## 18.1 Introduction

In this lecture, we show how to use the *Randomized Rounding* to devise a polylogarithmic approximation algorithm for the group Steiner tree problem. Given a weighted undirected graph with some subsets of vertices called groups, the group Steiner tree problem is defined as finding a minimum-weight subgraph containing at least on vertex from each group.

## 18.2 Group Steiner Tree Problem

The group Steiner tree problem was introduced by Reich and Widmayer [1]. It arises in wire routing with multi-port terminals in physical VLSI design. The problem can be stated as follows: given an undirected gragh $G = (V, E)$ with the cost function $c : E \rightarrow \mathbb{R}^+$, and subsets of vertices, which we call *groups*, $g_1, g_2, ...g_k \in V$. The objective is to find the minimum cost subtree $T$ of $G$ that contains at least on vertex from each group $g_i$. Formally, find a conected subgraph $T = (V', E')$ that minimizes $\sum_{e \in E'} c_e$ such that $V' \cap g_i \neq \phi$ for all $i \in 1, ..., k$. We use $n$ to denote $|V|$, $m$ to denote $|E|$, and $N$ to denote the size of the largest group, $N = max_i |g_i| \leq n$.

We assume that groups are pairwise disjoint. This valid through the following transformation: if a vertex $v$ occurs in $p$ groups, $p \geq 1$, attach $p$ new vertices to $v$ with zero-cost edges. Each leaf of this star is assigned to one of the groups while $v$ does not belong to any group.

We also assume, without loss of generality, that $C$ is metric, that is, $\forall i, j, k, C_{ij} \leq C_{ik} + C_{kj}$. The assumption holds because in any given graph, if $C_{ij}$ is greater than the shortest path from $i$ to $j$, any algorithm solving the problem will substitute the edge $(i, j)$ with its shortest path to decrease the weight of its solution. The last assumption allows us to take the metric completion of the graph.

We now present some facts about the group Steiner tree problem:

**Fact 18.2.1** *The group Steiner tree problem is a generalization of the Steiner tree problem. Therefore, it is NP hard.*

**Fact 18.2.2** *The group Steiner tree problem is equivalent the set cover problem, thus, cannot be approximated to a fact* $o(\ln k)$ *unless* $P = NP$.

The above fact can be explained as follows. Given a collection of weighted subsets of a given ground set, the set cover problem consists of finding a minimum-weight sub-collection whose union is the entire set. To reduce this problem to a group Steiner tree problem, we build a star with a lef for each set. Every element in the set cover problem defines a group of leaves in the star in a natural way, namely, the leaves corresponding to the sets that contain this element. The equivalence is completed by giving the edges the weights of the corresponding sets.

The idea of using randomized rounding to solve the group Steiner tree problem was introduced by the paper of Garg, Konjevod and Ravi [2]. In this paper, the authors presented a randomized algorithm that solves the problem on trees with an $O(\log k \log N)$ approximation ratio. For the case of tree metrics, the algorithm first solves a linear programming relaxation of the group Steiner tree problem. Then, an extension of randomized rounding is employed to get the solution subtree. The bound on the cost of the tree follows from the rounding process. On the other hand, the paper showed that the solution tree actually covers all groups with reasonable probability using Janson's inequality [3](In this lecture, we use the inclusion-exclusion principle to prove this result). The following sections present the components of the randomized algorithm.

## 18.3 Linear Programming Formualtion of the Problem

We consider the group Steiner tree problem on a tree $T'' = (V.E)$ with nonnegative costs $c$ on its edges. We study the rooted version where a pre-specified root vertex $r$ is required to be in the solution subtree. To solve the unrooted version, we can run through the different vertices in the smallest group as the choice for the root $r$, and pick the best solution among these runs. For any subset of vertices $S \subseteq V$, let $\delta(S)$ denote the set of edges with exactly one endpoint in $S$. The ILP of the group Steiner tree problem is as follows:

Minimize $\sum_{e \in E} c_e x_e$

$\sum_{e \in \delta(E)} x_e \geq 1$, for all $S \subseteq V$ such that $r \in S$ and $S \cap g_i = \phi$ for some $i$

$x_e \in \{0, 1\}$

The number of constraints in this program is exponential. However, it can be solved in polynomial time based on the fact that a separation oracle can be constructed using a minimum cut procedure.

**Claim 18.3.1** *The ILP program for the group Steiner tree problem is feasible.*

**Proof:** We give the proof by contradiction. Assuming the solution of the ILP does not get a Steiner tree, then, there exits a group $g_i$ such that none of the elements of such group is connected to the $cc(root)$. This contradicts with the first constraint of the ILP ■

The ILP program can be relaxed by substituting the second constraint by the following one:
$0 \geq x_e \leq 1, \forall e \in E$

Using the max-flow min-cut theorem, we can interpret the constraints of the relaxed LP program as a requirement that any cut separating the root from all vertices of a given group must have capacity of at least one. We can think of adding a new source vertex for this group with edges to all vertices in it of infinite capacity and interpret the value $x_e$ as the capacity of the edge $e$.

The LP constraints, together with the max-flow min-cut theorem, imply that any solution $x$ must support a flow of at least one unit from this source to the root. In other words, the installed capacity $x$ is sufficient to support a total flow of value at least one from the vertices of any group to the root.

Let $x$ be the optimal LP solution, and $T'$ the support of $x$. Since $T''$ is a tree, $T'$ is a tree as well. We denot $z*$ the optimal value of the objective function.

## 18.4 Random Experiment

This section explains the randomized rounding process. It is assumed, without loss of generality, that all group vertices are leaves of $T'$. Thus, internal group vertices can be made leaves by inserting a zero cost edge.

**Fact 18.4.1** *For every edge $e \in E(T')$, $x_e \leq x_f$, where $f$ is the parent edge of $e$.*

This follows from the fact that $T'$ is a tree. If $e$ had a higher value than its parent in the optimal solution, then, the cut could have been simply pushed to the parent of $e$.

Assuming a fictitious edge above $r$ with flow of 1. Also, assuming that we damp all the probabilities of the leaf edges, i.e. those edges incident on leaf nodes, by $\Delta = \theta(\log N)$, we consider the following random experiment.

Repeat for $O(\log N \log k)$ times
1. Pick an edge e with probability $\frac{x_e}{x_f}$. If $e$ is incident on $r$, we include it with probability $x_e$, while a leaf edge $g$ is picked with a probability $\frac{x_g}{\Delta}$.
2. Retain all chosen edges that are incident on the root $r$.
3. Consider the connected component of the root as the new root $r'$.

We think of the iterations of the above program as phases of the random experiment. Let $T$ denote the resulting tree.

**Claim 18.4.2** *For all phases, the expected cost per phase is less than or equal to $\frac{z*}{\Delta}$.*

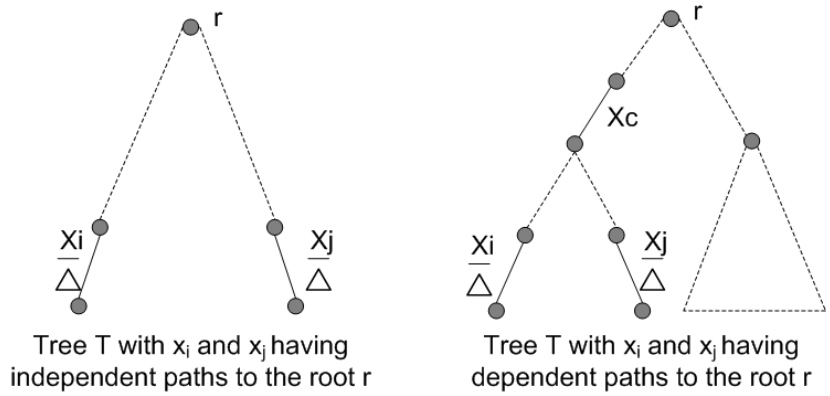
**Proof:** For any edge $e$, prob($e$ is chosen) $= \frac{x_e}{\Delta . x_f} \cdot \frac{x_f}{x_{parent(f)}} \ldots x_t$, where $t$ is the edge incident on $r$ falling on the path from $e$ to $r$. This is equal to $\frac{x_e}{\Delta}$. By linearity of expectation, the total expected cost per phase is equal to $\sum_{e \in E(T)} Prob(e \in T).c_e$ which is less than $\frac{z*}{\Delta}$. ■

**Theorem 18.4.3** *If we run the random experiment on a feasible solution of the original LP, then for every group $g$, the probability of including a vertex from $g$ in $T$ in one of the phases of the experiment is $\Omega(1/\log N)$. (Recall that $N$ is the maximum size of a group)*

**Proof:** Let $B_i$ be the event that $x_i$ was chosen at any phase. Applying the Inclusion-Exclusion principle to the probability that g is covered in a phase, we get the following:
$Prob(g\ covered) = Prob(\cup_i B_i) \geq \sum_i Prob(B_i) - \sum_{i,j} Prob(B_i \cap B_j)$

In order to get the lower bound of the LHS of the above inequality, we upper bound the RHS of the inequality. We start with the term $\sum_i Prob(B_i)$. Recall from the proof of claim 18.4.2, Prob(any leaf edge $e$ is chosen) $= \frac{x_e}{\Delta}$. Thus, $\sum_i Prob(B_i) = \sum_i \frac{x_i}{\Delta}$. From the LP constraint, $\sum_i x_e \geq 1$, Thus, $\sum_i Prob(B_i) \geq \frac{1}{\Delta}$.

Moving to the second term on the RHS of the inequality, $\sum_{i,j} Prob(B_i \cap B_j)$, this event can have two different values, depending on whether the events $B_i$ and $B_j$ are independent from, or dependent on, each other. Note that the two events can be dependent from each other if, and only if, the edges $x_i$ and $x_j$ have common paths to the root $r$. Figure 18.4 illustrates the two possibilities. We will address each of these two cases in turn.




In case $B_i$ is independent from $B_j$, this means that the path from $i$ to $r$ is not intersecting with the path from $j$ to $r$. In such case, $Prob(B_i \cap B_j) = Prob(B_i) \cdot Prob(B_j) = \frac{x_i}{\Delta} \cdot \frac{x_j}{\Delta}$. Thus, based on the LP constraint, $\sum_{i,j} Prob(B_i \cap B_j) \geq \frac{1}{\Delta^2}$.

In case the two edges $x_i$ and $x_j$ have common paths to the root, the two events, $B_i$ and $B_j$ will be dependent on each other. Thus, $Prob(B_i \cap B_j) = Prob(B_i) \cdot Prob(B_j|B_i) = \frac{x_i}{\Delta} \cdot \frac{x_j/\Delta}{x_c}$, where $c$ is the least common ancestor edge of both $i$ and $j$. Let $i \sim j$ means $i$ is dependent on $j$. Therefore, $\sum_{i \sim j} Prob(B_i \cap B_j) = \sum_{i \sim j} \frac{x_i}{\Delta} \cdot \frac{x_j/\Delta}{x_c}$. Again, based on the LP constarint, $\sum_{i \sim j} Prob(B_i \cap B_j) = \sum_{i \sim j} \frac{x_i \cdot x_j}{x_c \cdot \Delta^2} \geq \frac{1}{\Delta^2}$. Thus, both cases can be lower bounded by $\frac{1}{\Delta^2}$.

Gathering the two parts of the inequality, we can lower bound the $Prob(g\ covered)$ as follows:
$Prob(g\ covered) = Prob(\cup_i B_i) \geq \sum_i Prob(B_i) - \sum_{i,j} Prob(B_i \cap B_j)$.
$Prob(g\ covered) = \Omega(\frac{1}{\Delta}) - \Omega(\frac{1}{\Delta^2}) = \Omega(\frac{1}{\Delta})$. ■

Based on theorem 18.4.3, as well as the fact that the random experiment iterates for a number of times which upper bounds the probability that a group is covered in one phase, the following theorem holds.

**Theorem 18.4.4** *There is a randomized polynomial algorithm that finds a group Steiner tree on an underlying graph, which is a tree, of cost no more than* $O(\log N \log k)$ *times the optimum, where* $N$ *is the maximum size of a group and* $k$ *is the number of groups.*

## 18.5 Conclusions

In this lecture, we presented a polylogarithmic approximation algorithm for the group Steiner tree problem on a tree. Using Bartal's algorithm for embedding into metric spaces [4], an algorithm for general graphs is presented in [2].

## References

[1] G. Reich and P. Widmayer. Beyond Steiner's problem: A VLSI oriented generalization. In *Graph-Theoretic Concepts in Computer Science WG89,* volume 411 of *Lecture Notes in Com-*

*puter Science*, pages 196-210. 1990.

[2] N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group Steiner tree problem. In *Journal of Algorithms,* volume 37, pages 66-84. 2000.

[3] S. Janson. Poisson Approximations for large deviations. In *Random Structures and Applications,* volume 1, pages 221-230. 1990.

[4] Y. Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *FOCS,* pages 184-193, 1996.