14.1 Introduction

Today we're going to finish off the $O(\log n \log k)$-approximation to Group Steiner Tree on Trees that we started last class, and then introduce a new concept of *tree embeddings* and show how to use them to reduce GST to GST on trees.

14.2 Review from Last Class

Let's remember the GST on Trees problem, and the algorithm and where we were in the analysis.

- **Input:**
  - Tree $G = (V, E)$
  - Edge costs $c : E \to \mathbb{R}_{\geq 0}$
  - Root vertex $r \in V$
  - K groups $g_1, g_2, \ldots, g_k$, where each $g_i \subseteq V$
- **Feasible:** Tree $T \subseteq E$ such that $\forall i \in [k]$, $\exists v \in g_i$ such that T has a path between r and v.
- **Objective:** $\min \sum_{e \in T} c(e)$

Theorem 14.2.1 [Garg, Konjevod, Ravi 2000] There exists an $O(\log n \log k)$-approximation to GST on trees.

14.2.1 Linear Programming Relaxation

Consider the following LP relaxation, which basically requires that for every cut which separates r from some group, at least one edge crosses the cut:

$$\begin{aligned} \text{minimize:} \quad & \sum_{e \in E} c_e \cdot x_e & \textbf{(GST-LP)} \\ \text{subject to:} \quad & \sum_{e \in (S, \bar{S})} x_e \geq 1 \quad \forall i \in [k],\ \forall S \subseteq V \text{ such that } r \in S,\ g_i \cap S = \emptyset \\ & 0 \leq x_e \leq 1 \quad \forall e \in E \end{aligned}$$

14.2.2 Rounding Algorithm

Lemma 14.2.2 *Let $e \in E$, $p(e)$ be the parent edge of e (remember that G is a tree). Then in any optimal $\vec{x}$, $x_{p(e)} \geq x_e$.*

Algorithm 1 GKR Rounding Algorithm for GST

for each $e \in E$ **do**
  Mark e with probability $\frac{x_e}{x_{p(e)}}$. If e is incident on r, then mark e with probability x_e.
end for
Include e if e and all its ancestors are marked.
return T

Consider the following rounding algorithm:

Lemma 14.2.3 $\mathbf{Pr}[\textit{include } e] = x_e$ *for all* $e \in E$.

Corollary 14.2.4 $\mathbf{E}[ALG] \leq LP$.

Claim 14.2.5 *Using GKR rounding,* $\forall i \in [k]$,

$$\mathbf{Pr}[g_i \textit{ connected to } r] \geq \frac{1}{O(\log |g_i|)} \geq \frac{1}{O(\log n)}.$$

We showed how to prove Theorem 14.2.1 by assuming **Claim 14.2.5**, and were in the middle of trying to prove Claim 14.2.5. Let $g = g_i$ be the group that we're trying to prove the claim for.

Definition 14.2.6 *Let FAIL be the event that* g *is not connected to* r.

Lemma 14.2.7 *If* $x'_e \leq x_e$ *for all* $e \in E$*, then*

$$\mathbf{Pr}\left[FAIL \textit{ using } x'\right] \geq \mathbf{Pr}[FAIL \textit{ using } x]$$

Proof: We prove this by considering one edge at a time and then using induction. So suppose that the only difference between x and x' is on one edge $e = \{v, u\}$ (with $v = p(u)$) where $x'_e < x_e$. Note that the probability of connecting every vertex outside of the subtree T rooted at u is exactly the same in x and in x', so we'll only need to worry about the subtree that goes through e.

Suppose for simplicity that there are two children edges $e_1 = \{u, w\}$ and $e_2 = \{u, z\}$ of u (the higher degree case has more complex math but everything works out basically the same). Let T_1 be the subtree rooted at the child node of e_1, and let T_2 be the subtree rooted at the child node of e_2. Using the fractional solution x, let

$$p_1 = \mathbf{Pr}[\text{fail to connect } T_1 \cap g \text{ to } w],$$
$$p_2 = \mathbf{Pr}[\text{fail to connect } T_2 \cap g \text{ to } z].$$

Then

$$\mathbf{Pr}[\text{fail to connect } T \cap g \text{ to } v] = (1 - x_e) + x_e \left(\left(1 - \frac{x_{e_1}}{x_e}\right) + \frac{x_{e_1}}{x_e} p_1\right)\left(\left(1 - \frac{x_{e_2}}{x_e}\right) + \frac{x_{e_2}}{x_e} p_2\right).$$

Here the first term is the probability that e is not picked, and then if e is picked we take the product of failing to connect T_1 to u and failing to connect T_2 to u (since to fail on T we would have to fail on both and they're independent). Each of these terms can be broken into the probability of not marking e_1 (or e_2) plus the probability of picking e_1 (or e_2) times to probability of failing in the tree below that (p_1 or p_2 respectively).

When we simplify this expression, we get that

$$\mathbf{Pr}[\text{fail to connect } T \cap g \text{ to } r] = 1 - x_{e_1}(1-p_1) - x_{e_2}(1-p_2) + \frac{x_{e_1}x_{e_2}(1-p_1)(1-p_2)}{x_e}.$$

This expression clearly increases as x_e decreases. Thus we are less likely to fail using x than we are using x'. ■

This lemma means that decreasing x values can't help us, so if $x'_e \leq x_e$ for all $e \in E$ and probability of connecting g to r using x' is at least $\frac{1}{\log|g|}$, then the same is true using x (which is what we are trying to prove).

Now consider the following construction of x'.

1) Remove all leaves not in g and all unnecessary edges.

2) Reduce x values until minimally feasible (exactly one unit of flow is sent to g, or equivalently the min $r-g$ cut is equal to 1).

3) Round down to the next power of 2; now the flow is at least $\frac{1}{2}$ because all edges will be at least half of their original value (equivalently, the minimum $r-g$ cut is at least 1/2).

4) Delete all edges with $x_e \leq \frac{1}{4|g|}$. Since there are at most $|g|$ leaves, the flow is still at least

$$\frac{1}{2} - |g| \cdot \frac{1}{4|g|} = \frac{1}{4}.$$

5) If $x_e = x_{p(e)}$, then contract e (since our rounding will mark e with probability 1 anyway).

x' is the values that we get after these modifications.

Lemma 14.2.8 *The height of the tree is at most* $O(\log|g|)$

14.3 Finishing the Proof via Janson's Inequality

Now let's do the new stuff. We now want to show that if we round using x', the probability we connect g to r is at least $\frac{1}{\log|g|}$. For $v \in G$, let's abuse notation and let $x_v = x_e$ where e is the edge leading to v (recall that WLOG v is a leaf). Note that the *expected* number of terminals of g which are connected is

$$\sum_{v \in g} \mathbf{Pr}[v \text{ connected to r}] = \sum_{v \in g} x'_v \geq 1/4,$$

since as discussed in x we are still able to send 1/4 total flow to g. If we had concentration then we would be basically be done: it would be unlikely for us to be below the expectation, but the only integer below the expectation is 0, so it would be unlikely for us to get 0, so we would likely get 1. Unfortunately, we can't apply Chernoff since these events are not independent.

It turns out that we'll need a new tool: *Janson's Inequality.* Before stating it formally, we need to set up some notation

- Let S be a ground set of items.
- Let $P_1, \ldots, P_k$ be subsets of S.
- Let $p_e \in [0, 1]$ for each $e \in S$.
- Let S' be the set obtained by adding each $e \in S$ with probability p_e.
- Let $\mathcal{E}_i$ be the event that $P_i \subseteq S'$.
- Let $\mu = \sum_{i=1}^{k} \mathbf{Pr}[\mathcal{E}_i]$ and $\Delta = \sum_{i \sim j} \mathbf{Pr}[\mathcal{E}_i \cap \mathcal{E}_j]$ where $i \sim j$ if $P_i \cap P_j \neq \emptyset$ (i.e., if $\mathcal{E}_i$ and $\mathcal{E}_j$ are dependent).

Theorem 14.3.1 (Janson's inequality) *If $\mu \leq \Delta$, then the probability that* none *of the events happen is*

$$\boldsymbol{Pr}\left[\bigcap_i \overline{\mathcal{E}_i}\right] \leq e^{-\frac{\mu^2}{2\Delta}}.$$

To apply Janson's inequality to the GST setting,

- $S = E$.
- P_i = path from r to $v_i \in g$.
- S' is the set of marked edges, so $p_e = x_e / x_{p(e)}$
- For $v_i \in g$, the event $\mathcal{E}_i$ is the event that all edges on the $v_i - r$ path are marked, i.e., the event that v_i is connected to r.

Claim 14.3.2

$$1 \geq \mu = \sum_i P[\mathcal{E}_i] \geq \frac{1}{4}.$$

Proof: For each $v_i \in g$, the probability of $\mathcal{E}_i$ is, by Lemma 14.2.3, the x value of the edge incident on v_i. This is exactly the amount of flow sent to v_i. Since at least 1/4 flow is sent in total to vertices in g, $\sum_i P[\mathcal{E}_i] \geq 1/4$. On the other hand, since we made x' minimal, the total flow to g is at most 1. ■

Claim 14.3.3

$$\Delta = O(\log |g|).$$

Proof: Let $H = O(\log |g|)$ be the height of the tree. For $u \in g$, let $\Delta_u = \sum_{v \in g: v \sim u} \mathbf{Pr}[\mathcal{E}_u \cap \mathcal{E}_v]$ (so $\Delta = \sum_{u \in g} \Delta_u$). For $v \in g$ with $v \sim u$, let e be the lowest edge shared by $r - u$ and $r - w$ paths, and let $c(e)$ be the child node of e. Then

$$\mathbf{Pr}[\mathcal{E}_v | \mathcal{E}_u] = \frac{x'_v}{x'_{p(v)}} \cdot \frac{x'_{p(v)}}{x'_{p(p(v))}} \cdots \frac{x'_{c(e)}}{x'_e} = \frac{x'_v}{x'_e},$$

where we have slightly abuse notation to identify a vertex with the edge to its parent. Thus we get that

$$\mathbf{Pr}[\mathcal{E}_v \cap \mathcal{E}_u] = \frac{x'_u x'_v}{x'_e}.$$

Now let $F(e) = \{v \in g : \text{lowest edge on } r - u \text{ path and } r - v \text{ path is } e\}$. Then

$$\sum_{v \in F(e)} \mathbf{Pr}[\mathcal{E}_u \cap \mathcal{E}_v] = \frac{x'_u}{x'_e} \sum_{v \in F(e)} x'_v \leq \frac{x'_u}{x'_e} x'_e = x'_u,$$

where we've use the flow constraints in the inequality. This let's us bound Δ_u:

$$\Delta_u = \sum_{e \in r - u \text{ path}} \sum_{v \in F(e)} \mathbf{Pr}[\mathcal{E}_v \cap \mathcal{E}_u] \leq \sum_{e \in r - u \text{ path}} x'_u \leq H x'_u.$$

And we can now final bound Δ:

$$\Delta = \sum_{u \in g} \Delta_u \leq H \sum_{u \in g} x'_u \leq H,$$

as claimed. ■

By plugging μ and Δ from the claims into Janson's inequality, we get that

$$\mathbf{Pr}[\text{success}] = 1 - \mathbf{Pr}\left[\bigcap_i \bar{\mathcal{E}}_i\right] \geq 1 - e^{-\frac{1}{O(\log |g|)}} \geq \frac{\frac{1}{O(\log |g|)}}{1 + \frac{1}{O(\log |g|)}} = \frac{1}{O(\log |g|)},$$

where in addition to Janson's inequality we've used the fact that $1 - e^{-x} \geq \frac{x}{x+1}$ for all $x \geq -1$. This proves **Claim 14.2.5**, which in turn implies the claimed approximation ratio of $O(\log n \log k)$.

14.4 Tree Embeddings

So now we have an $O(\log n \log k)$-approximation for Group Steiner Tree as long as the input is a tree. This was a highly nontrivial algorithm and analysis – how can we possibly hope to extend it all the way to general graphs? We're going to do this by using a technique called *metric embeddings*: we're going to *embed* general metric spaces into trees.

Recall the definition of a metric space:

Definition 14.4.1 *A pair (V, d) is a metric space if for all $u, v, w \in V$:*

1. $d(u,v) = 0 \iff u = v$
2. $d(u,v) = d(v,u)$
3. $d(u,v) \le d(u,w) + d(w,v)$

Note that it is common to simply refer to the metric as d instead of the pair (V,d). We're going to be concerned with a special type of metric known as a *tree metric*.

Definition 14.4.2 *A* tree metric (V', T) *for a set of nodes V is a tree T on vertices V', where $V \subseteq V'$ are the leaves of T. Every edge of T has a nonnegative length.*

The distance in T between any two vertices $u, v \in V'$ is denoted $d_T(u,v)$, where the distance in T is the length of the unique $u - v$ path in T.

Definition 14.4.3 *Let (V,d) be a metric and (V',T) a tree metric for V. Then (V,d) embeds into T with distortion α if $d(u,v) \le d_T(u,v) \le \alpha \cdot d(u,v)$ for all $u, v \in V$.*

Intuitively, if we can embed (V,d) into some tree (V',T) with small distortion, then T is "like" the original metric space so we might hope that we can just solve any problem that we care about it on T instead of on the original metric. Unfortunately, this is not always possible: even simple metric spaces like the cycle C_n might require large distortion to embed into any tree. This is trivial to see if we required T to be a subtree of the input graph, but since we're not requiring that, this is a bit harder to prove. It is possible to show that C_n requires distortion at least $\frac{n-1}{8}$ to embed into any tree.

What can we do? Let's take inspiration from the cycle: there's no tree which allows small distortion, but if we fix some pair $u, v \in V$, then a *random* subtree of C_n is pretty good in expectation! For example, if u and v are adjacent in C_n, then with probability $1/n$ they get distance n, while with probability $\frac{n-1}{n}$ they're still at distance 1. So the expected distance is at most 2. So for any pair of nodes the expected distortion is small, even though once we instantiate some particular tree, there *will be* some pair which is badly distorted. As it turns out, though this kind of expected distortion is enough for many applications.

The best and provably optimal result for doing this is due to Fakcharoenphol, Rao, and Talwar, who proved the following theorem.

Theorem 14.4.4 *Let (V,d) be a metric. Then there is a randomized, polytime algorithm that produces a tree metric (V',T) for V such that*

1. *$d(u,v) \le d_T(u,v)$ for all $u, v \in V$, and*
2. *$\mathbf{E}[d_T(u,v)] \le O(\log n) \cdot d(u,v)$ for all $u, v \in V$.*

In other words, this theorem gives an embedding into a *distribution* of *dominating* trees (a distribution of trees each of which does not contract any pair). This algorithm is tight: there are metrics for which any embedding into a distribution of dominating trees requires distortion $\Omega(\log n)$.

We're going to spend the next couple of classes proving this theorem and analyzing tree embeddings, but before we do that, let's show why they're useful. It's not hard to see that almost any problem which involves distances can be turned into a problem on trees by using this theorem and losing an extra $O(\log n)$ in the approximation ratio, but let's see this for a particular problem: Group Steiner Tree.

14.5 Group Steiner Tree on General Metrics

Recall the GST problem:

- **Input**:, A graph $G = (V, E)$, edge costs $c : E \to \mathbb{R}_{\geq 0}$, a root vertex $r \in V$, and groups $g_1, \ldots, g_k \subseteq V$.
- **Feasible solution**: A tree T such that for all $i \in [k]$, there is some $v \in g_i$ such that T has a path between r and v.
- **Objective**: $\min \sum_{e \in T} c(e)$

We now know that Garg, Konjevod, and Ravi (GKR) gave an $O(\log n \log k)$-approximation when the input graph is a tree, and that the problem is $\Omega(\log^{2-\epsilon} n)$-hard to approximate even on trees. How can we design an approximation algorithm for general metrics? Use FRT to change the input into a tree!

Slightly more formally, consider the following algorithm:

1. Extend c to a metric space (V, d) where $d(u, v)$ is the minimum cost of any $u - v$ path.
2. Use FRT (Theorem 14.4.4) to embed (V, d) into a tree (V', T) with distortion $O(\log n)$. Note that since V are the leaves of T, all of the terminals (vertices in the groups) are now leaves.
3. Make a new group which is just $\{r\}$, and then use the GKR algorithm to get a subtree T' of T which is an $O(\log |V'| \log k)$-approximation to the optimal solution on T.
4. Shortcut T' to get a cycle C only on terminals.
5. Use C (with one arbitrary edge removed) as our solution in the metric space (V, d). To get a solution on G, replacing any edge of C which doesn't exist in G with a path of the same length.

This algorithm clearly gives a feasible solution: GKR returns a tree which connects at least one terminal from each group (including r) to the root of T, so C has r and at least one terminal from each group. Thus C gives a feasible solution. The algorithm also clearly takes only polynomial time. So we just need to analyze the approximation ratio.

Theorem 14.5.1 *This algorithm is a $O(\log^2 n \log k)$-approximation, i.e.,*

$$\mathbf{E}[c(C)] \leq O(\log^2 n \log k) \cdot OPT.$$

Proof: Let's set up some notation.

- Let S be the terminals connected by OPT (so $S \cap g_i \neq \emptyset$ for all $i \in [k]$)
- Let C_S be the cycle on S obtained by shortcutting OPT (so $c(C_S) \leq 2 \cdot OPT$).
- T will be the (random) tree built by FRT, with costs c_T or d_T.
- Let $OPT(T)$ denote the optimal solution in T.
- Let T_S be the subtree of T induced by S (i.e., the subtree of T which consists of all the paths from nodes in S up to the LCA of S.

Now we can actually prove the theorem.

$$
\begin{aligned}
\mathbf{E}[c(C)] &\leq \mathbf{E}[c_T(C)] && \text{distances in } T \text{ are nondecreasing} \\
&\leq \mathbf{E}\big[2 \cdot c_T(T')\big] && \text{shortcutting costs at most a factor of 2} \\
&= 2 \cdot \mathbf{E}\big[c_T(T')\big] && \text{linearity of expectation} \\
&\leq 2\mathbf{E}[O(\log n \log k) \cdot c_T(OPT(T))] && \text{GKR} \\
&= O(\log n \log k) \cdot \mathbf{E}[c_T(OPT(T))] && \text{linearity of expectations} \\
&\leq O(\log n \log k) \cdot \mathbf{E}[c_T(T_S)] && \text{by definition of } OPT(T) \\
&\leq O(\log n \log k) \cdot \mathbf{E}[c_T(C_S)] && C_S \text{ a cycle on leaves of } T_S \\
&= O(\log n \log k) \cdot \mathbf{E}\left[\sum_{(u,v) \in C_S} d_T(u,v)\right] && \text{by definition} \\
&= O(\log n \log k) \cdot \sum_{(u,v) \in C_S} \mathbf{E}[d_T(u,v)] && \text{linearity of expectations} \\
&\leq O(\log n \log k) \cdot \sum_{(u,v) \in C_S} (O(\log n) \cdot d(u,v)) && \text{FRT} \\
&= O(\log^2 n \log k) \cdot \sum_{(u,v) \in C_S} d(u,v) && \text{linearity of expectations} \\
&\leq O(\log^2 n \log k) \cdot 2 \cdot OPT && \text{shortcutting} \\
&= O(\log^2 n \cdot \log k) \cdot OPT && \text{asymptotic notation}
\end{aligned}
$$

■

So combining FRT with GKR gives an $O(\log^2 n \log k)$-approximation to GST in general! This is still the state of the art. The question of whether this extra $\log n$ loss can be avoided is still an extremely important open question in approximation algorithms.

14.6 Metric Embeddings in General

We're not going to talk too much about general metric embeddings, but our approach for GST can be generalized to many other problems and other metrics. Let's see this a bit abstractly.

Definition 14.6.1 *(V, d) embeds into (V, d') with distortion α is $d(u, v) \le d'(u, v) \le \alpha d(u, v)$ for all $u, v \in V$.*

There are equivalent definitions based on contraction rather than expansion or on both, which are slightly more natural in some contexts, but this definition is more intuitive based on what we've been doing.

Now suppose that we have a β-approximation for some problem in d', but not in d. Then consider the algorithm which first embeds d into d' with distortion α, and then uses the β-approximation for d'. If the problem that we care about has costs which are just sums of distances (like many of the problems we've been thinking about), then we get that

$$\begin{aligned} c(ALG) = \sum_{\{u,v\} \in ALG} d(u, v) \le \sum_{\{u,v\} \in ALG} d'(u, v) \le \beta \sum_{\{u,v\} \in OPT(d')} d'(u, v) \le \beta \sum_{\{u,v\} \in OPT} d'(u, v) \\ \le \beta\alpha \sum_{\{u,v\} \in OPT} d(u, v) = \beta\alpha \cdot c(OPT) \end{aligned}$$

Handling probabilistic embeddings, like we did with FRT for GST, just involves putting expectations in the right places, but it all works out the same. So as long as our problem is "about" distances, we can use metric embeddings to transform the input metric into a "simpler" metric (like a tree) by paying the distortion in the approximation ratio.