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

Lecture 17: March 17

Scribe: Vijaykrishna G, Nipun Mittal | *Lecturer: Rohit Gurjar*

Steiner Tree/Forest. A graph G with edge weights and some vertex pairs $(s_1, t_1), (s_1, t_1), ..., (s_k, t_k)$ are given. The goal is to compute a forest that connects s_i to t_i for each i and has minimum possible weight.

Interesting cuts : $S \subseteq V$ s.t. $\exists i$, $s_i \in S$ and $t_i \in \overline{S}$.

- When $k = 1$, Steiner Tree/Forest problem is equivalent to Shortest Path problem.
- When $k = {}^{|V|}C_2$ i.e, every pair of vertices needs to be connected then the problem is equivalent to Minimum Spanning Tree problem.

1 Algorithm Outline for the Steiner Tree Problem

Step 1: Start with a feasible dual solution. The simplest solution one can think of is $\forall S \;\; y_S = 0$.

Step 2: Try to find a primal feasible *integral* solution that satisfies complementary slackness condition i.e,

$$\text{you can set } x_e = 1 \text{ if } \sum_{S: e \in \delta(S)} y_S = w_e \quad \text{(i.e., tight edge)}$$

In other words, check whether the desired kind of Steiner forest can formed using only the tight edges.

- If yes, then we are done and we have found a solution.
- Otherwise, find a set S s.t. $s_i \in S$ and $t_i \in \overline{S}$ for some i (which implies interesting cut) and for each edge $e \in \delta(S)$, we have $x_e = 0$.

For ease of understanding, one can think that $x_e = 0$ if and only if the edge e is non-tight. We will see later that this is not exactly true, a tight edge can possibly have $x_e = 0$.

Step 3: For such a set S, try to increase y_S till some dual constraint(s) i.e, edge(s) becomes tight.

Since after one iteration at least one edge will become tight, we can easily conclude that the algorithm will terminate after finitely many iterations. We elaborate more on the implementation details. As discussed in the last lecture, we will try to increase many dual variables simultaneously.

We will maintain the subgraph G' with edge set $\{e \in E : x_e = 1\}$ and its connected components. At any stage, for any connected component $S \subseteq V$ of G', all the edges in $\delta(S)$ are potentially non-tight. However, we only need *active sets* among the connected components.

Active set. $T \subseteq V$ is an active set if T forms a connected component in G' and for some i, $s_i \in T$ and $t_i \notin T$.

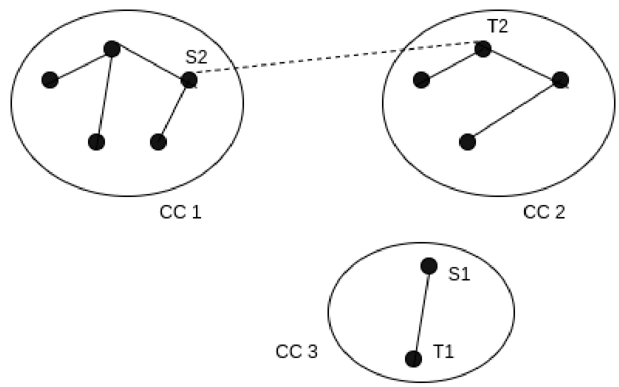

Figure 1: Connected components where CC1 and CC2 are interesting cuts (active sets) while CC3 is not.

We increase the dual variables y_T for all active sets T simultaneously until at least one edge becomes tight. Note that multiple edges can become tight simultaneously, however, we set $x_e = 1$ only for one of them at a time (chosen arbitrarily) and include it in G'. The reason for this is as follows: any new edge that becomes tight connects two different components in G' and we want to maintain that the set of edges $\{e : x_e = 1\}$ forms a forest. Adding multiple edges simultaneously might lead to cycles. The example given below will clarify this.

Pruning: After algorithm stops we'll have subset of edges F as the solution. Now, there may be some spurious edges in the solution, that is, any edge after whose removal the solution remains a valid Steiner forest (for each i, s_i is connected to t_i). It is simple to identify spurious edges. Since F is a forest, there is a unique path between s_i and t_i, for each i. Any edge that does not lie on any of these paths is spurious. Recall the example from the last lecture where $\{e_1, e_2\}$ was the output of the algorithm, but e_2 was spurious and was removed. All spurious edges can removed at once (i.e., setting $x_e = 0$ for that edge).

Now, let us look at an example to understand this.

2 Example from Vazirani's book

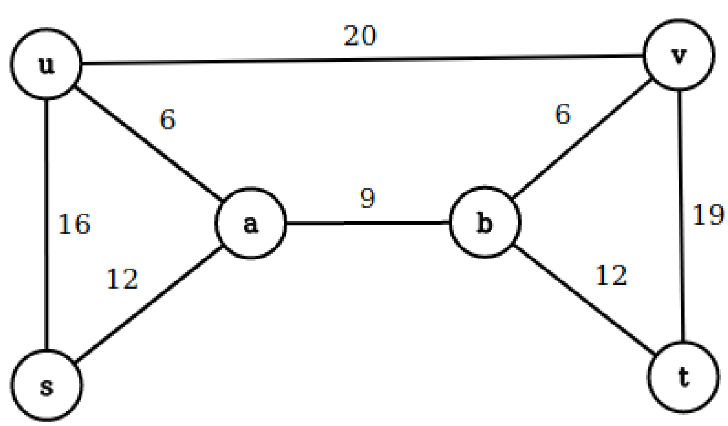

Figure 2: Example. Terminal pairs are (u,v) and (s,t)

Consider the graph given in Figure 2. The terminal pairs are (u, v) and (s, t). That is, we want to compute a forest that connects u to v and s to t. Recall that the dual constraints are given by the following.

$$\text{For each } e \in E, \quad \sum_{\substack{S \subseteq V \\ S \text{ is interesting} \\ \text{and } e \in \delta(S)}} y_S \leq w_e.$$

Initially $y_S = 0$ for all S and there are no tight edges, that is, $x_e = 0$ for each edge e.

Iteration 1: The graph G' has no edges and thus, the connected components are individual vertices. According to the definition of active sets we get the following active dual variables.

Active: $y_u = 0$, $y_s = 0$, $y_v = 0$, $y_t = 0$.

Trying to add ϵ to each of the four variables. The dual constraints look like this (we hide the dual variables that are still zero).

$$\begin{aligned}
2\epsilon = y_u + y_s &\leq w_{(u,s)} = 16. \\
\epsilon = y_s &\leq w_{(s,a)} = 12. \\
\epsilon = y_u &\leq w_{(u,a)} = 6. \\
2\epsilon = y_u + y_v &\leq w_{(u,v)} = 20. \\
0 = 0 &\leq w_{(a,b)} = 9. \\
\epsilon = y_v &\leq w_{(v,b)} = 6. \\
\epsilon = y_t &\leq w_{(b,t)} = 12. \\
2\epsilon = y_v + y_t &\leq w_{(t,v)} = 19.
\end{aligned}$$

The largest value ϵ can take without violating the above constraints is 6. So set

$$y_u = y_s = y_v = y_t = 6.$$

The constraints for the two edges (u, a) and (v, b) become tight (shown in red). We pick one of the two tight edges, say (u, a), and set $x_{(u,a)} = 1$.

Iteration 2: Now, our G' has a single edge (u, a). Connected components are $\{u, a\}$ and all other single vertices.

Active: $y_{\{u,a\}} = 0$, $y_s = 6$, $y_v = 6$, $y_t = 6$.

Trying to add ϵ to each, you get $\epsilon = 0$, because the edge (v, b) is already a tight edge so you cannot increase y_v any further. We count (v, b) as the new tight edge found in this iteration and set $x_{(v,b)} = 1$.

Iteration 3: G' has a edges $(u, a), (v, b)$. Connected components are $\{u, a\}, \{v, b\}, \{s\}, \{t\}$.

Active: $y_s = 6$, $y_t = 6$, $y_{\{u,a\}} = 0$, $y_{\{v,b\}} = 0$. Trying to add ϵ to each of the four variables.

$$y_s = 6 + \epsilon,\ y_t = 6 + \epsilon,\ y_{\{u,a\}} = \epsilon,\ y_{\{v,b\}} = \epsilon.$$

The dual constraints look like this (we hide the dual variables that are still zero).

$$\begin{aligned}
y_u + y_s + y_{u,a} = 6 + 6 + \epsilon + \epsilon &\leq w_{(u,s)} = 16. \\
y_s + y_{u,a} = 6 + \epsilon + \epsilon &\leq w_{(s,a)} = 12. \\
y_u = 6 &\leq w_{(u,a)} = 6. \\
y_u + y_v + y_{\{u,a\}} + y_{\{v,b\}} = 6 + 6 + \epsilon + \epsilon &\leq w_{(u,v)} = 20. \\
y_{\{u,a\}} + y_{\{v,b\}} = \epsilon + \epsilon &\leq w_{(a,b)} = 9. \\
y_v = 6 &\leq w_{(v,b)} = 6. \\
y_t + y_{\{v,b\}} = 6 + \epsilon + \epsilon &\leq w_{(b,t)} = 12. \\
y_v + y_t + y_{\{v,b\}} = 6 + 6 + \epsilon + \epsilon &\leq w_{(t,v)} = 19.
\end{aligned}$$

The largest value ϵ can take without violating the above constraints is 2. That makes the constraint for edge (u, s) tight (shown in red). So we get

$$y_s = y_t = 8,\ y_{u,a} = y_{v,b} = 2,\ \text{and } y_u = y_v = 6 \text{ (remained unchanged)}.$$

Since (u, s) becomes tight, we set $x_{(u,s)} = 1$.

Iteration 4: G' has a edges $(u, a), (v, b), (u, s)$. Connected components are $\{u, s, a\}, \{v, b\}, \{t\}$.
Active: $y_{\{u,s,a\}} = 0$, $y_t = 8$, $y_{\{v,b\}} = 2$. Adding ϵ to each of the three.

$$y_{\{u,s,a\}} = \epsilon,\ y_t = 8 + \epsilon,\ y_{\{v,b\}} = 2 + \epsilon.$$

The dual constraints look like this (we hide the dual variables that are still zero).

$$\begin{aligned}
y_u + y_s + y_{u,a} = 6 + 8 + 2 &\leq w_{(u,s)} = 16.\\
y_s + y_{u,a} = 8 + 2 &\leq w_{(s,a)} = 12.\\
y_u = 6 &\leq w_{(u,a)} = 6.\\
y_u + y_v + y_{\{u,a\}} + y_{\{v,b\}} + y_{\{u,s,a\}} = 6 + 6 + 2 + 2 + \epsilon + \epsilon &\leq w_{(u,v)} = 20.\\
y_{\{u,a\}} + y_{\{v,b\}} + y_{\{u,s,a\}} = 2 + 2 + \epsilon + \epsilon &\leq w_{(a,b)} = 9.\\
y_v = 6 &\leq w_{(v,b)} = 6.\\
y_t + y_{\{v,b\}} = 8 + \epsilon + 2 + \epsilon &\leq w_{(b,t)} = 12.\\
y_v + y_t + y_{\{v,b\}} = 6 + 8 + \epsilon + 2 + \epsilon &\leq w_{(t,v)} = 19.
\end{aligned}$$

The largest value ϵ can take without violating the above constraints is 1. That makes the constraint for the edge (b, t) tight (shown in red). Set $x_{(b,t)} = 1$. Dual variables become

$$y_{\{u,s,a\}} = 1,\ y_t = 9,\ y_{\{v,b\}} = 3 \text{ and } y_s = 8,\ y_{u,a} = 2,\ y_u = y_v = 6 \text{ (remained unchanged)}.$$

Iteration 5: G' has a edges $(u, a), (v, b), (u, s), (b, t)$. Connected components are $\{u, s, a\}, \{v, b, t\}$.
Active: $y_{\{u,s,a\}} = 1$, $y_{\{v,b,t\}} = 0$. Adding ϵ to each,

$$y_{\{u,s,a\}} = 1 + \epsilon,\ y_{\{v,b,t\}} = \epsilon.$$

The dual constraints look like this (we hide the dual variables that are still zero).

$$\begin{aligned}
y_u + y_s + y_{u,a} = 6 + 8 + 2 &\leq w_{(u,s)} = 16.\\
y_s + y_{u,a} = 8 + 2 &\leq w_{(s,a)} = 12.\\
y_u = 6 &\leq w_{(u,a)} = 6.\\
y_u + y_v + y_{\{u,a\}} + y_{\{v,b\}} + y_{\{u,s,a\}} + y_{\{v,b,t\}} = 6 + 6 + 2 + 3 + 1 + \epsilon + \epsilon &\leq w_{(u,v)} = 20.\\
y_{\{u,a\}} + y_{\{v,b\}} + y_{\{u,s,a\}} + y_{\{v,b,t\}} = 2 + 3 + 1 + \epsilon + \epsilon &\leq w_{(a,b)} = 9.\\
y_v = 6 &\leq w_{(v,b)} = 6.\\
y_t + y_{\{v,b\}} = 9 + 3 &\leq w_{(b,t)} = 12.\\
y_v + y_t + y_{\{v,b\}} = 6 + 9 + 3 &\leq w_{(t,v)} = 19.
\end{aligned}$$

The largest value ϵ can take without violating the above constraints is 1. That makes the constraint for the edge (u, v) tight (shown in red). Set $x_{(u,v)} = 1$. Dual variables become

$$y_{\{u,s,a\}} = 2,\ y_{\{v,b,t\}} = 1,\ \text{and } y_t = 9,\ y_{\{v,b\}} = 3,\ y_s = 8,\ y_{u,a} = 2,\ y_u = y_v = 6 \text{ (remained unchanged)}.$$

Now, G' has a edges $(u, a), (v, b), (u, s), (b, t), (u, v)$. Finally, s is connected to t and u is connected to v.

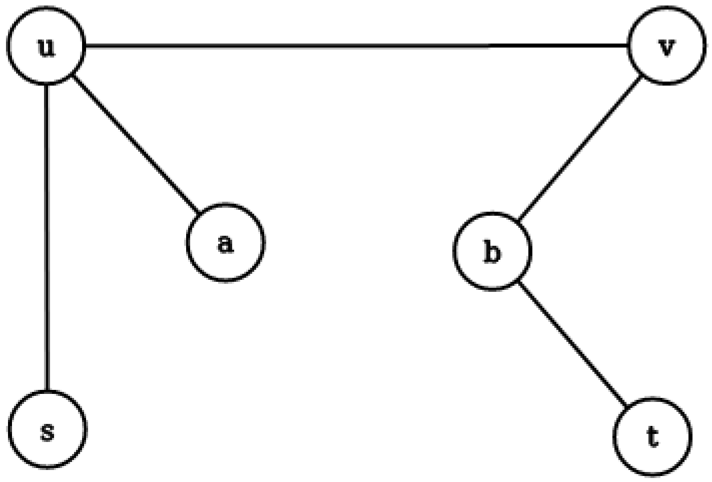

Figure 3: After iteration 5

Pruning: In the solution obtained, if removing an edge e still makes it a feasible solution then remove it. Order of removal of edges does not matter. So the final solution would be to just remove the edge (u, a) and we obtain $\{(v, b), (u, s), (b, t), (u, v)\}$ as the Steiner forest, whose cost is 54. Compare this to the optimal solution $\{(u, a), (s, a), (a, b), (b, v), (b, t)\}$, whose cost is 45.

Exercise: Run this algorithm for finding shortest path and MST(Minimum Spanning Tree) in the graph are match its result with the optimal answer.

3 Analysis of the Algorithm

Claim 17.1. *The above algorithm gives a 2-approximate solution for Steiner forest.*

This section is devoted to prove this claim. Let F' be the final set of edges obtained after the pruning step. Now we will analyze the complementary slackness conditions with respect to our solution. The first (dual) condition which was

$$x_e \neq 0 \implies \sum_{\substack{S \subseteq V \\ S \text{ is interesting} \\ \text{and } e \in \delta(S)}} y_S = w_e$$

is always satisfied because we have used this only to proceed in the algorithm.

The second (primal) condition which says

$$y_S \neq 0 \implies \sum_{e \in \delta(S)} x_e = 1 \qquad i.e,\ |F' \cap \delta(S)| = 1$$

is desired. It means that final solution should take exactly one edge from each $\delta(S)$ s.t. $y_S \neq 0$. Recall that we did not care about this condition in the algorithm. Note that, in the discussed example, $y_u \neq 0$, but there are two edges from $\delta(u)$ in the final solution. So, certainly our solution cannot be guaranteed to be optimal.

We have seen a usual technique to bound the approximation factor, which requires the complementary slackness conditions to be satisfied approximately. Towards this, let us define a variable γ as

$$\gamma = \max_{\substack{S \\ y_S \neq 0}} |F' \cap \delta(S)|$$

Exercise: Prove that F' is γ-approx solution.

However, this approach does not give us a 2-factor guarantee. One can construct examples where γ is larger than 2.

We will do a slightly different analysis to prove that the above algorithm is 2-approx. Note that it will be sufficient to show that

$$\begin{aligned} \sum_e x_e w_e &\leq 2 \sum_S y_S \qquad (1) \\ \text{i.e., (cost of our primal solution)} &\leq 2 \times \text{(cost of our dual solution)} \end{aligned}$$

Why? Because cost of our dual solution is at most the optimal dual value, which is equal to optimal primal value, which in turn, is at most the cost of optimal Steiner forest (optimal Steiner forest is the optimal solution of the integer LP).

To prove (1) let us first write the primal cost as

$$\sum_e x_e w_e = \sum_{e \in F'} w_e = \sum_{e \in F'} \sum_{S : e \in \delta(S)} y_S = P \text{ (say)}.$$

The last equation holds because all edges in F' are tight. Let us write the dual cost as simply

$$\sum_S y_S = D \text{ (say)}.$$

At the beginning of the algorithm, both the sums P and D are zero, since $y_S = 0$ for all S . As the dual variables increase, we will see how much do the sums P and D increase relative to each other. At any iteration of the algorithm, suppose there are k active sets (dual variables) $y_{S_1}, y_{S_2}, \ldots, y_{S_k}$ and suppose we add ϵ to each of these variables in this iteration. **The total increase in D will be exactly $k\epsilon$ in this iteration.**

To calculate total increase in P, we add up the increase in the sum $\sum_{S:e\in\delta(S)} y_S$ for each edge $e \in F'$. First consider an edge e whose both end-points are inside an active set, say S_i. Note that $e \notin \delta(S_j)$ for any $1 \leq j \leq k$. Thus, there is no increase in the $\sum_{S:e\in\delta(S)} y_S$.

Now, consider any edge e whose end-points are in two different active sets, say S_{i_1}, S_{i_2}. Thus, $e \in \delta(S_{i_1})$ and $e \in \delta(S_{i_2})$, but $e \notin \delta(S_j)$ for any j other than i_1 or i_2. For this edge e, the sum $\sum_{S:e\in\delta(S)} y_S$ will see an increase of exactly 2ϵ. We can have at most $k-1$ edges going across active sets because F' is a forest. **Thus, the total increase in P will be at most $2\epsilon(k-1)$ in this iteration.**

In conclusion, $P \leq 2((k-1)/k) \times D \leq 2D$, which is the desired inequality.

Exercise (Survivable Network Desgin): In this problem, we are given pairs of vertices along with numbers $(s_1, t_1, n_1), (s_1, t_1, n_2), ..., (s_k, t_k, n_k)$ and want to find out the minimum weight subgraph which has at least n_i edge-disjoint paths between s_i and t_i, for each i. Design a similar algorithm for this problem using the same primal dual scheme seen in this lecture. Is it a 2-approx algorithm?