of dual solutions, as discussed above. At each step i, let C_i denote the incurred (modified) cost, and suppose $C_i \in (2^\ell, 2^{\ell+1}]$ for some integer ℓ. We first try to place a ball in the ℓ-th dual with radius $2^\ell / \log k$ centered at either s_i or t_i. (If both centers are feasible, then we pick one arbitrarily.) If neither center is feasible, then there must exist terminals x, y whose balls overlap with the balls around s_i and t_i, respectively. In this case, we place an "accounting edge" $\{x, y\}$ in the graph corresponding to the ℓ-th dual.

Let $G_\ell = (V_\ell, E_\ell)$ denote the graph containing the accounting edges in the ℓ-th dual, and let $D(\ell)$ denote the cost of this dual solution. Since every vertex of V_ℓ is the center of a ball with radius $2^\ell / \log k$, we have

$$\frac{2^\ell}{\log k}|V_\ell| \leq D(\ell). \tag{1}$$

Note that the above expression is an inequality because V_a does not include vertices that are not incident to any accounting edges. Furthermore, at each time i, we either place a dual ball with cost within a constant factor of $C_i / \log k$ or we add an edge to E_ℓ that accounts for the cost of the missing dual ball. Thus, if $T(\ell)$ denotes the time steps associated with the ℓ-th dual, we have

$$\sum_{i \in T(\ell)} C_i \leq O(\log k) \cdot \left(D(\ell) + |E_\ell| \frac{2^\ell}{\log k} \right). \tag{2}$$

We now state a lemma that allows us to invoke Theorem 2 to conclude the proof; we defer the proof of this lemma to the end of this section.

Lemma 3. *The girth of the accounting graph G_ℓ is $\Omega(\log k)$.*

Together, Lemma 3 and Theorem 2 imply $|E_a| = O(|V_a|)$. Combining this with (1) and (2), we can conclude that the total cost of the algorithm is at most

$$\sum_{i=1}^{k} C_i \leq O(\log k) \cdot \sum_\ell D(\ell) = O(\log^2 k) \cdot OPT,$$

where the final inequality follows from the fact that there are essentially $O(\log k)$ different dual solutions (see Lecture 15 for more details), and each provides a lower bound on OPT. □

Proof of Lemma 3. Let C be a cycle in G_ℓ containing $|C|$ edges. Let $e = \{x, y\}$ denote the last edge placed in C; suppose this occurred at time i. Then at time i, the algorithm sought the shortest s_i-t_i distance using modified edge costs; we will bound this cost $C_i \in (2^\ell, 2^{\ell+1}]$ against $|C|$.

Consider any edge $e' = \{x', y'\} \neq e$ of the cycle C. This edge, being an accounting edge, exists because we could not place a ball around s_j or t_j for some $j < i$. Without loss of generality, assume that the ball around s_j would intersect with the ball around x', and the same for t_j and y' (see Fig. 1). Then one x'-y' path follows the sequence (x', s_j, t_j, y'). At time i, the modified cost of traversing s_j to t_j is zero, so this is a x'-y' path with modified cost at most $4 \cdot 2^\ell / \log k$.

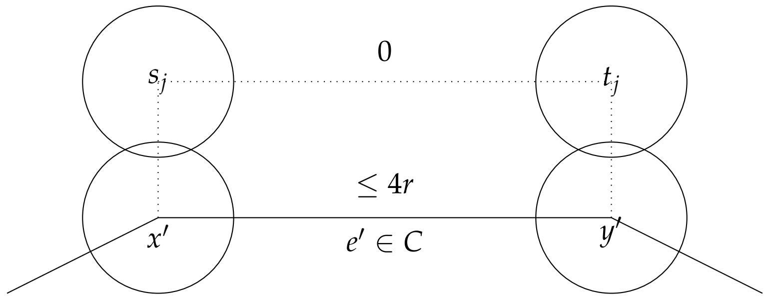

Figure 1: A diagram illustrating the proof of Lemma 3. Each of the four balls has radius $r = 2^\ell / \log k$. Since e' was placed before e, at time $i > j$, the modified cost between s_j and t_j was 0. Thus, the modified cost between x' and y' is at most $4r$.

Applying this logic to every edge of C, we see that at time i, there exists a path from s_i to t_i whose modified distance is at most $(|C|-1)2^{\ell+2}/\log k$. Adding in modified cost from s_i to x and from y to t_i, we can conclude

$$2^\ell < C_i \leq \frac{2^{\ell+2}}{\log k}(|C|-1) + \frac{2^\ell + 2}{\log k} = \frac{2^{\ell+2}}{\log k}|C|.$$

Rearranging the above yields $|C| \geq \log k/4 = \Omega(\log k)$. □

2.2 Augmented Greedy

We now present a slightly modified algorithm that we call the augmented greedy algorithm. At time i, if the algorithm incurs cost $C_i \in (2^\ell, 2^{\ell+1}]$, then we try to place a ball in the ℓ-th dual with radius $2^{\ell-3}$ (rather than $2^\ell / \log k$). If this is not feasible at neither s_i nor t_i, then there must exist terminals x, y such that (WLOG) the balls around x and s_i overlap, and the same for y and t_i. In this case, we add the edges of shortest (modified) x-s_i path and y-t_i path.

Theorem 4. *The augmented greedy algorithm is $O(\log k)$-competitive.*

Proof. The proof follows the same structure as the proof of Theorem 1. We first notice that at each step i, we incur cost $C_i \in (2^\ell, 2^{\ell+1}]$, as well as (potentially) the modified cost of a x-s_i path and y-t_i path, where x, y are described above. However, since x and s_i have overlapping balls (and the same for y and t_i), this additional cost is at most $4 \cdot 2^{\ell-3} = 2^{\ell-1} \leq C_i$. Thus, the overall cost of the algorithm is at most double the cost of the standard greedy algorithm.

Let $G_\ell = (V_\ell, E_\ell)$ denote the graph containing the accounting edges in the ℓ-th dual, and let $D(\ell)$ denote the cost of this dual solution. As we saw in the proof of Theorem 1, we have

$$2^{\ell-3}|V_\ell| \leq D(\ell)$$

and the total cost incurred at steps associated with the ℓ-th dual is

$$\sum_{i \in T(\ell)} C_i \leq O(1) \cdot \left(D(\ell) + |E_\ell| 2^{\ell-3} \right).$$

Again, since there are $O(\log k)$ duals, it suffices to prove $|E_\ell| = O(|V_\ell|)$. To do this, we will prove a stronger claim, namely, the accounting graph is acyclic.

For contradiction, let C be a cycle in G_a and consider the last cycle $e = \{x, y\}$ added to C; suppose e was added at time i. Now let $e' = \{x', y'\} \neq e$ be any edge in C; suppose e' was added at time $j < i$. Notice that at time i, there exists a x'-y' path with modified cost zero, by following the sequence (x', s_j, t_j, y') (assuming the balls around x' and s_j overlap). (We can see this in Fig. 1 with $r = 0$.) Thus, we arrive at a contradiction:

$$2^\ell < C_i \leq 2 \cdot 2^{\ell-3} + 0 \cdot (|C| - 1) + 2 \cdot 2^{\ell-3} = 2^{\ell-1}. \qquad \square$$

3 Summary

In this lecture, we proved that the greedy algorithm for the online Steiner forest problem is $O(\log^2 k)$-competitive, and a very similar analysis showed that the augmented greedy algorithm is $O(\log k)$-competitive. Both proofs use the technique of dual fitting introduced in Lecture 15.

References

[AAB04] Baruch Awerbuch, Yossi Azar, and Yair Bartal. On-line generalized steiner problem. *Theoretical Computer Science*, 324(2-3):313–324, 2004.

[IW91] Makoto Imase and Bernard M Waxman. Dynamic steiner tree problem. *SIAM Journal on Discrete Mathematics*, 4(3):369–384, 1991.