IOE 691: Approximation Algorithms Date: 03/29/2017

Lecture Notes: Prize-Collecting Steiner Tree (Primal-Dual)

Instructor: Viswanath Nagarajan Scribe: Yanzhe Lei

1 Prize-Collecting Steiner Tree Problem

The Prize-Collecting Steiner Tree Problem (PCST) is an extension of the Steiner Tree Problem where each vertex left out of the tree pays a penalty. The goal is to find a tree that minimizes the sum of its edge costs and the penalties for the vertices left out of the tree.

Consider a undirected graph $G = (V, E)$ with non-negative cost for the edges $e \in E$. There is a unique root $r \in V$. The goal of PCST problem is to find a tree T with root being r such that its total cost is minimized, where the total cost of T is defined as the sum of all cost of the included edges plus the sum of the penalties $\pi_u \geq 0$ for any $u \in V$ not included in the tree. Mathematically, the cost of a tree T is defined as $\text{cost}(T) = \sum_{e \in T} c_e + \sum_{u \notin T} \pi_u$.

We now formulate PCST as an integer program. Let x_e be the indicator variable with $x_e = 1$ denoting $e \in T$. Let z_u be the indicator variable with $z_u = 1$ denoting $u \notin T$. For any set of vertices $S \subseteq V$, let δS be the set of edges that lie on the boundary of S. The LP relaxation of the integer programming formulation of the PCST problem is given as follow.

$$
\begin{aligned}
\text{(P')} \quad \min \quad & \sum_{e \in E} c_e x_e + \sum_{u \in V} \pi_u z_u \\
s.t. \quad & \sum_{e \in \delta S} x_e + z_u \geq 1 \qquad \forall u \in V, S \ni u, S \subseteq V \setminus \{r\} \\
& x_e \in \{0, 1\} \qquad \forall e \in E \\
& z_u \in \{0, 1\} \qquad \forall u \in V
\end{aligned}
$$

The above formulation is not so convenient, since for every vertex u, the number of set S such that $S \ni u$ is exponential in $|V|$. Therefore, we use the indicator variable for every subset of V instead of the indicator variable for every vertex. Formally, for every $R \subseteq V$, we define the indicator variable z_R with $z_R = 1$ when R is the set of vertices that are not covered by the tree T. For any set $S \subseteq V$, define $\pi(S) = \sum_{u \in S} \pi_u$. (P') can then be formulated equivalently as follow.

$$
\begin{aligned}
\text{(P)} \quad \min \quad & \sum_{e \in E} c_e x_e + \sum_{R \subseteq V} \pi(R) z_R \\
s.t. \quad & \sum_{e \in \delta S} x_e + \sum_{S: R \supseteq S} z_R \geq 1 \qquad \forall S \subseteq V \setminus \{r\} \\
& x_e \in \{0, 1\} \qquad \forall e \in E \\
& z_R \in \{0, 1\} \qquad \forall R \subseteq V
\end{aligned}
$$

We can further write the linear relaxation of the above integer program. By abuse of notation both formulations will be called (P).

$$
\begin{aligned}
\text{(P)} \quad \min \quad & \sum_{e \in E} c_e x_e + \sum_{R \subseteq V} \pi(R) z_R \\
s.t. \quad & \sum_{e \in \delta S} x_e + \sum_{S: R \supseteq S} z_R \geq 1 \qquad \forall S \subseteq V \setminus \{r\} \\
& x_e \geq 0 \qquad \forall e \in E \\
& z_R \geq 0 \qquad \forall R \subseteq V
\end{aligned}
$$

By assigning dual variable y_S to each of the constraint, the dual problem of the above (P) is defined as follows.

$$
\begin{aligned}
\text{(D)} \quad \max \quad & \sum_{S \subseteq V \setminus \{r\}} y_S \\
s.t. \quad & \sum_{S: e \in \delta S} y_e \leq c_e, \qquad \forall e \in E \\
& \sum_{S: S \subseteq R} y_S \leq \pi(R), \qquad \forall R \subseteq V \\
& y_S \geq 0 \qquad \forall S \subseteq V
\end{aligned}
$$

Observation 1.1 *For any feasible dual solution y, if $S, S' \subseteq V \setminus \{r\}$ are both tight for the penalty constraint, then so are $S \cap S'$ and $S \cup S'$.*

Proof: We have the following chain of inequalities:

$$
\begin{aligned}
\pi(S \cap S') + \pi(S \cup S') &\geq \sum_{T \subseteq S \cup S'} y_T + \sum_{T \subseteq S \cap S'} y_T \\
&\geq \sum_{T \subseteq S} y_T + \sum_{T \subseteq S'} y_T \\
&= \pi(S) + \pi(S') \\
&= \pi(S \cap S') + \pi(S \cup S').
\end{aligned}
$$

We explain the inequalities line by line. Note that y is a feasible solution, thus the first inequality comes from the constraint. The second one follows from a simple counting argument of subsets in $S, S', S \cup S', S \cap S'$ respectively. The third line comes from the condition that the penalty constraints for both S and S' are tight, and the fourth line again comes from a simple counting argument of elements in $S, S', S \cup S'$ and $S \cap S'$.

This chain of inequalities implies that $\pi(S \cap S') + \pi(S \cup S') = \sum_{T \subseteq S \cup S'} y_T + \sum_{T \subseteq S \cap S'} y_T$, which holds if and only if $S \cup S'$ and $S \cap S'$ are both tight. ■

2 Primal-Dual Algorithm for PCST Problem

We now describe the primal-dual algorithm introduced in Goemans and Williamson (1995). The algorithm maintains a forest F of edges, a set of connected component $\mathcal{B}$, and a subset of $\mathcal{B}$, denoted by $\mathcal{C}$, which contains all connected components that are "active". Initially, F is empty, hence $\mathcal{B}$ is the collection of all the singletons. All components except the root r are considered active, hence $\mathcal{C}$ equals to $\mathcal{B}$. The algorithm loops, in each iteration doing one of two things. First, the algorithm may add an edge between two connected components of F. If the resulting component contains the root r, it becomes inactive; otherwise it is still active. Second, the algorithm may decide to "deactivate" a component. Intuitively, a component is deactivated if the algorithm decides it is willing to pay the penalties for all vertices in the component. The main loop terminates when all connected components of F are inactive. The algorithm stops by removing some edges satisfying certain properties.

Algorithm 1 Primal-Dual algorithm for PCST

Initialization. Set $F = \emptyset$, $y_S = 0$ for all $S \subseteq V \setminus \{r\}$, $\mathcal{B} = \{\{u\} : u \in V\}$, $\mathcal{C} = \{\{u\} : u \in V \setminus \{r\}\}$.
while $\mathcal{C}$ is not empty **do**
  Uniformly increase the dual variables y_R for all $R \in \mathcal{C}$ until:
    either (i) $\sum_{S:e' \in \delta S} y_S = c_{e'}$ for some $e' \in \delta(R')$ where $R' \in \mathcal{C}$
    or (ii) $\sum_{S:S \subseteq R'} y_S = \pi(R')$ for some $R' \in \mathcal{C}$
  if (i) **then**
    Include e' in F, i.e. $F \leftarrow F \cup \{e'\}$
    Merge the two components connected by e'
  else
    Set R' to be inactive, i.e. $\mathcal{C} \leftarrow \mathcal{C} \setminus \{R'\}$
  Update $\mathcal{B}$ and $\mathcal{C}$ accordingly
Let the connected component in $\mathcal{B}$ that contains r be F.
Obtain tree T from F as follows. For all edges $e \in F$, delete e if $\pi(F_e) = \sum_{S \subseteq F_e} y_S$, where F_e is the connected component in $F \setminus \{e\}$ that does not contain r.
Return the tree T

We now characterise the performance of algorithm 1.

Theorem 2.1 *Algorithm 1 is a 2-approxiamtion algorithm for PCST.*

Remark 2.1 *In fact, we will show that the primal-dual algorithm has a performance that is even stronger than 2-approximation. Specifically, we show that the output of the algorithm, T, satisfies $\sum_{e \in T} c_e + 2 \cdot \sum_{u \notin T} \pi_u \leq 2 \cdot OPT$, where OPT is the minimum cost Steiner Tree. The fact that the multiplicand 2 in front of the penalty term is the same as the approximation ratio is called Lagrangian multiplier preserving property. As we shall discuss in the next lecture, this is useful in obtaining a constant approximation algorithm for the related k-MST problem.*

2.1 Proof of Theorem 2.1

We first make the following observation.

Lemma 2.1 *At each step of the algorithm, the dual variable y_S is always feasible.*

Proof. Let $\mathcal{B}$ be the connected components at any point in the algorithm. The dual constraints corresponding to edge costs are trivially satisfied by construction. For the dual constraints corresponding to the set penalties, we proceed by induction and consider three cases:

- $R \in \mathcal{B}$. The penalty constraint is ensured to be satisfied by construction.
- $R \subsetneq S' \in \mathcal{B}$. There is no increase in $\sum_{S:S\subseteq R} y_S$ from this point onward.
- Now consider any other $R \subseteq V$. The key is to notice that the dual variable $y_S > 0$ only if S is a subset of the elements in $\mathcal{B}$. Therefore,

$$\sum_{S:S\subseteq R} y_S = \sum_{S:S\subseteq R} y_S \cdot \mathbf{1}\{S \subseteq S' \text{ for some } S' \in \mathcal{B}\} = \sum_{S'\in\mathcal{B}} \sum_{S:S\subseteq R\cap S'} y_S \leq \sum_{S'\in\mathcal{B}} \pi(R\cap S') \leq \pi(R)$$

  The inequality holds because the penalty constraint of $R \cap S'$ is satisfied (previous case).

Using these properties, we can show by induction that all penalty constraints are satisfied. ■

Since the dual variables are always feasible, by weak duality it suffices to show the following lemma.

Lemma 2.2 *Let T be the output of algorithm 1 and y_S be the dual variables at the end of the while loop. The following inequality holds:*

$$\sum_{e\in T} c_e + 2 \cdot \sum_{u\notin T} \pi_u \leq 2 \cdot \sum_{S\subseteq V\setminus\{r\}} y_S \tag{1}$$

Proof: By definition of the algorithm, $c_e = \sum_{S:e\in\delta S} y_S$ for all $e \in T$. Hence

$$\sum_{e\in T} c_e = \sum_{e\in T} \sum_{S:e\in\delta S} y_S = \sum_{S\subseteq V\setminus\{r\}} y_S \cdot |T \cap \delta S| \tag{2}$$

where the last equality holds by simply rearranging the summations.

For every node $u \notin T$, it either belongs to the inactive set that does not contain r (call the corresponding component R_u), or it belongs to F_e for some e removed in line 5 of algorithm 1. If it is the first scenario, by the deactivating condition (cf. the second condition in line 3), $\pi(R_u) = \sum_{S\subseteq R_u} y_S$. If it the second scenario, by the condition under which we remove edge e (cf. the second condition in line 5), $\pi(F_e) = \sum_{S\subseteq F_e} y_S$. Since any two connected components does not overlap with each other, it is clear that

$$\sum_{u\notin T} \pi_u \leq \sum_{S\subseteq V\setminus T} y_S \tag{3}$$

Combining (2) and (3), we know that the LHS of (1) is upper bounded by

$$P(y) := \sum_{S\subseteq V\setminus\{r\}} y_S \cdot |T \cap \delta S| + 2 \cdot \sum_{S\subseteq V\setminus T} y_S.$$

Let $D(y) = \sum_{S\subseteq V\setminus\{r\}} y_S$. We then want to show that $\frac{dP}{dt} \leq 2 \cdot \frac{dD}{dt}$, where t is ticking whenever some dual variable increases.

Fix any time t and the corresponding components $\mathcal{B}$ and $\mathcal{C}$. Define $\mathcal{C}_1 := \{S \in \mathcal{C} : S \nsubseteq V \setminus T\}$ and $\mathcal{C}_2 := \{S \in \mathcal{C} : S \subseteq V \setminus T\}$. We know that

$$\frac{dP}{dt} = \sum_{S \in \mathcal{C}} |T \cap \delta S| + 2 \cdot \sum_{S \in \mathcal{C}_2} 1 = \sum_{S \in \mathcal{C}_1} |T \cap \delta S| + 2 \cdot |\mathcal{C}_2|$$

We also know that $\frac{dD}{dt} = 2 \cdot |\mathcal{C}|$. Since $\mathcal{C}_1$ and $\mathcal{C}_2$ are disjoint subsets that span $\mathcal{C}$ together, we only need to show that $\sum_{S \in \mathcal{C}_1} |T \cap \delta S| \leq 2 \cdot |\mathcal{C}_1|$.

Let H be an auxiliary graph where the vertices correspond to components in $\mathcal{B}$. The edges in H are the edges of T that are added after time t: note that any such edge must be between distinct components of $\mathcal{B}$. Moreover, for any $S \in \mathcal{B}$, $|T \cap \delta S|$ equals the degree $deg_H(S)$ of vertex (component) S in graph H. This is because only edges added after time t may "cross" any of the components in $\mathcal{B}$. We will show

Claim 2.1 $\sum_{S \in \mathcal{C}_1} deg_H(S) \leq 2 \cdot |\mathcal{C}_1|$.

Let $\mathcal{I} \subseteq \mathcal{B} \setminus \mathcal{C}$ be the set of inactive components that have positive degree in H. Note that H is a tree on components $\mathcal{C}_1 \cup \mathcal{I}$. To prove the claim, we first note that, since H is a tree, then its average degree must be less than two, i.e. $\sum_{S \in \mathcal{C}_1 \cup \mathcal{I}} deg_H(S) = 2 \cdot |\mathcal{C}_1| + 2|\mathcal{I}| - 2$. Moreover, we show that all but one of the leaves of H must correspond to active components.

Claim 2.2 *If $S \in \mathcal{I}$ and $r \notin S$, then* $\deg_H(S) \geq 2$.

Proof: Note that $deg_H(S) \geq 1$ as $S \in \mathcal{I}$. For a contradiction, assume that $\deg_H(S) = 1$. Let $e \in T$ be the unique edge in T connecting to S. Let $e_1, \cdots, e_k$ be edges with at least one end-point in S that get removed in obtaining tree T from F; also let $F_1, \cdots, F_k$ denote the corresponding subtrees that are tight for the penalty constraint. So $\pi(F_i) = \sum_{R \subseteq F_i} y_R$ for all $i = 1, \cdots, k$.

As $S \in \mathcal{I}$ and $r \notin S$ we also have $\pi(S) = \sum_{R \subseteq S} y_R$. By Observation 1.1, it follows that $\left(\cup_{i=1}^k F_i\right) \cup S$ is also tight for the penalty constraint. Note that $F_e = \left(\cup_{i=1}^k F_i\right) \cup S$ is the subtree (not containing r) that corresponds to edge e in tree F. As F_e is tight for the penalty constraint, edge e must have been removed in T, a contradiction! $\blacksquare$

Now we prove Claim 2.1.

$$\begin{aligned}
\sum_{S \in \mathcal{C}_1} deg_H(S) &= \sum_{S \in \mathcal{C}_1 \cup \mathcal{I}} deg_H(S) - \sum_{S \in \mathcal{I}} deg_H(S) \\
&\leq \sum_{S \in \mathcal{C}_1 \cup \mathcal{I}} deg_H(S) - 2(|\mathcal{I}| - 1) - 1 \\
&\leq 2 \cdot |\mathcal{C}_1| + 2|\mathcal{I}| - 2 - 2(|\mathcal{I}| - 1) - 1 \\
&\leq 2 \cdot |\mathcal{C}_1|
\end{aligned}$$

This completes the proof that $\frac{dP}{dt} \leq 2 \cdot \frac{dD}{dt}$ and hence Lemma 2.2. $\blacksquare$

References

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