600.469 / 600.669 Approximation Algorithms **Lecturer:** Michael Dinitz
Topic: Randomized Rounding: Group Steiner Tree **Date:** 3/24/15
Scribe: David Gong

13.1 Group Steiner Tree (GST)

Input:
- Graph $G = (V, E)$
- Edge costs $c_e \geq 0$, $e \in E$
- Root vertex $r \in V$
- K groups $g_1, g_2, \ldots, g_k$, where each $g_i \subseteq V$

Feasible: Tree T 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 13.1.1 *GST contains set cover.*

Proof of Theorem 13.1.1: Let $(U, \mathcal{S})$ be a set cover instance. Then construct a star with

- Leaf for each $S \in \mathcal{S}$
- Group g_e for each $e \in U$ where $g_e = \{S \in \mathcal{S} \mid e \in S\}$.

Consider a set cover $S_1, \ldots, S_k$. Then $S_1, \ldots, S_k$ is a GST solution. Conversly, consider a GST solution $S_1, \ldots, S_k$. Then $S_1, \ldots, S_k$ is a set cover. ■

Theorem 13.1.2 *It is NP-hard to approximate GST better than $\Omega(\log n)$-hard to approximate GST.*

Theorem 13.1.3 [Halperin, Krauthgamer, 2003] $\forall \epsilon > 0$, GST is $\Omega(\log^{2-\epsilon} n)$-hard to approximate.

Assumptions:

- G is a tree.
- If $v \in g_i$ for any i, then v is a leaf.

Theorem 13.1.4 [Garg, Konjevod, Ravi] There exists an $O(\log n \log k)$-approximation to GST on trees.

13.1.1 A Linear Program for GST

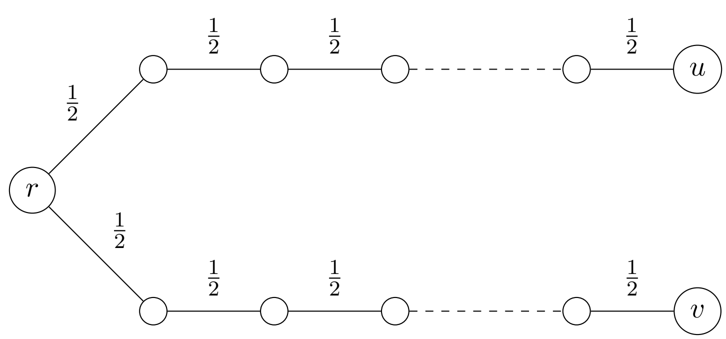
$$\begin{aligned}
\text{minimize:} \quad & \sum_{e \in E} c_e \cdot x_e && \textbf{(GST-LP)} \\
\text{subject to:} \quad & \sum_{e \in (S, \bar{S})} x_e \geq 1 \quad \forall i \in [k],\ \forall S \subseteq V \text{ such that } r \in S,\ g_i \cap S = \emptyset \\
& 0 \leq x_e \leq 1 \quad \forall e \in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

Notice that there are exponential number of constraints. The following method of separation resolves this:

- For each $i \in [k]$, add terminal t_i adjacent to all nodes in g_i with edges of value 1.
- Compute the minimum $r - t_i$ cut by using max-flow min-cut.
- If the minimum cut is less than 1, then the violated constraint has been found.
- Otherwise there are no violated constraints.

It is also not hard to see based on max-flow min-cut that this is equivalent to an LP which requires us to send one unit of flow from each r to to t_i (the "fake" terminal adjacent to all of g_i). Then x_e variables then are interpreted as capacities. We will make use of this flow-based interpretation later.

Independent randomized rounding is not appropriate in this problem. Consider the following tree



where there are $\frac{n}{2} - 1$ nodes on both the $r - u$ and $r - v$ paths (not counting r, u, or v). Suppose that $g_1 = \{u, v\}$. Then if we sample each edge independently with probability equal to its LP value,

$$P(\text{connect } u \text{ to } r) = \frac{1}{2^{\frac{n}{2}}}$$

Lemma 13.1.5 *Let $e \in E$, $p(e)$ be the parent edge of e (remember that G is a tree). Then in any optimal $\vec{x}$, $x_{p(e)} \geq x_e$.*

13.1.2 Rounding Algorithm

The rounding algorithm presented by **[GKR]** is as follows

Algorithm 1 GKR Rounding Algorithm for GST

for each x_e **do**
  For each edge e, independently 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

Lemma 13.1.6 *$P[include\ e] = x_e$.*

Proof of Lemma 13.1.6: Pick any an edge e and suppose e has i ancestors. Then

$$\begin{aligned} P[e \text{ included}] &= \frac{x_e}{x_{p(e)}} \cdot \frac{x_{p(e)}}{x_{p^2(e)}} \cdot \frac{x_{p^2(e)}}{x_{p^3(e)}} \cdot \dots \cdot \frac{x_{p^{i-1}(e)}}{x_{p^i(e)}} \cdot x_{p^i(e)} \\ &= x_e. \end{aligned}$$

■

Corollary 13.1.7 *$E(ALG) \leq LP$.*

Proof of Corollary 13.1.7:

$$\sum_{e \in E} c_e \cdot E[\mathbf{1}_{e \in ALG}] = \sum_{e \in E} c_e \cdot x_e = LP.$$

■

Claim 13.1.8 *Using GKR rounding, $\forall i \in [k]$,*

$$P[g_i \text{ connected to } r] \geq \frac{1}{\log |g_i|} \geq \frac{1}{\log n}.$$

We will first prove that by assuming **Claim 13.1.8**, we can acheive an $O(\log n \log k)$ approximation

Proof: First, suppose GKR rounding is run $O(\log n \log k)$ times. Now fix some g and notice that

$$\begin{aligned} P[g \text{ not connected to } r] &\leq \left(1 - \frac{1}{\log |g|}\right)^{O(\log n \log k)} \\ &\leq e^{-\log k} \\ &= \frac{1}{k}. \end{aligned}$$

Now for each $i \in [k]$, let P_i be the least expensive $r - g_i$ path. Then it is certainly true that $c(P_i) \leq OPT$. Now if g_i is not connected, then add P_i. Then notice that

$$\begin{aligned} E[\text{cost}] &\leq O(\log n \log k) \cdot OPT + \sum_{i=1}^{k} \frac{1}{k} \cdot OPT \\ &= O(\log n \log k) \cdot OPT. \end{aligned}$$

This is to say that adding the shortest paths to the disconnected groups does not significantly hurt us because the probability that a group is disconnected is small. ■

The rest of these notes will be aimed at setting up the proof of **Claim 13.1.8**. First we give a lemma that gives the general idea behind the proof. Let us fix some g, then

Definition 13.1.9 *Let FAIL be the event that g is not connected to r.*

Lemma 13.1.10 *If $x'_e \leq x_e \ \forall e \in E$, then*

$$P[FAIL \text{ } using \text{ } x'] \geq P[FAIL \text{ } using \text{ } x]$$

Now consider the following construction of x'.

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

2) Reduce x values until minimally feasible (exactly one unit of flow is sent to g).

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.

4) Delete all edges with $x_e \leq \frac{1}{4|g|}$; now the flow is 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 include e with probability 1 anyway).

Lemma 13.1.11 *The height of the tree is at most $O(\log |g|)$*

Proof of Lemma 13.1.11: At each level, x values go down by at least a factor of 2 since we rounded to powers of 2 an d contracted edges with the same value as their parent. Because of steps 2 and 4, we know that

$$\frac{1}{4|g|} \leq x_e \leq 1.$$

Hence the number of levels is at most $\log(4|g|) = O(\log |g|)$. ■

In order to continue with the introduction of Janson's inequality, we must first set up notation

- Let S be a ground set.
- Let $p_e \in [0,1]$ for each $e \in S$.
- Let $P_1, \dots, P_k$ be subsets of 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 $\mu = \sum_{i=1}^{k} P[\mathcal{E}_i]$ and $\Delta = \sum_{i \sim j} P[\mathcal{E}_i \cap \mathcal{E}_j]$ where $i \sim j$ if $P_i \cap P_j \neq \emptyset$.

Theorem 13.1.12 (Janson's inequality)

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

To apply Janson's inquality to the GST setting,

- $S = E$.
- $P_i =$ path from r to $v_i \in g$.
- $\mathcal{E}_i =$ event that g is connected to r using v_i.

Claim 13.1.13

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

Proof: For each $v_i \in G$, the probability of $\mathcal{E}_i$ is, by Lemma 13.1.6, 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, $\sum_i P[\mathcal{E}_i] \geq 1/4$. ∎

Claim 13.1.14

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

Proof: We did not have time to cover this in class. A proof can be found in the CMU notes linked to from the course schedule (scribed by Amitabh Basu, now a professor of AMS at JHU). ∎

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

$$P\left[\bigcap_i \bar{\mathcal{E}}_i\right] \leq e^{-\frac{1}{\log |g|}} \approx \left(1 - \frac{1}{\log |g|}\right)$$

so the probability of success is at least $\frac{1}{\log |g|}$. This proves **Claim 13.1.8** so the $O(\log n \log k)$ approximation is correct.

References

HK03 E. Halperin and R. Kruathgamer, Polylogarithmic Inapproximability. *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC)*, 585-594, 2003.

GKR00 N. Garg, G. Konjevod, and R. Ravi, A polylogarithmic approximation algorithm for the group Steiner tree problem, *SODA* 2000.