CS602 Applied Algorithms **2019-20 Sem II**

Lecture 15: March 05

Scribe: Dibyangshu M *Lecturer: Rohit Gurjar*

Steiner Tree Problem: Given a graph $G = (V, E)$ with edge costs and a set of terminals $T \subseteq V$, find the minimum cost subgraph $G' = (V, F)$ where $F \subseteq E$, such that T is connected in G'.

Steiner Forest Problem: Given a graph $G = (V, E)$ and sets of terminals $T_i \subseteq V$, $1 \leq i \leq k$, find the minimum cost subgraph $G' = (V, F)$ where $F \subseteq E$, such that for $1 \leq i \leq k$, T_i is connected in G'.

In the last lecture we looked at the Euclidean Steiner Tree problem where the cost of every edge is the Euclidean distance between its endpoints.

Algorithm A for the Steiner Tree problem:

Input : Graph G and a set of terminals T,

Build a complete graph $H = (T, F)$ such that $\forall$ $(t_1, t_2) \in F$, $w(t_1, t_2) = d_G(t_1, t_2)$.
$w(t_1, t_2)$ = weight of the edge (t_1, t_2),
$d_G(t_1, t_2)$ = distance between t_1 and t_2 in G.

Compute a minimum spanning tree in H.

For each edge in the tree, take the corresponding path in G and output the union of all such paths.

Output: A Steiner tree.

Claim 15.1 (Homework)**.** *The above algorithm is 2-approximation algorithm for graphical and Euclidean Steiner tree.*

Claim 15.2. *Let c_{opt} be the cost of the optimal solution for the Euclidean Steiner tree problem, and let c_{alg} be the cost of the solution given by the above algorithm. Then:* $\frac{c_{alg}}{c_{opt}} = \frac{2}{\sqrt{3}}$

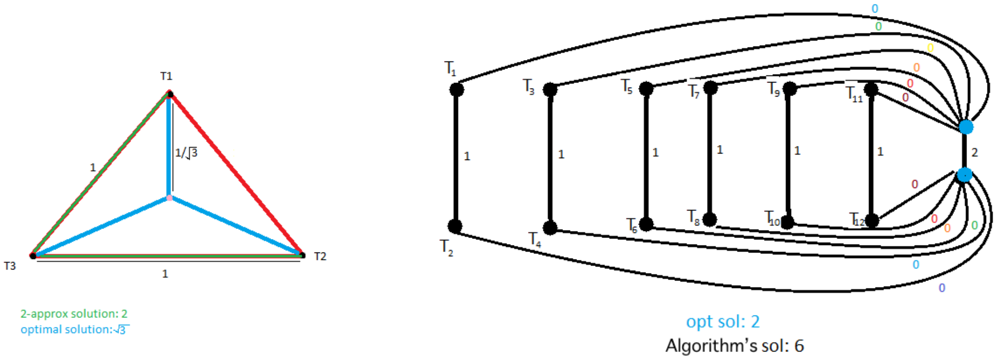
It is interesting to note that despite earlier claims of a proof, Claim 15.2 remains an open conjecture till date. The best widely accepted upper bound for the problem is 1.2134, by Chung & Graham (1985).

While this algorithm gives a good approximation for Steiner tree, it does not do that for the Steiner forest problem (see Figure 1). Now we shall use LP-duality theory to design approximations algorithms that will solve both Steiner tree and Steiner Forest problems, and in fact, can work for more general problems. For this, we will first try to express Steiner Tree/Forest as a hitting-set for a family of sets, which are going to be some interesting cut-sets. And then write an integer program accordingly.

Definition 15.3 (Cut and cut-set)**.** *A cut is a non-empty subset of vertices. For a cut $S \subseteq V$, the corresponding cut-set is the set of edges*

$$\delta(S) = \{(u, v) \in E \mid u \in S,\ v \notin S\}.$$

Definition 15.4 (Interesting Cuts for Steiner Trees)**.** *Let $G = (V, E)$, and $T \subseteq V$ a set of terminals. A cut $S \subseteq V$ is* interesting *(for T) if S splits T, that is, $S \cap T \neq \emptyset$ and $S \cap T \neq T$.*



(a) Euclidean Steiner tree

(b) Not a good approximate solution

Figure 1: The optimal solution and the algorithm A's solution. (a) The first example is that of Euclidean Steiner tree with three nodes, where the optimal solution is a 3-star with a Steiner node in the center. On the other hand the algorithm A's output will be 2 edges of the triangle. The ratio in their lengths is $2/\sqrt{3}$. (b) The second example is that of Graphical Steiner forest. Our sets of terminals are $\{T_1, T_2\}, \{T_3, T_4\}, \ldots, \{T_{11}, T_{12}\}$. Using the shortest path lengths as weights and then computing the Steiner forest on the graph with only terminals as vertices will give us cost 6. While the optimal Steiner forest uses the zero cost edges and the edge with cost 2. The ratio in the costs is 3, and can be made arbitrary high.

The motivation for this definition is the observation that every Steiner tree $F \subseteq E$ will have $\delta(S) \cap F \neq \emptyset$ for every interesting cut $S \subseteq V$. Similarly, we can define interesting cuts for Steiner forests.

Definition 15.5 (Interesting Cuts for Steiner Forests)**.** *Let $G = (V, E)$ and $T_i \subseteqq V$ for $1 \le i \le k$ be the sets of terminals. A cut $S \subseteq V$ is interesting if $\exists\ 1 \le i \le k$ such that: $S \cap T_i \neq \emptyset$ and $\overline{S} \cap T_i \neq \emptyset$.*

Again observe that, for any Steiner forest $F \subseteq E$ that connects each T_i, we have $\delta(S) \cap F \neq \emptyset$ for every interesting cut $S \subseteq V$.

Claim 15.6. *For a graph $G = (V, E)$ and $F \subseteq E$, if every cut-set $\delta(S)$ intersects with F, then $G = (V, F)$ is connected.*

Proof. If a subgraph of a connected graph is disconnected then there must be one cut S for which all the cut-edges in $\delta(S)$ are missing. □

Claim 15.7 (Homework)**.** *For a graph $G = (V, E)$ with a set of terminals T, if $F \subseteq E$ is such that $F \cap \delta(S) \neq \emptyset$ for each interesting cut $S \subseteq V$ (Definition 15.5), then all terminals T are connected in F.*

Using this hitting-set criterion, we can write the following Integer Program for the Steiner Tree Problem.

$$\begin{array}{rcl} \text{min} & \sum_{e \in E} w_e x_e & \text{s.t.} \\ \text{for each } e \in E, & x_e \in \{0, 1\} & \\ \text{for each interesting cut } S \subseteq V, & \sum_{e \in \delta(S)} x_e \ge 1 & \end{array}$$

After relaxation of the integral constraints we get the following LP:

$$\begin{array}{rcl} \text{min} & \sum_{e \in E} w_e x_e & \text{s.t.} \\ \text{for each } e \in E, & x_e \ge 0 & \\ \text{for each interesting cut } S \subseteq V, & \sum_{e \in \delta(S)} x_e \ge 1 & \end{array}$$

Note that the constraint $x_e \leq 1$ would be redundant. Now, the corresponding Dual LP would be:

$$\begin{array}{rll} \max & \displaystyle\sum_{\substack{S\subset V \\ S \text{ is interesting}}} y_S & \text{s.t.} \\ \text{for each interesting } S \subseteq V & y_S \geq 0, & \\ \text{and for each } e \in E, & \displaystyle\sum_{\substack{S\subseteq V \\ S \text{ is interesting} \\ \text{and } e\in\delta(S)}} y_S \leq w_e & \end{array}$$

In the next lecture we look at the Primal-Dual algorithm for the Steiner-Tree Problem.