COM S 783 / OR&IE 634 Approximation Algorithms **Fall Semester, 2003**

Lecture 16-17: November 3, 2003

Lecturer: Yuval Rabani *Scribe: Anirban Dasgupta*

1 Introduction

In the last lecture, we were giving a primal-dual algorithm for the Steiner tree problem. We are given the graph $G(V, E)$, a cost function on the edges $c : E \to \mathbb{Q}^+$ and set of terminal nodes $T \subset V$. The objective is to find out a connected subgraph that spans T.

Let F' be the set of edges computed in the first phase of the algorithm and F be the edge set after cleanup. Then

Lemma 1. At the end, the final y values form a dual feasible solution and F is a primal feasible solution.

Proof: That the final values of y are feasible is obvious from the algorithm. We prove that F is a feasible primal. To show this, we only need to argue that F spans all terminals. Since F is obtained from F' by removing only those edges which do not affect terminal connectivity, it is sufficient to show that F' is feasible. We can show that F' is true by induction on the number of steps and using the two following facts

- The set of active cuts $S_1, S_2, \ldots$, covers all the terminals.
- The edge set always spans the active cuts.

□

Now we want to compare the cost of the two solutions. Here's some wishful thinking : let's write down the primal and dual slackness conditions.

$$x_e > 0 \;\Rightarrow\; \sum_{S|e\in\partial S} y_S = c_e: \;\; \textbf{Primal complementary slackness}$$

$$y_S > 0 \;\Rightarrow\; \sum_{e\in\partial S} x_e = 1: \;\; \textbf{Dual complementary slackness}$$

The primal slackness is always maintained by the algorithm. We do not expect the dual slackness to hold, for then we would have an exact algorithm. Instead, what we might hope for is something like this :

$$y_S > 0 \;\Rightarrow\; \sum_{e\in\partial S} x_e \le 2 \tag{1}$$

Note that this is similar to what we had in vertex cover. In case this holds, we have a 2-approximation for our problem because

$$\begin{aligned}\sum_e c_e x_e &= \sum_e \sum_{S|e\in\partial S} y_S \\ &= \sum_S y_S \sum_{e\in\partial S} x_e \\ &\leq 2\sum_S y_S\end{aligned}$$

Unfortunately our wishful thinking does not really work out. We do not have equation 1 true for all of our cuts. But the good news is that (and herein is the main contribution of the paper) we can make the claim true "on an average". Define $deg_F(S)$ to be the *number of edges in F that crosses the boundary of the cut S*. We have the following claim.

Lemma 2. If we raise y_S for all active cuts S by Δ, then the following holds:

$$\Delta \sum_{\text{active } S} deg_F(S) \leq 2\Delta(\#\text{active sets})$$

Proof: Equivalently, we want to show that

$$\sum_{\text{active } S} deg_F(S) \leq 2(\#\text{active sets})$$

Let $S_1, S_2, \ldots, S_m$ be the set of active cuts. Remove from F all edges that cross these cuts. We then get a collection of connected components, some of them included in $S_1, S_2, \ldots, S_m$ and possibly others. Contract these connected components into supernodes and consider the graph on the supernodes, plus the removed edges. This will be a tree, as F is a tree. The average degree of any tree is at most 2. But we are not yet done, as some of the supernodes might not be active components. But, the leaves are definitely all active components, and hence the average degree of the inactive components is at least 2. Thus, the average degree of the active components is at most 2. $\square$

Given the above lemmas, we have the following theorem about our algorithm.

Theorem 3. The algorithm we discussed gives a 2 approximate Steiner tree.

Proof: From Lemma 2, we have the fact that for any time period of the algorithm over which each active y_S increased by an amount Δ,

$$\Delta \sum_{\text{active } S} deg_F(S) \leq 2\Delta(\#\text{active sets})$$

which implies that

$$\sum_S deg_F(S) y_S \leq 2\sum_S y_S$$

and hence,

$$\begin{aligned}
\sum_{e\in F} c_e &= \sum_{e\in F}\sum_{e\in\partial S} y_S \\
&= \sum_{S}\sum_{F\cap\partial S} \\
&= \sum_{S} deg_F(S) y_S \\
&\leq 2\sum_{S} y_S
\end{aligned}$$

Hence, we have a 2-approximate Steiner tree. $\square$

For the **Steiner Forest** problem, the same algorithm as before works, except that the components are made inactive when their connectivity requirements are met. We get a 2-approximation.

2 Prize Collecting Steiner Tree

The input for this problem is the same as for the Steiner tree except that now each node has a penalty $p_v \geq 0$ of not being connected. We also specify a root r which must be in the tree. The goal is to find a tree rooted at r such that

$$\sum_{e\in T} c_e + \sum_{v\notin T} p_v$$

The LP-relaxation for the problem is as follows.

$$\begin{aligned}
\min\sum_{e} c_e x_e &+ \sum_{i\in V} p_i z_i \\
\sum_{e\in\partial S} x_e &\geq 1 - z_i \;\; \forall \text{cuts } S \; s.t. \; i\in S, r\notin S \\
x_e, z_i &\geq 0
\end{aligned}$$

where z_i is the indicator variable of whether a vertex is spanned and x_e indicates whether an edge is taken in the tree. The dual program is

$$\begin{aligned}
&\max\sum_{S} y_S \\
\forall e \quad &\sum_{S|e\in\partial S} y_S \leq c_e \\
\forall i \quad &\sum_{S|i\in S} y_{S,i} \leq p_i
\end{aligned}$$

Define, $y_S = \sum_{i \in S} y_{S,i}$. Then we can rewrite the LP as

$$\max \sum_{S} y_S \tag{2}$$

$$\forall e \quad \sum_{S | e \in \partial S} y_S \quad \leq \quad c_e \tag{3}$$

$$\forall T \not\ni r \quad \sum_{S \subseteq T} y_S \quad \leq \quad \sum_{i \in T} p_i \tag{4}$$

The algorithm to solve is is the same dual-ascent procedure as before, except that we now perform different actions based on which constraint get tight. Once a component joins the root-component, it is frozen. We also have a cleanup phase at the end in which we scan the edges in reverse order, and any edge whose removal does not affect the objective function or forms a cycle, is removed. Suppose, F is the final set of edges and y_S be the dual solutions. Then,

Theorem 4.

$$c(F) \quad = \quad \sum_{e \in F} c_e + \sum_{i \notin F} p_i \leq 2 \sum_{S} y_S$$

Proof: **Sketch** We prove a stronger claim :

$$\sum_{e \in F} c_e + 2 \sum_{i \notin F} p_i \leq 2 \sum_{S} y_S$$

The proof is similar to the one we did for Steiner tree. Look at any node which is not connected to the root. Every node must be in some component which was frozen. So the nodes not connected to the root must have had their components frozen before they were connected to the root. Hence, for all such nodes, the constraint 4 is tight. Thus

$$2 \sum_{i \notin F} p_i = 2 \sum_{S \not\subseteq S_r} y_S \tag{5}$$

where S_r is the component containing the root. For the component containing r, we can apply the Steiner tree argument. At any given step, if we take all the active components, and contract them to a single vertex, the edges connecting them forms a tree. Formally, consider at any given step, all he active components S that eventually connect to the root plus the component containing the root.

Lemma 5. If for all such active components S, we raise each y_S by Δ, then

$$\Delta \sum_{S \text{ that will join } r} \leq 2\Delta\#(\text{such components})$$

The proof follows the same technique as before. Combining the Lemma 5 and the equation 5, we have our 2-approximation for the prize collecting Steiner tree problem.

3 Lagrangian Relaxations

Next, we use the primal-dual schema to solve the k-MST problem. The input is an undirected graph $G(V, E)$, with edge costs $c : E \to \mathbb{Q}^+$ and a value $k \in \{1, \ldots, |V|\}$. We want to find the lowest cost subtree of G that spans k nodes. We will use the prize-collecting Steiner tree algorithm to solve the k-MST, and the particular type of approximation guarantee that we have proved is going to prove crucial.

The linear program for the problem is

$$\begin{array}{rrcl} & \min \sum_{e} c_e x_e & & \\ & \sum_{i \in V} z_i & = & n - k \\ \forall S \not\ni r, \forall i \in S & \sum_{e \in \partial S} x_e & \geq & 1 - z_i \\ & x_e, z_i & \geq & 0 \end{array}$$

The Lagrangian relaxation of this program is as follows :

$$\begin{array}{rrcl} & \min \sum_{e} c_e x_e & + & \lambda \sum_{i \in V} z_i - \lambda(n-k) \\ \forall S \not\ni r, \forall i \in S & \sum_{e \in \partial S} x_e & \geq & 1 - z_i \\ & x_e, z_i & \geq & 0 \end{array}$$

Consider the problem where the program is

$$\begin{array}{rrcl} & \min \sum_{e} c_e x_e & + & \lambda \sum_{i \in V} z_i \\ \forall S \not\ni r, \forall i \in S & \sum_{e \in \partial S} x_e & \geq & 1 - z_i \\ & x_e, z_i & \geq & 0 \end{array}$$

This is exactly the program for the prize collecting Steiner tree. Assume we have a solution T to this program such that the tree T spans exactly k nodes. It is obvious that T is a feasible solution for the k-MST. We claim that T has a good approximation guarantee.

Suppose $\mathcal{D}_{PST}$ is the dual value of this prize collecting Steiner tree program, and $\mathcal{D}_{k-MST}$ is the value from the LP-relaxation of k-MST and OPT_{k-MST} is the optimal value for k-MST. Then,

$$\begin{array}{rcl} \sum_{e \in T} c_e + 2 \sum_{i \notin T} \lambda & \leq & 2\mathcal{D}_{PST} \\ \sum_{e \in T} c_e & \leq & 2\mathcal{D}_{PST} - 2\lambda(n-k) \end{array}$$

$$
\begin{aligned}
&\leq \mathcal{D}_{k-MST} \\
&\leq OPT_{k-MST}
\end{aligned}
$$

Hence, if we can guarantee that the optimal solution to the prize collecting problem has the correct number of nodes, we have a approximation guarantee. Note that, the number of nodes in the solution depends on λ. If $\lambda = 0$, then since the penalty is zero, the optimal tree T is just $T = \{r\}$. Whereas, if $\lambda = \infty$, then everyone would like to get connected to the root and hence T is the MST. We will do binary search for λ in he range $[0, nc_{\max}]$ with $c_{\max}$ being the maximum cost of an edge. We can find two such value λ_1 and λ_2, such that $\lambda_1 \approx \lambda_2$ and the resulting trees are T_1 and T_2 with each tree T_i spanning k_i nodes, $k_1 < k < k_2$. Also, assume that the corresponding dual solutions are y_1 and y_2. Though it is not clear that we can find one specific λ which achieves k, we can always find this range. Now define μ_1 and μ_2 such that

$$
\begin{aligned}
\mu_1 + \mu_2 &= 1 \\
\mu_1(n - k_1) + \mu_2(n - k_2) &= n - k
\end{aligned}
$$

We can form a feasible fraction solution of the k-MST using a linear combination of the two solutions with weights μ_1 and μ_2.

$$
\begin{aligned}
\mu_1 c(T_1) + \mu_2 c(T_2) + 2(\mu_1\lambda_1(n - k_1) + \mu_2\lambda_2(n - k_2)) &\leq 2(\mu_1 c(y_1) + \mu_2 c(y_2)) \\
\mu_1 c(T_1) + \mu_2 c(T_2) + 2\lambda(n - k) &\leq 2(\mu_1 c(y_1) + \mu_2 c(y_2)) \\
&\leq 2\mathcal{D}_{k-MST} \\
&\leq OPT_{k-MST}
\end{aligned}
$$

where the last but one inequality is valid as $\mu_1 c(y_1) + \mu_2 c(y_2)$ is a valid feasible solution to the relaxation of the k-MST and hence is smaller than the dual.

Now, if $\mu_2 \geq 1/2$, we can just pick T_2 and trim it and would have a constant factor approximation guarantee. If μ_1 is big, we take T_1 and attach to T_1 a small portion of T_2 in order to get k nodes.

References

[1] Ajit Agrawal, Philip Klein, R. Ravi, When Trees Collide: An Approximation Algorithm for the Generalized Steiner Problem on Networks, SIAM Journal on Computing, v.24 n.3, p.440-456, June 1995.

[2] A. Blum, R. Ravi, S. Vempala, A constant factor approximation algorithm for the k-MST problem. JCSS 58(1):101-108, 1999.

[3] F. Chudak, T. Roughgarden, and D.P. Williamson. Approximate k-msts and k-steiner trees via the primal-dual method and lagrangean relaxation. IPCO 2001: 60-70.

[4] N. Garg. A 3-approximation for the minimum tree spanning k vertices. FOCS 1996, 302-309.

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