Massachusetts Institute of Technology — Lecturer: Adriana Lopez
18.434: Seminar in Theoretical Computer Science — March 7, 2006

Steiner Trees and Forests

1 Steiner Tree

Problem Given an undirected graph $G = (V, E)$, a cost function $c : E \to \mathbb{Q}^{+}$, and a partition of V into two sets R and S, find a minimum cost tree in G that contains all the vertices in R and any subset of the vertices in S.

We would like to restrict this problem to instances in which the *triangle inequality* is satisfied, that is, instances in which G is a complete undirected graph and

$$\forall u, v, w \in V : c(u, w) \leq c(u, v) + c(u, w)$$

The problem of solving the Steiner Tree problem with these restrictions is called the *metric Steiner tree problem.* The next theorem shows that an instance of the Steiner tree problem can be reduced in polynomial time to an instance of the metric Steiner tree problem, so that solving the latter implies being able to solve the first.

Theorem 1 *There is an approximation factor preserving reduction from the Steiner tree problem to the metric Steiner tree problem.*

Proof: We want to transform, in polynomial time, an instance I of the Steiner tree problem to an instance I' of the metric Steiner tree problem. Given I, consisting of graph $G = (V, E)$, create I' as follows. For every $u, v \in V$, create an edge (u, v) in G' with cost $c(u, v)$ equal to the cost of the shortest path from u to v in G. Let OPT and OPT' be the cost of an optimal solution in I and I', respectively. We want to show that $OPT = OPT'$. We will show that $OPT \leq OPT'$ and that $OPT' \leq OPT$.

Let e and e' be the edges connecting two vertices u and v in G and G', respectively. Since e is a path from u to v and e' has cost equal to the cost of the shortest path from u to v, we see that $c(e) \leq c(e')$. Thus, for every edge $e \in E$, we have that its cost in G is less than or equal to its cost in G'. This means that the cost of the optimal solution in I is at most the cost of the optimal solution in I', and so $OPT' \leq OPT$.

Next, let T' be an optimal solution to I'; we want to find a solution T in I such that $c(T) \leq c(T')$. Since $OPT \leq c(T)$ for all $T \subseteq E$, this would mean that $OPT \leq c(T') = OPT'$.

Every edge in T' corresponds to a shortest path in G, so replace each edge by this path to get a subgraph $H \subseteq G$. H might contain cycles so remove edges until H is acyclic (but still connected) and call this new subgraph T. Notice that $c(H) = c(T')$, and since $T \subseteq H$, we have that $c(T) \leq c(H) = c(T')$. △

Definition 1 *A* **minimum spanning tree (MST)** *of a graph $G = (V, E)$ is a subset $T \subseteq E$ that connects all vertices in V and whose total cost $c(T) = \sum_{e \in T} c(e)$ is minimized.*

A minimum spanning tree on R is clearly a solution to the metric Steiner tree problem. It is not optimal but its cost is always within a factor of 2 from the optimal solution, as the next theorem illustrates.

Theorem 2 *The cost of an MST on R is within $2 \cdot OPT$.*

Proof: Let OPT be the cost of an optimal solution T to an instance of the metric Steiner tree problem. If we double each edge in T, we get an Eulerian graph connecting all vertices in R and possibly some vertices in S. We can find an Eulerian tour of this graph, whose cost will be $2 \cdot OPT$. Taking this tour and *short-cutting* vertices in R that have already been visited and vertices in S, we obtain a Hamiltonian cycle on R. Because G is complete, the *short-cut* edges are in E and because of the triangle inequality, they do not increase the cost of the tour. Deleting one edge from this Hamiltonian cycle yields a spanning tree of R with cost at most $2 \cdot OPT$. Thus, if M is a minimum spanning tree on R, we have that $c(M) \leq 2 \cdot OPT$. △

2 Steiner Forest

Problem Given an undirected graph $G = (V, E)$, a cost function $c : E \to \mathbb{Q}^+$, and a collection of disjoint subsets of V: $S_1, S_2, \ldots, S_k$, find a minimum cost subgraph in G in which any two vertices belonging to the same set S_i are connected.

Define a *connectivity requirement function* $r : V \times V \to \{0, 1\}$ as follows:

$$r(u, v) = \begin{cases} 1 & \text{if } u \text{ and } v \text{ belong to the same set } S_i; \\ 0 & \text{otherwise.} \end{cases}$$

Then an equivalent formulation of the Steiner forest problem is to find a minimum cost subgraph $H \subseteq G$ such that if $r(u,v) = 1$, then there exists a path from u to v in H, or in other words, if $r(u,v) = 1$, then u and v are in the same connected component of H.

2.1 LP-Relaxation

We define a function $f : \wp(V) \to \{0,1\}$ as follows:

$$f(S) = \begin{cases} 1 & \text{if } \exists\ u \in S \text{ and } v \in \bar{S} \ \text{ such that } r(u,v) = 1; \\ 0 & \text{otherwise.} \end{cases}$$

For $S \subseteq V$, let $\delta(S)$ denote the set of edges crossing the cut $(S, \bar{S})$. f can be viewed as a function on all cuts $(S, \bar{S})$ in G, which specifies the minimum value of $\delta(S)$ in any feasible solution.

Also, for each edge $e \in E$ we introduce the variable x_e defined below

$$x_e = \begin{cases} 1 & \text{if } e \text{ is picked in the subgraph;} \\ 0 & \text{otherwise.} \end{cases}$$

The integer program is:

$$\text{minimize} \sum_{e \in E} c_e x_e$$

$$\text{subject to} \sum_{e:\ e \in \delta(S)} x_e \geq f(S), \ \ S \subseteq V$$

$$x_e \in \{0,1\}, \quad e \in E$$

The LP-relaxation is obtained by letting $1 \geq x_e \geq 0$. The upper bound on x_e is redundant, so we can drop it. Therefore, the LP-relaxation is:

$$\text{minimize} \sum_{e \in E} c_e x_e$$

$$\text{subject to} \sum_{e:\ e \in \delta(S)} x_e \geq f(S), \quad S \subseteq V$$

$$x_e \geq 0, \quad e \in E$$

The dual program is:

$$\text{maximize} \sum_{S \subseteq V} f(S) \cdot y_S$$

$$\text{subject to} \sum_{S:\ e \in \delta(S)} y_S \leq c_e, \ e \in E$$

$$y_S \geq 0, \quad S \subseteq V$$

2.2 Terminology

Say that:

- set S has been *raised* in a dual solution if $y_S > 0$
- edge e is *tight* if $\sum_{S:\ e \in \delta(S)} y_S = c_e$
- at any iteration, set S is *unsatisfied* if $f(S) = 1$ but no picked edge crosses the cut $(S, \bar{S})$
- set S is *active* if it is a minimal unsatisfied set in the current iteration

Notice that raising a set S for which $f(S) = 0$ does not affect the dual objective function. So we can assume that such sets are never raised.

Primal Conditions: For each $e \in E$, $x_e = 1$ implies $\sum_{S:\ e \in \delta(S)} y_S = c_e$. In other words, every edge that is picked must be tight.

The algorithm starts with null primal and dual solutions. At any given iteration, the current primal solution indicates which sets need to be raised and the current dual solution indicates which edge needs to be picked.

Lemma 3 *Set S is active iff it is a connected component in the currently picked forest and $f(S) = 1$.*

Proof: Let S be an active set. If S contains part of a connected component, then there would already be an edge crossing the cut $(S, \bar{S})$ which would contradict the fact that S is active. So S must be a union of connected components. Since $f(S) = 1$, there must be vertices $u \in S, v \in \bar{S}$ such that $r(u, v) = 1$. Let S' be the connected component that contains u, then S' is also unsatisfied. Since S is minimal, we must have $S' = S$. The reverse follows by the definition of an active set. $\triangle$

2.3 Algorithm

By lemma 3, all active sets can be easily found in the current iteration. The algorithm will then raise the dual variables of these sets in a synchronized manner until some edge goes tight. A tight edge is then picked arbitrarily and the iteration terminates. This process ends when a primal feasible solution F is found. However, F might contain *redundant* edges, that is, edges without which F would still be a feasible solution. So the algorithm drops all redundant edges and returns the newly *pruned* F.

1. **Initialization**
   $F \leftarrow \emptyset$
   for each $S \subseteq V$, $y_S \leftarrow 0$
2. **Edge augmentation**
   while there exists an unsatisfied set do:
   simultaneously raise y_S for each active set S, until some edge e goes tight $F \leftarrow F \cup \{e\}$
3. **Pruning**
   return $F' = \{e \in F \mid F - \{e\} \text{ is primal infeasible}\}$

Lemma 4 *At the end of the algorithm, F' and $\mathbf{y}$ are primal and dual feasible solutions, respectively.*

Proof: At the end of step 2, all connectivity conditions are met (otherwise, another iteration would follow). Moreover, only connected sets are raised, so no edge between two vertices in the same component can go tight, and therefore F is acyclic, i.e. it is a forest. Hence, if $r(u, v) = 1$ then there is a unique path from u to v in F, and each edge in this path is non-redundant and will not be removed in step 3. Thus, F' is a feasible solution. Now, when an edge goes tight, the current iteration terminates and active sets are redefined, so no edge is overtightened. This means that $\mathbf{y}$ is a feasible solution to the dual program.

$\triangle$

Lemma 5 *Consider any iteration of the algorithm and let C be a component of the graph formed by the currently picked edges. If $f(C) = 0$ then $deg_{F'}(C) \neq 1$.*

Proof: Suppose, for the sake of contradiction, that $f(C) = 0$ and $deg_{F'}(C) = 1$. Let e be the unique edge connecting C and $\bar{C}$. Since $e \in F'$, we know that e is not redundant, so there exist vertices u, v such that $r(u, v) = 1$ and e is part of the unique path between them. Since e is the only edge crossing the cut $(C, \bar{C})$, it must be the case that one of the vertices is in C and the other is in $\bar{C}$. But we know $r(u, v) = 1$, so $f(C) = 1$ which is a contradiction. $\triangle$

Lemma 6 $\sum_{e \in F'} c_e \leq 2 \sum_{S \subseteq V} y_S$

Proof: Every edge picked by the algorithm is tight, so

$$\sum_{e \in F'} c_e = \sum_{e \in F'} \left(\sum_{\text{S: } e \in \delta(\text{S})} \text{y}_\text{S} \right)$$

If we change the order of the summations, we get

$$\sum_{e \in F'} c_e = \sum_{S \subseteq V} \left(\sum_{e \in \delta(S) \cap F'} y_S \right) = \sum_{S \subseteq V} \deg_{\text{F}'}(S) \cdot y_S$$

It is therefore sufficient to show that

$$\sum_{S \subseteq V} \deg_{\text{F}'}(S) \cdot y_S \leq 2 \sum_{S \subseteq V} y_S$$

We will prove that in each iteration, the change in the left hand side is at most the change in the right hand side. Consider an iteration and let Δ be the amount by which the dual variables were raised in this iteration. We want to prove that:

$$\Delta \times \left(\sum_{S \text{ active}} \deg_{\text{F}'}(S) \right) \leq 2\Delta \times (\# \text{ active sets})$$

or equivalently,

$$\frac{\sum_{S \text{ active}} \deg_{\text{F}'}(S)}{\# \text{ active sets}} \leq 2$$

Let $H = (V, F')$ and consider the set of connected components with respect to F at the current iteration. Shrink each one of these components to a node and call the new graph

H'. Let V' and E' denote the set of vertices and edges of H', respectively. Notice that all vertices picked before the current iteration have been shrunk. Further notice that if the node $s \in V'$ corresponds to the connected component S in H, then

$$\deg_{\mathrm{F}'}(S) = \deg_{H'}(s)$$

Call a node $s \in H'$ *active* if it corresponds to an active set in H and *inactive* otherwise. We want to prove that

$$\frac{\sum_{s \text{ active}} \deg_{H'}(s)}{\# \text{ active nodes in } H'} \leq 2$$

or equivalently, that

$$\sum_{s \in V'} \deg_{H'}(s) - \sum_{s \text{ inactive}} \deg_{H'}(s) \leq 2 \times (\# \text{ active nodes in } H')$$

Notice that if $s \in V'$ is an active node, then $\deg_{H'} > 0$ because there must be an edge incident to it to satisfy connectivity requirements. If s is inactive then by lemma 5 we know that $\deg_{H'}(s) \neq 1$, so either $\deg_{H'}(s) = 0$ or $\deg_{H'}(s) \geq 2$.

By the handshaking lemma, we know that

$$\sum_{s \in V'} \deg_{H'}(s) = 2|E'| = 2|V'| - 2 \times (\# \text{ connected components of } H') \leq 2|V'|$$

This inequality still holds if we remove isolated vertices from V' because each time we remove an isolated vertex, both $|V'|$ and the number of connected components decrease by 1. So remove any isolated vertices from H'; notice that any vertex we remove is inactive and that the inactive nodes that remain have degree greater than or equal to 2. Thus

$$\sum_{s \text{ inactive}} \deg_{H'}(s) \geq 2 \times (\# \text{ of inactive nodes in} H')$$

This means that

$$\begin{aligned} \sum_{s \in V'} \deg_{H'}(s) - \sum_{s \text{ inactive}} \deg_{H'}(s) &\leq 2|V'| - 2 \times (\# \text{ of inactive nodes in} H') \\ &= 2 \times (\# \text{ active nodes in } H') \end{aligned}$$

This proves our claim. △

Theorem 7 *The algorithm given above is a 2-factor approximation algorithm to the Steiner forest problem.*

Proof: By lemma 4, we know that F' and $\mathbf{y}$ are primal and dual feasible solutions, respectively. By lemma 5, we know that $\sum_{e \in F'} c_e \leq 2 \sum_{S \subseteq V} y_S$. Since no set S for which $f(S) = 0$ was ever raised, we know that $2 \sum_{S \subseteq V} y_S = 2 \sum_{S \subseteq V} f(S) \cdot y_S$. This is the dual objective function, which means that if OPT_f is the optimal solution to the dual program, then

$$\sum_{S \subseteq V} f(S) \cdot y_S \leq OPT_f$$

But we know that the optimal solution to the fractional program is at most the optimal solution to the integer program, so

$$2 \sum_{e \in F'} c_e \leq 2 \cdot OPT_f \leq 2 \cdot OPT$$

△

References

[1] Vazirani, Vijay V. "Approximation Algorithms" Chapters 3.1 and 22, Springer 2003