1 April 9th, 2019

Review

Steiner Forest problem Given a graph $G = (v, E)$ with positive edge weights c_e for each edge $e \in E$ and connectivity requirements

$$r(u,v) = \begin{cases} 1 & \text{if } u,\, v \text{ must be connected} \\ 0 & \text{otherwise} \end{cases}.$$

- $S^\star$: Collection of subsets $S \subseteq V$ such that for some (u, v) for which some $r(u, v) = 1$, for some $u \in S$ and $v \notin S$.
- $\delta(S)$: Set of all edges crossing cut $(S, \overline{S}$.

1.1 Primal Dual for Steiner Forest

As an LP, this problem is:

$$\begin{aligned} \text{minimize:} \quad & \sum_e c_e x_e \\ \text{subject to:} \quad & \sum_{e:e\in\delta(S)} x_e \geq 1, \quad \forall S \in S^\star \\ & x_e \in \{0,1\} \end{aligned}$$

Dual LP:

$$\begin{aligned} \text{maximize} \quad & \sum_{S\in S^\star} y_s \\ \text{subject to:} \quad & \sum_{S\in S^\star : e\in\delta(S)} y_s \leq c_e, \quad \forall e \in E \\ & y_s \geq 0 \end{aligned}$$

In words, we are raising the dual of each set $S^\star$ while making sure that no edge leaving it is becoming over-tight.

The number of dual dual variables is equal to the number of cuts for which an edge must cross the cut, which could be exponential. However, many y_s will remain 0, so it will still run in polynomial.

Consider the following:

- Set $y_{\{t_1\}} = 1$, picking all the edges leaving t_1.
- Doing this, all sets in $S^\star$ will be satisfied, as they all have some (t_1, v) crossing the cut.
- The solution has 5, but in general it might have k (for K_k). However, the lower bound is only 1, making the lower bound useless.

The break through idea is to raise the dual variable simultaneously. We can do this beaus any dual feasible solution is a valid lower bound, so we can grow the duals in any way. Note that for this example, we can raise all dual variables to 0.5, meaning our lower bound is $\frac{k+1}{2}$. (Note that we are only concerned with the y_S for single term sets, with all others still 0).

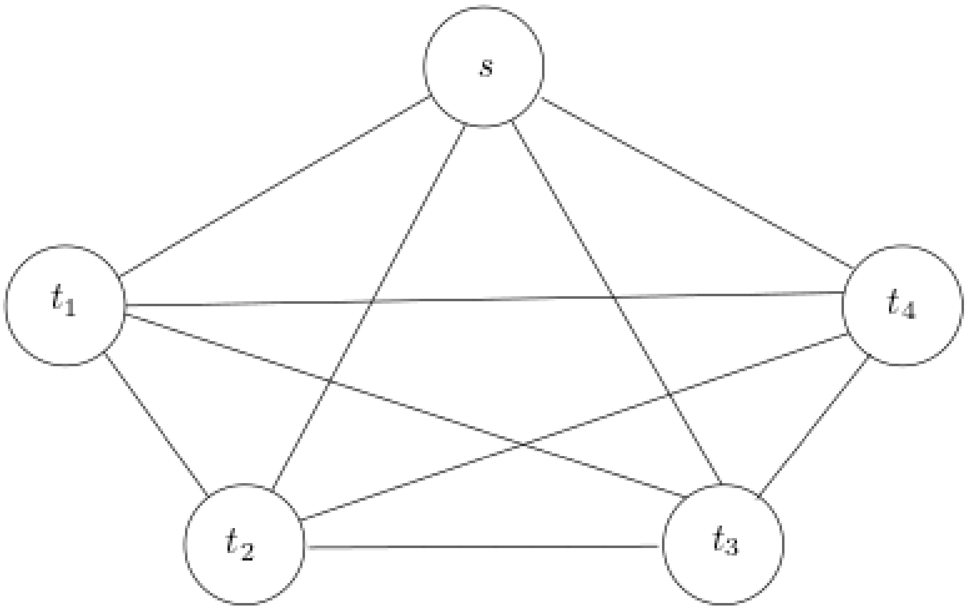

Figure 1: Complete graph with all edges with weight 1. Note that all subsets are in $S^\star$

1.2 Approximation Algorithm for the Steiner Forest Problem

Remark 1.1 — NEW IDEA: Raise duals in a synchronized manner. We are not trying to satisfy a singly unsatisfied primal constraint, but we are trying out many possibilities at the same time.

This might seem ad hoc at first, but it turns out that it could be applied to many situations. Instead of raising one dual at a time, raise them all together.

Remark 1.2 — Terminology:

- We say that edge e **feels** dual y_s if $y_s > 0$ and $e \in \delta(S)$.
- We say that edge e is **tight** if the total amount of dual it feels equals its cost, i.e.
$$\sum_{S \in S^\star : e \in \delta(S)} y_e = c_e.$$
- We say that edge s is **over-tight** if the total amount of dual it feels exceeds its cost.
- We say that a set S has been **raised** if $y_S > 0$

Theorem 1.3

The dual is trying to maximize the sum of dual variables y_S subject to the condition that no edge feels more dual than its cost - i.e. no edge is over-tight.

Let us consider how out algorithm will work with this example.

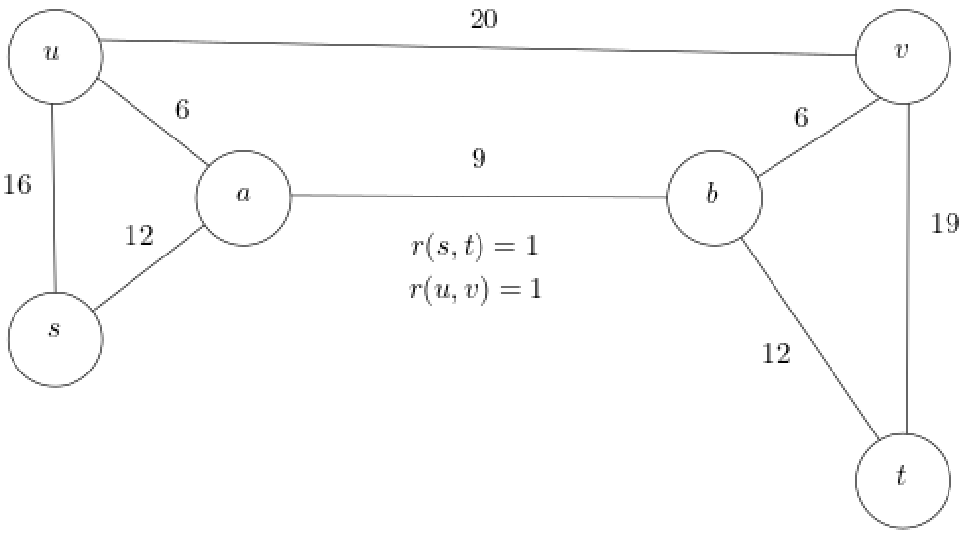

Figure 2: Optimal Solution: $(u, a), (s, a), (a, b), (b, t), (b, c)$

- Let us consider the single vertex cuts of $\{u\}, \{s\}, \{v\}, \{t\}$. Note that $\{a\}, \{b\} \notin S^\star$. We call a, b **inactive components**
- We can raise all of these dual by 6, which will make $(u, a), (b, v)$ tight.
- If we follow the logic in the past, we might create a cycle, so for this example, we pick (u, a).
- This means that $\{s\}, \{u, a\}, \{v\}, \{t\}$ are the new **active** components since they are in $S^\star$.
- Now we raise the duals of the new active components, note that $\{u\}$ is inactive, but $\{u, a\}$ is active.
- We do this by going through all edges and finding the most serious constraint. Since (v, b) is already tight, we raise them by 0 and pick (v, b).

> **Remark 1.4** — For each iteration, we pick exactly one edge, and we are allowed to raise the duals by 0.

- We raise $\{s\}, \{u, a\}, \{v, b\}, \{t\}$ by 2, which makes (s, u) tight, and so we pick it.
- Now we have three active components $\{u, a, s\}, \{v, b\}, \{t\}$, and we could raise them by 1, making (b, t) tight.

> **Remark 1.5** — Make sure you keep track if the edge is adjacent to 1 or 2 active components.

- Now we have two active components $\{u, a, s\}, \{b, v, t\}$. We can raise them by 1 each, making (u, v) tight.
- After this, we have 1 component which is inactive.

> **Remark 1.6** — After doing this, we need to check if every edge is needed, as some are redundant, e.g. (u, a). (Special pruning step).

- The solution we return is $16 + 20 + 6 + 12 = 44$.
- The lower bound is $6 \times 4 + 2 \times 4 + 1 \times 3 + 1 \times 2 = 37$.

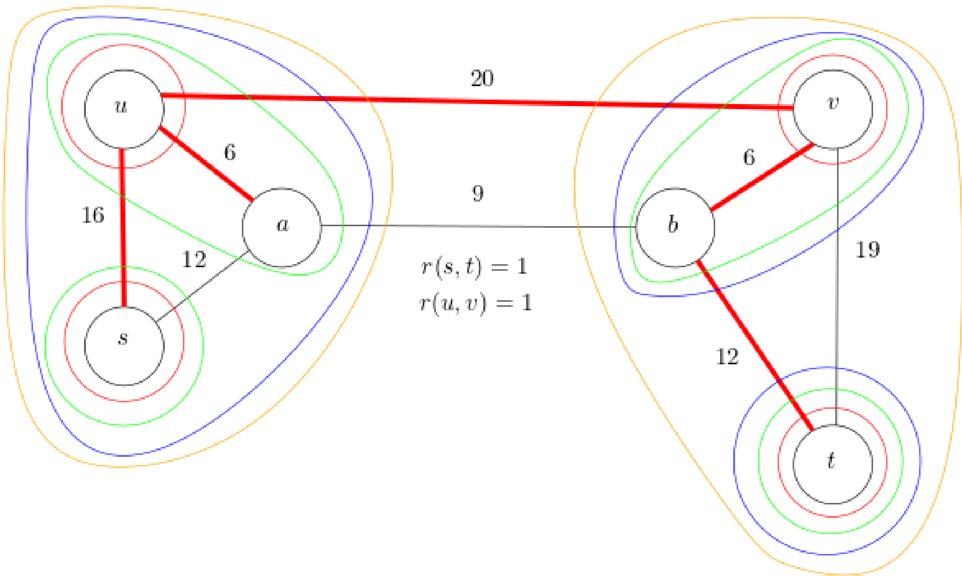

Figure 3: Example of the Algorithm

Algorithm 1.7

- Say the algorithm has picked some edges so far, which forms a forest F.
- We say that a set S is **unsatisfied** if $S \in S^\star$ but there is no picked edge crossing the cut $(S, \overline{S})$.
- Clearly if F is not primal feasible, then there is a connected component in F that is unsatisfied. We say that this component is **active**.
- In each iteration, we raise the dual of each active component until some edge goes tight. We pick one of the tight edges, and repeat.
- We stop when all connectivity requirements are satisfied, i.e. no sets are unsatisfied.
- Finally do the pruning step by removing the redundant edges.