Primal-Dual Algorithm for Steiner Forest[1]

- ***The Steiner Forest Problem.*** In this problem, we are given an undirected graph $G = (V, E)$ with edge weights $c(e) \geq 0$. Furthermore, we are given a collection of k pairs of vertices $\{s_i, t_i\}_{i=1,\ldots,k}$. The objective is to find a subgraph $H = (U, F) \subseteq G$ such that s_i is connected to t_i in H, for all $1 \leq i \leq k$, and $\texttt{cost}(F) = \sum_{e \in F} c(e)$ is minimized.

  The Steiner forest problem generalizes several problems:

  - The shortest path problem ($k = 1$)
  - The minimum spanning tree problem ($k = \binom{|V|}{2}$, all points connected)
  - The Steiner tree problem ($k = \binom{|R|}{2}$ corresponding to some required $R \subseteq V$)

  As with the Steiner Tree problem, the output is acyclic, since it is always possible to break a cycle (which removes an edge) and span the same vertices. Furthermore, the output takes the form of a collection of disjoint trees, hence the name "Steiner forest." In this lecture, we will see a clever 2-approximation using a primal-dual algorithm.

- *The Primal LP.* The primal LP relaxation is standard: we assign a variable $x_e \in [0, 1]$ to each edge to determine if (or rather, "how much") it is chosen, with the objective of minimizing $\sum_{e \in F} c(e)x_e$. But how can we express the constraint that all required terminal pairs are connected? One way is to consider "blocking" subsets $S \subseteq V$ which satisfy $|\{s_i, t_i\} \cap S| = 1$ for some $1 \leq i \leq k$, that is, the subset S "separates" s_i and t_i. Observe that s_i is connected to t_i in H if and only if $|\partial_H S| \geq 1$ for any S sepearating s_i and t_i. Let $\mathcal{B}$ be the set of all such blocking subsets. We can therefore add the constraint

  $$\forall S \in \mathcal{B} : \sum_{e \in \partial_H S} x_e \geq 1$$

  Since the number of blocking sets is exponential (there are $2^{|V|-2}$ sets which exclude a given vertex and contain another), it seems unwieldy to assign a constraint for each one. We could alternatively formulate our constraints in terms of *flow* by creating auxiliary variables $f_e^{(i)}$ representing flow (where x_e is the capacity of each edge) and stating that each required terminal connection must have at least 1 flow. Although this uses only $k|F|$ constraints, we will proceed with the LP as expressed above, and show how to use it to design our approximation algorithm. This lecture demonstrates that having many constraints is not necessarily bad.

- *The Dual LP.* We can mechanically extract the dual LP as follows:

$$\begin{array}{llll} \text{minimize} & \displaystyle\sum_{e \in E} c(e)x_e & \qquad \text{maximize} & \displaystyle\sum_{S \in \mathcal{B}} y_S \\ \text{subject to} & \displaystyle\sum_{e \in \partial_H S} x_e \geq 1, \quad \forall S \in \mathcal{B} & \qquad \text{subject to} & \displaystyle\sum_{S \in \mathcal{B} : e \in \partial_H S} y_S \leq c(e), \quad \forall e \in E \\ & x_e \geq 0, \quad \forall e \in E & & y_S \geq 0, \quad \forall S \in \mathcal{B} \end{array}$$

[1] *Lecture notes by Finn Hulse, lightly edited by Deeparnab Chakrabarty. Last modified : 25th Feb, 2022*
These have not gone through scrutiny and may contain errors. If you find any, or have any other comments, please email me at deeparnab@dartmouth.edu. Highly appreciated!

Note that in the primal we don't need to upper bound x_e because any x_e value above 1 could be reduced at the benefit of the program without violating a constraint.

- *The Primal-Dual Schema for Steiner Forest.* Although the primal LP has an absurd number of constraints, the corresponding dual has major benefits: we only need to worry about satisfying $|E|$ constraints and we have the flexibility to increase y_S on any blocking subset $S \in \mathcal{B}$ we choose. We initialize $\mathcal{A}$, our set of active blocking sets, to contain the $2k$ singleton blocking sets consisting of each required terminal. Let F be an initially empty graph to which we add edges until it becomes a valid Steiner Forest. We initialize all y values to 0, and at each stage of the algorithm, we raise the y values of the active blocking sets until at least one edge e becomes tight ($\sum_{S:e\in\partial_H S} y_S = c(e)$). At this point, we add e to F and consolidate e with the blocking set(s) which make it tight.

```
1: procedure PD-STEINER-FOREST($G, \{\{s_1,t_1\},\dots\{s_k,t_k\}\}, c(e)$ on edges):
2:     $\mathcal{A} \leftarrow \{\{s_1\},\{t_1\},\dots\{s_k\},\{t_k\}\}$
3:     $F \leftarrow \varnothing$
4:     ▷ Invariant : every $S \in \mathcal{A}$ is connected in $F$.
5:     while $\mathcal{A} \neq \varnothing$ do
6:         Raise $y_S$ for all $S \in \mathcal{A}$ until some edge $(u,v)$ becomes tight (assume $u \in S_1$)
7:         Add $(u,v)$ to $F$
8:         if $v \in S_2$ for some $S_2$ then ▷ Replace $S_1$ and $S_2$ with $S_1 \cup S_2$ if it is a valid blocking set
9:             Remove $S_1$ and $S_2$ from $\mathcal{A}$
10:            if $S_1 \cup S_2 \in \mathcal{B}$ then
11:                Add $S_1 \cup S_2$ to $\mathcal{A}$
12:        else ▷ Replace $S_1$ with $S_1 \cup v$ if it is a valid blocking set
13:            Remove $S_1$ from $\mathcal{A}$
14:            if $S_1 \cup v \in \mathcal{B}$ then
15:                Add $S_1 \cup v$ to $\mathcal{A}$
16:    ▷ At this stage $F$ connects every terminal; however it may contain cycles.
17:    (Reverse Delete:) Consider the edges of $F$ in the reverse order they were added, deleting them if they are unnecessary.
18:    return $F$
```

Note that two active blocking sets S_1 and S_2 can simultaneously make an edge e tight: in that case, we "merge" those sets as $S_1 \cup S_2$, which we keep in $\mathcal{A}$ if it is a blocking set. It is possible that adding e causes a blocking set to no longer be a blocking set, in which case the desired path has been achieved and we can remove the set altogether from $\mathcal{A}$. Since the goal is to connect every terminal pair, this loop only terminates once $\mathcal{A} = \varnothing$. Lastly, we go backwards in order of edges added and remove unnecessary edges.

- ***Analysis.***

Theorem 1. PD-STEINER-FOREST is a 2-approximation algorithm.

Proof. Suppose there are T loops and label $\mathcal{A}_t$ to be the state of $\mathcal{A}$ at step t. Step t uniformly distributes some Δ_t mass onto the y values of all elements of $\mathcal{A}_t$. Therefore, the value of the dual at the end of the algorithm is

$$\sum_{S\in\mathcal{B}} y_S = \sum_{t=1}^{T} \Delta_t \cdot |\mathcal{A}_t| \tag{1}$$

Now, an edge e is added to F only if the total y incident on e equals $c(e)$. Therefore, we get that

$$\forall e \in F: \quad c(e) = \sum_{S: e\in\partial S} y_S = \sum_{t=1}^{T} \sum_{S\in\mathcal{A}_t: e\in\partial S} \Delta_t$$

and hence the total cost of F is

$$\begin{aligned}\texttt{cost}(F) = \sum_{e\in F}\left(\sum_{t=1}^{T} \sum_{S\in\mathcal{A}_t: e\in\partial S} \Delta_t\right) &= \sum_{t=1}^{T}\Delta_t \sum_{S\in\mathcal{A}_t}\left(\sum_{e\in F: e\in\partial S} 1\right) \\ &= \sum_{t=1}^{T}\Delta_t \sum_{S\in\mathcal{A}_t} |F\cap\partial S| \qquad (2)\end{aligned}$$

Now, for a fixed $S \in \mathcal{A}_t$, the number $|F \cap \partial S|$ can be arbitrarily large, and it seems unclear how to bound the RHS of (2). Note that we would like to compare this with (1). What saves the day is the *reverse delete* step, and the fact that F is a forest. We elaborate on this.

Fix some $1 \le t \le T$ and consider the graph $F[\mathcal{A}_t]$ induced on the graph F by the disjoint sets $\mathcal{A}_t$. More precisely, we have a node for every $S \in \mathcal{A}_t$ and a link between S_i and S_j iff there is some $(u, v) \in F$ such that $u \in S_1$ and $v \in S_2$. Figure 1 illustrates what such a graph might look like. The

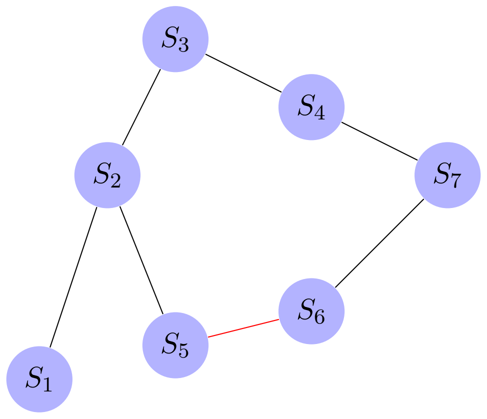

Figure 1: Example of $F[\mathcal{A}_t]$, the graph induced on F by $\mathcal{A}_t$. The red edge wouldn't exist after reverse delete.

main observation is the following.

Claim 1. $F[\mathcal{A}_t]$ is a forest.

Proof. Suppose not, and suppose $F[\mathcal{A}_t]$ contains a cycle as purported in the picture above. Consider the last edge $e = (u, v)$ that was added to F with $u \in S_i$ and $v \in S_j$; in the picture this is the red edge and $i = 5$ and $j = 6$. By the invariant of the algorithm, right before (u, v) was added to F, all the S_i's are connected, and in particular, there is a path from u to v in $F \setminus e$. In turn, *any* pair of vertices, in particular any pair of terminals s_a, t_a which were connected in F at this point, is also connected in $F \setminus e$. Therefore, in the *reverse delete* step, this edge e must have been deleted. This is the contradiction which establishes the claim. $\square$

Now we are almost done. Since $F[\mathcal{A}_t]$ *must* be acyclic, the total number of edges is at most $|A_t| - 1$. Returning to (2), we get

$$\mathtt{cost}(F) \leq \sum_{t=1}^{T} \Delta_t \sum_{S \in \mathcal{A}_t} |F \cap \partial S| < \sum_{t=1}^{T} 2|\mathcal{A}_t|\Delta_t$$

Comparing with (1), we get that $\mathtt{cost}(F) < 2\mathtt{dual} \leq 2\mathtt{opt}$, completing the proof of the theorem. $\square$

Notes

The first 2-approximation for Steiner Forest is in the paper [1] by Agrawal, Klein, and Ravi. The exposition above is from the influential paper [2] by Goemans and Williamson which also gives primal-dual algorithms for many other “constrained forest” problems. The integrality gap of the LP is also 2, even for the case when all pairs are terminals (the spanning tree problem), and the above algorithm remains the best approximation algorithm known for the Steiner forest problem to date. A strengthening of this relaxation is proposed in the paper [3] by Könemann, Leonardi, Schäfer, and van Zwam; unfortunately, that relaxation also has a gap arbitrarily close to 2.

References

[1] A. Agrawal, P. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized Steiner problem in networks. *SIAM J. Comput.*, 24:440–456, 1995.

[2] M. X. Goemans and D. P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing (SICOMP)*, 24(2):296–317, 1995.

[3] J. Könemann, S. Leonardi, G. Schäfer, and S. H. van Zwam. A group-strategyproof cost sharing mechanism for the steiner forest game. *SIAM Journal on Computing (SICOMP)*, 37(5):1319–1341, 2008.