IOE 691: Approximation Algorithms | Date: 2/22/2017

Lecture Notes: Group Steiner Tree

Instructor: Viswanath Nagarajan | Scribe: Parker Koch

1 Problem Definition

Definition 1.1 *Let $G = (V, E)$ be a tree of depth H equipped with a root node $r \in V$ and an edge weighting $c : E \to \mathbb{R}_+$, and define groups $S_i \subset V$ for $i = 1, \ldots, k$. The group Steiner tree problem seeks a subtree $T \subseteq G$ of minimum total edge weight such that for each group S_i, at least one vertex $v \in S_i$ is connected to r in T.*

Note that the group Steiner tree problem is at least as hard as set cover: For any instance of set cover, we can construct an instance of group steiner tree for which solving the tree solves set cover. This is done by constructing a star graph with a vertex and group for each set and element in the set cover instance, respectively. Figure 1 shows an example of this conversion.

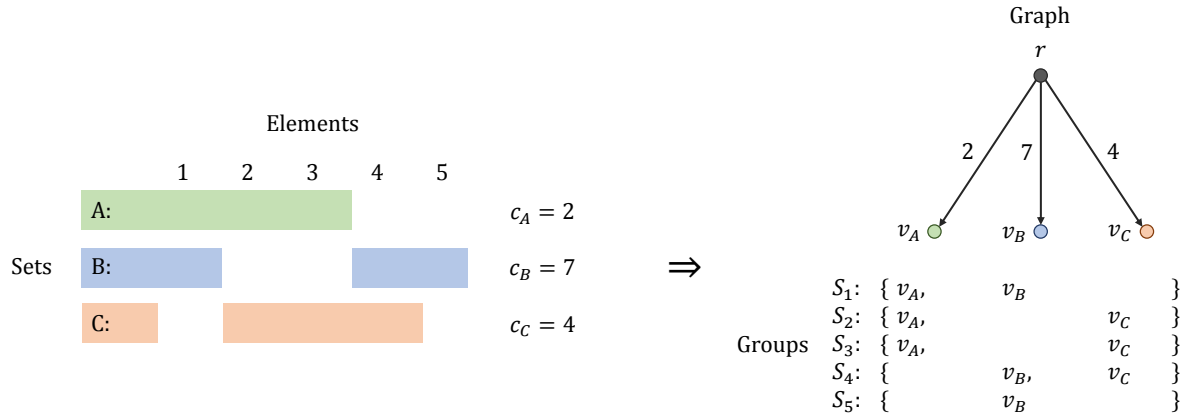

Figure 1: Converting an instance of set cover into an instance of group Steiner tree. The three sets become vertices, each contained in the groups corresponding to the elements those sets contain. Choosing vertices to connect the groups to r is equivalent to choosing the sets to cover the elements, at the same costs.

Observation 1.1 *Without loss of generality, we may assume that only leaf nodes belong to the sets S_i.*

Proof: If v belongs to S_i for some i and v has degree greater than one, replace v with a new vertex v' that belongs to none of the sets S_i, then connect v to v' with a zero-weight edge (disconnecting v from everything else). Figure 2 shows an example of this transformation. Any feasible solution in the original tree corresponds exactly to an equally costly, feasible solution in the new one, and vice versa. Thus, solving the problem on the new tree is equivalent. ■

Observation 1.2 *Without loss of generality, we may assume that each node belongs to at most one set.*

Proof: If v belongs to multiple sets, replace v with a new vertex v' that belongs to none of the sets, and for each set S_i that v belonged to, create a new vertex that belongs to that set and is connected only to v' by a zero-weight edge. As before, the instances are equivalent and an example is shown in Figure 2. ■

Figure 2: Examples of reductive equivalence transformations. On the left, a non-leaf vertex is pulled out to a leaf in order to ensure that all vertices belonging to sets are leaf vertices. On the right, a vertex contained in two sets is split into two new vertices to ensure that all vertices belong to at most one set. Both transformations preserve the optimal solution.

1.1 LP Relaxation

Let $p(e)$ be the parent edge of any edge $e \in E$, i.e., the last edge in the unique path from r to the tail of e, and let $p(v)$ be the parent edge of any vertex $v \in V$. The Group Steiner Tree problem can be written as an IP with variables x_e, equal to 1 if edge e is chosen to be part of the solution T and 0 otherwise. However, including only these can lead to LP solutions that round to IP solutions that are $O(n)$-bad. We can mitigate this by introducing flow variables $f_{e,i} \leq x_e$:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{e \in E} c_e x_e \\
\text{subject to} \quad & x_{p(e)} \geq x_e \qquad \forall e \in E \\
& f_{e,i} \leq x_e \qquad \forall e \in E, \forall 1 \leq i \leq k \\
& \sum_{e' : p(e') = e} f_{e',i} = f_{e,i} \qquad \forall e \in E \setminus \{p(v) : v \in S_i\}, \forall 1 \leq i \leq k \\
& \sum_{v \in S_i} f_{p(v),i} = 1 \qquad \forall 1 \leq i \leq k \\
& x_e, f_{e,i} \in \{0, 1\} \qquad \forall e \in E, \forall 1 \leq i \leq k
\end{aligned}
$$

In words, the constraints ensure 1) if an edge is selected, so is its parent; 2) we can only have flow along selected edges; 3) flow in matches flow out at all vertices except the root and vertices in S_i; 4) total flow into S_i is 1; and 5) the variables are integers.

In the LP, we replace $\{0, 1\}$ with $[0, 1]$.

Independent randomized rounding does not work for this problem. Consider the tree in Figure 3: if we choose each edge e with probability x_e, the probability of obtaining a feasible solution can be made arbitrarily small by increasing the height of the tree.

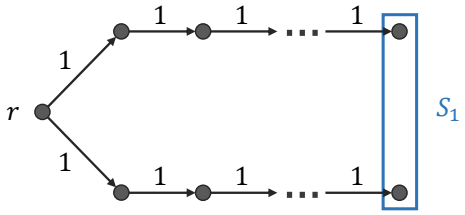

Figure 3: Pathological example of a tree on which independent randomized rounding of the LP-optimal x_e values yields feasible solutions with arbitrarily low probability. Say the tree has $2n + 1$ vertices; the optimal LP solution will have $x_e = 1/2$ for every $e \in E$, and choosing each edge independently with probability x_e will yield a feasible solution with probability at most $(1/2)^{n-1}$.

1.2 GKR Rounding & Analysis

Instead, we use **GKR rounding**, named for authors Garg, Konjevod, and Ravi [1]:

1. Choose each edge independently with probability $x_e/x_{p(e)}$, or just x_e if e is incident on the root;
2. Discard any disconnected edges.

Call the tree generated this way T, and for each group S_i, let $T_i = S_i \cap T$ be the vertices in T that belong to S_i. This tree has a useful property:

Lemma 1.1 *The expected cost of T is the optimal value of the LP, LP*.*

Proof: The probability that e belongs to T is easy to compute. Let $r \to e_1 \to \ldots \to e_q \to e$ be the path from r to e; e can only belong to T if they were all chosen in step 1, i.e.

$$\begin{aligned} Pr[e \in T] &= Pr[e_1 \text{ chosen}] \cdot Pr[e_2 \text{ chosen}] \cdots Pr[e_q \text{ chosen}] \cdot Pr[e \text{ chosen}] \\ &= x_{e_1} \cdot \frac{x_{e_2}}{x_{e_1}} \cdots \frac{x_{e_q}}{x_{e_{q-1}}} \cdot \frac{x_e}{x_{e_q}} \\ &= x_e \end{aligned}$$

and so:

$$\mathbb{E}[cost(T)] = \sum_{e \in E} c_e Pr[e \in T] = \sum_{e \in E} c_e x_e = \text{LP}^*$$

■

The following lemma is useful in further proofs:

Lemma 1.2 *In the optimal LP solution, for every edge $e \in E$, $x_e = \max_i f_{e,i}$. Futhermore, for $v \in S_i$, $x_{p(v)} = f_{p(v),i}$.*

Proof: The first is true by the optimality of the solution; if $x_e > f_{e,i}$ for all i, then we can decrease x_e without penalty, decreasing the objective in contradiction to the solution's optimality. The second follows from the fact that the flow in/flow out constraint reduces to $f_{p(v),j} = 0$ for any $v \in S_i$ and $j \neq i$. ■

We now begin work to prove that for each $i \in [1, k]$, $Pr[T \text{ connects } i] \geq \frac{1}{4H}$, where H is the height of G. The following "altered" version of T_i is useful in the proof:

$$\bar{T}_i = \begin{cases} T_i & \text{if } |T_i| \leq 2H \\ \emptyset & \text{otherwise} \end{cases}$$

Lemma 1.3 *For $v \in S_i$, $\mathbb{E}[|T_i| \,\big|\, v \in T_i] \leq H$.*

Proof: Let $e_1 \to e_2 \to \cdots \to e_q$ be the edges on the path from r to v. For each $j = 0, \ldots, q-1$, let $G_j \subset G$ be the subtree rooted at the head of e_j containing all descendents except those of e_{j+1}, with G_0 rooted at r (see Figure 4 for a diagram). We can break T_i into disjoint subsets $T_i \cap G_j$ and observe:

$$\mathbb{E}[|T_i \cap G_j| \,\big|\, v \in T_i] = \sum_{u \in S_i \cap G_j} Pr[u \in T_i \,\big|\, v \in T_i] = \sum_{u \in S_i \cap G_j} \frac{x_{p(u)}}{x_{e_j}} = \sum_{u \in S_i \cap G_j} \frac{f_{p(u),i}}{x_{e_j}} \leq \frac{f_{e_j}}{x_{e_j}} \leq 1$$

The second equality above is yielded after collapsing the probabilities up the tree from u to r, stopping at e_j because we know it belongs to T_i, as it is an ancestor of $v \in T_i$. The third equality follows from Lemma 1.2, and the first inequality comes from conservation of flow through the subtree G_j. Summing over j:

$$\mathbb{E}[|T_i| \,\big|\, v \in \bar{T}_i] = \sum_{j=0}^{q-1} \mathbb{E}[|T_i \cap G_j| \,\big|\, v \in \bar{T}_i] \leq q \leq H\,.$$

■

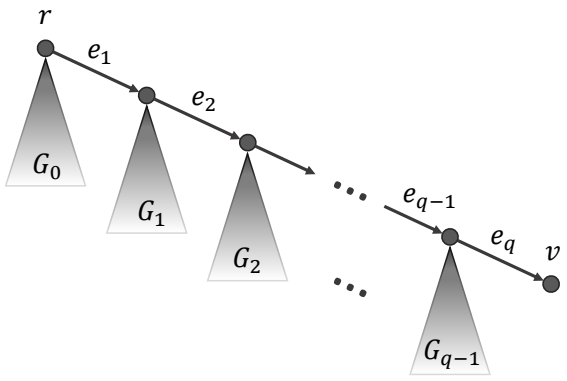

Figure 4: Depiction of subtree definitions for Lemma 1.3. Each subtree G_j may contain some vertices in T_i that we wish to count.

Lemma 1.4 $\mathbb{E}[|\bar{T}_i|] \geq 1/2$.

Proof:

$$\mathbb{E}[|\bar{T}_i|] = \sum_{v \in S_i} Pr[v \in \bar{T}_i] = \sum_{v \in S_i} Pr[v \in T_i] Pr[v \in \bar{T}_i \,\big|\, v \in T_i] = \sum_{v \in S_i} f_{p(v),i} Pr[|T_i| \leq 2H \,\big|\, v \in T_i]$$

Using the conditional expectation derived in the previous lemma, we can employ Markov's inequality to bound the conditional probability term above:

$$Pr[|T_i| \leq 2H \mid v \in T_i] \geq 1 - \frac{\mathbb{E}[|T_i| \mid v \in T_i]}{2H} \geq 1 - \frac{H}{2H} = \frac{1}{2},$$

and observe that we are done:

$$\mathbb{E}[|\bar{T}_i|] = \sum_{v \in S_i} f_{p(v),i} Pr[|T_i| \leq 2H \mid v \in T_i] \geq \sum_{v \in S_i} f_{p(v),i} \frac{1}{2} = \frac{1}{2}.$$

■

We are ready to state the main analytical result of GKR rounding:

Theorem 1.1 $\Pr[T \textit{ connects } S_i] \geq 1/4H$.

Proof: Note that by definition, $|\bar{T}_i| \leq 2H$. Then, since $|\bar{T}_i|$ is an integer-valued random variable:

$$\begin{aligned}
\frac{1}{2} \leq \mathbb{E}\left[|\bar{T}_i|\right] &= \mathbb{E}\left[|\bar{T}_i| \mid |\bar{T}_i| \geq 1\right] Pr\left[|\bar{T}_i| \geq 1\right] + \mathbb{E}\left[|\bar{T}_i| \mid |\bar{T}_i| = 0\right] Pr\left[|\bar{T}_i| = 0\right] \\
&= \mathbb{E}\left[|\bar{T}_i| \mid |\bar{T}_i| \geq 1\right] Pr\left[|\bar{T}_i| \geq 1\right] + 0 \\
&\leq 2H \cdot Pr\left[|\bar{T}_i| \geq 1\right].
\end{aligned}$$

Dividing through, we get $\Pr[T \text{ connects } S_i] = \Pr[|T_i| \geq 1] \geq Pr[|\bar{T}_i| \geq 1] \geq 1/4H$. ■

2 Full Approximation Algorithm

In practice, we want an algorithm that guarantees high, constant probability of feasibility, and low cost among feasible solutions. We can repeatedly apply GKR rounding to achieve this in an acceptable amount of runtime. For constant $\epsilon > 0$, run GKR rounding $N = \lceil 4H \log(k/\epsilon) \rceil$ times and take the solution T to be the union of all the returned trees $T_1, \ldots, T_N$.

Theorem 2.1 *The above is an $O(H \log k)$-approximation algorithm for the Group Steiner Tree Problem. Specifically, for constant $\epsilon > 0$, we can guarantee that the probability of returning an infeasible solution is at most ϵ, the expected cost among feasible solutions is bounded by $O(H \log k) \cdot OPT$, and the runtime is polynomial in $|G|$ and k.*

Proof: First, the runtime is obviously polynomial in $|G|$ and k, since $H \leq |G|$ and the actual rounding takes time linear in $|G|$. We can turn to the probability of T being infeasible. For any of the groups S_i:

$$\begin{aligned}
Pr[T \text{ does not connect } S_i] = \prod_{j=1}^{N} Pr[T_j \text{ does not connect } S_i] &\leq \prod_{j=1}^{N} \left(1 - \frac{1}{4H}\right) \\
= \left(1 - \frac{1}{4H}\right)^{4H \log(k/\epsilon)} &\leq e^{-\log(k/\epsilon)} = \frac{\epsilon}{k}
\end{aligned}$$

Taking the union bound over this happening for any of the k groups:

$$Pr[T \text{ is infeasible}] \leq \sum_i Pr[T \text{ does not connect } S_i] \leq \sum_k \frac{\epsilon}{k} = \epsilon\,.$$

Lastly, the expected cost of feasible solutions. First note

$$\begin{aligned}\mathbb{E}[cost(T)] &= \mathbb{E}[cost(T) \,\big|\, T \text{ is feasible}]Pr[T \text{ is feasible}] + \mathbb{E}[cost(T) \,\big|\, T \text{ is infeasible}]Pr[T \text{ is infeasible}] \\ &\geq \mathbb{E}[cost(T) \,\big|\, T \text{ is feasible}] \cdot (1-\epsilon) + 0\,,\end{aligned}$$

and from before,

$$\mathbb{E}[cost(T)] \leq \sum_{j=1}^{N} \mathbb{E}[cost(T_j)] \leq \sum_{j=1}^{N} \mathrm{LP}^* = \lceil 4H\log(k/\epsilon)\rceil \mathrm{LP}^*$$

Combining these and dividing by $1-\epsilon$:

$$\mathbb{E}[cost(T) \,\big|\, T \text{ is feasible}] \leq \frac{1}{1-\epsilon}\lceil 4H\log(k/\epsilon)\rceil \mathrm{LP}^* = O(H\log k)\mathrm{LP}^*\,.$$

■

As an example, if we want to guarantee that the algorithm will return a feasible solution at least 99% of the time, we would choose $\epsilon = 1/100$ and run GKR rounding at least $4H\log(k/\epsilon) = 4H\log 100k \approx 4H(4.6+\log k)$ times. Moreover, at the loss of a constant approximation factor, any tree can be modified into another tree with depth $H = O(\log n)$. This leads to an $O(\log n \cdot \log k)$-approximation algorithm for group Steiner on trees.

Hardness: It is known that the group Steiner problem on trees is hard to approximate to factor $\log^{2-\epsilon} k$ for any constant $\epsilon > 0$ [3, 2]. So the above approximation algorithm is nearly tight.

References

[1] N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group steiner tree problem. *Journal of Algorithms*, 37(1):66–84, 2000.

[2] E. Halperin, G. Kortsarz, R. Krauthgamer, A. Srinivasan, and N. Wang. Integrality ratio for group steiner trees and directed steiner trees. *SIAM Journal on Computing*, 36(5):1494–1511, 2007.

[3] E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pages 585–594, New York, NY, USA, 2003. ACM.