College of Computer & Information Science
Northeastern University
CS G399: Algorithmic Power Tools I

Fall 2007
9 October 2007
Scribe: Laura Poplawski

Lecture Outline:

- Network Design
  - Steiner Forest (primal-dual algorithm)
  - Generalized Steiner Network (iterated rounding algorithm) - postponed to next lecture

1 Steiner Forest

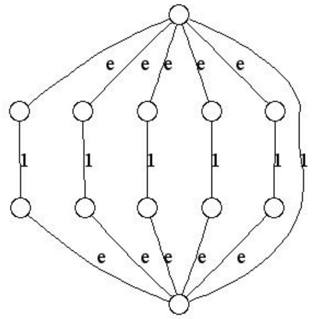
Problem 1. Given: graph $G = (V, E)$, edge cost function $C : E \rightarrow \mathcal{Q}$, collection of sets $\{S_1, S_2, \ldots, S_k\} \subseteq V$.
Goal: Determine subgraph H of G with least cost such that $\forall i$, S_i is connected in H.

2 Relationship to MST and Steiner Tree

We note that Minimum spanning tree (MST) is a special case of Steiner Forest in which $k = 1$ and $S_1 = V$ (we want to connect all the nodes using a min-cost subset of the edges). Minimum Steiner tree is a special case in $k = 1$ and S_1 is an arbitrary subset of V. Since Steiner Tree is NP-hard, Steiner forest is also NP-Hard.

There is a simple 2-approximation for Steiner Tree. Create a new graph consisting of all the nodes of the set S that we want to connect, plus an edge connecting each pair of nodes with length = the shortest path length from the original graph. Solve MST on this new graph and use this MST to create a Steiner tree on the original graph (by removing unnecessary edges). To verify that this is a 2-approximation: consider the minimum Steiner tree on the original graph, and consider a tour of all nodes in S that visits each edge of the Steiner tree at most twice. If we visit the nodes in the same order in our new graph, we will get a spanning tree that is at least as expensive as the MST, with cost at most the cost of the tour on the original graph, which is at most twice the cost of the min Steiner tree. The best known (poly-time) approximation factor is 1.55 (see [2]).

Since we have good approximations to Steiner Tree, it is tempting to solve Steiner Forest by merging together Steiner trees for each S_i. However, the following example shows that this could lead to an approximation ratio as high as k, the number of sets in the given instance.



3 Linear Program for Steiner Forest

There are two linear program approaches to solve this problem. One uses flows, the second uses cuts. To use flows, we would define a variable representing the flow across each edge. There would be a constraint for each pair $(u, v) \in S_i$ (for all i) saying that the flow between u and v must be at least 1. The capacity of the flow across each edge would be another variable, representing whether or not that edge is included in the solution. This LP has a polynomial number of constraints and a polynomial number of variables. However, it is not clear how to use a primal-dual approach with the flow-based LPs. Therefore, we will concentrate on the cut-based LP.

We define variables $x_e \in \{0, 1\}$, for all edges e. $x_e = 1$ if e is part of the forest, 0 otherwise.

Notation: $c_e =$ the cost of edge e. $\delta(S) =$ the set of all edges that are cut by cut $(S, \bar{S})$. $f(S)$ is a function defined on the set of all cuts of G. $f(S) = 1$ if there exists a pair (u, v) such that $(u, v) \in S_i$ for some i, $u \in S$, $v \in \bar{S}$.

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

$$\forall S, \sum_{e \in \delta(S)} x_e \geq f(S)$$

$$\forall e,\ x_e \geq 0$$

Note that this LP has an exponential number of constraints, and therefore cannot immediately be solved as-is in polynomial time. For the primal-dual approach we do not need to solve the LP, so we defer the solvability of the LP. When we consider the more general Steiner Network problem, we will in fact show that the LP is solvable in polynomial time.

Claim 1. *An integer solution to the above LP is exactly Steiner Forest.*

Proof. Given Steiner Forest H, construct solution X for the LP as follows.

$$x_e = \begin{cases} 1 & e \in H \\ 0 & e \notin H \end{cases}$$

Consider any cut S such that $f(S) = 1$. Then $\exists (u, v)$ such that $u \in S$, $v \in \bar{S}$, $(u, v) \in S_i$ for some i. u and v must have some path between them in H, since H is a valid Steiner forest. So there must be some edge e crossing cut S, $x_e = 1$, satisfying the constraint for S. So X is a valid solution to the LP. Clearly, the cost of H = the value of X.

Now take integer solution X to the LP. The first constraint of the LP combined with the fact that the objective is minimization implies that no x_e will have a value greater than 1. Therefore, each (integer) x_e value is 0 or 1. Construct a Steiner forest H by adding each edge that has $x_e = 1$.

Suppose H does not connect some S_i. Then, there exist $(u, v) \in S_i$ such that there is no path in H connecting u and v. Let set S = the set of all nodes connected to u in H (and notice that $v \notin S$). Take cut $(S, \bar{S})$. $f(S) = 1$ by definition of S. Since X is a valid solution to the LP, there must be some edge e crossing S such that $x_e = 1$. However, this contradicts our definition of S. So H must connect all S_i, and therefore be a valid Steiner forest. Again, clearly the cost of H = the value of X. □

Claim 2. *The above LP has an integrality gap $\geq 2 - 1/n$.*

Proof. Consider a cycle of length n, all edges have cost 1, $S_1 = V$ (MST on a cycle). The minimum Steiner forest has cost $n - 1$. However, the LP can be solved by setting $x_e = .5$ for all e, giving a value $\frac{n}{2}$. □

4 Dual to the LP

$$\text{maximize} \sum_S f(S) y_S$$

$$\forall e, \sum_{S: e \in \delta(S)} y_S \leq c_e$$

$$\forall S,\, y_S \geq 0$$

5 Geomans and Williamson primal-dual algorithm ([1])

Intuition: We will grow a tree starting at each $u \in S_i$ for each i. We will stop as soon as the primal solution is feasible. In each step, we increase the y_S values for cuts that make the primal infeasible until the dual constraint for some edge e becomes tight. We will set $x_e = 1$ for one such e, satisfying at least one more primal constraint, and remove any edges that might create a cycle. When the primal solution is feasible, we will go back and delete any unnecessary edges.

However the above approach, stated as is, turns out to be similar as solving separate Steiner tree problems at the same time, which we know will not give a good approximation ratio. To fix this, we will carefully pick which y_S to grow at each step. In the algorithm below, an "active set" is a minimal set whose primal constraint has not yet been satisfied.

The primal-dual algorithm follows:

```
$F \leftarrow 0$;
$y_S = 0, \forall S$;
initially, active sets = $\{\{u\} : u \in \cup_i S_i\}$
```

```
4: while $\exists$ active set $S$ for which primal constraint is not satisfied do
5:     raise $y_S$ uniformly for all active sets $S$ until some edge $e$ has $\sum_{S:e\in\delta(S)} y_S = c_e$. (notice: this
       may raise the $y_S$ values by 0 if an edge is already tight).
6:     add $e$ to $F$ (if multiple $e$ became tight, pick only one of them)
7:     remove all tight edges that haven't been added yet that would form a cycle in $F$
8: set $F' = F$ - $\{e$ : removing $e$ does not violate any connecting constraint$\}$.
9: return $F'$.
```

Observation: active sets are connected components in F.

Lemma 1. *F' is primal feasible and y is dual feasible.*

Proof. By design, F is a forest which satisfies all constaints.

Let e be an edge removed from F at step 8. Then there are no pairs on opposite sides of e that want to be connected. So F' is primal feasible.

Now show that Y is dual feasible. Consider edge e.

If e never becomes tight, then the constraint for e must be obeyed.

If e does become tight at some point, then after this point, there is no active set S such that e crosses S. To see this, we note that for an active set S (a) if e crosses S and we included e in the solution, then the cut S constraint is satisfied by e; (b) if e crosses S and we removed e from the solution in step 7, then including e would close a cycle; either all nodes in the cycle are part of S, all nodes are not part of S, or some chosen edge in the cycle crosses S. Since we increase y_S only for active sets, $\sum_{S:e\in\delta(S)} y_S$ will remain $= c_e$. □

Lemma 2. *$cost(F') \leq 2 \cdot cost(y)$. That is,*

$$\sum_{e\in F'} c_e \leq 2 \sum_{S:f(S)=1} y_S$$

Proof.

$$\begin{aligned}
\sum_{e\in F'} c_e &= \sum_{e\in F'} \sum_{e\in\delta(S)} y_S \text{(since we only picked tight edges } e\text{)} \\
&= \sum_{S:f(S)=1} \left[y_S \sum_{e\in F'\cap\delta(S)} 1 \right]
\end{aligned}$$

If we could say that $degree_{F'}(S) \leq 2$, we would be done (since we can just use the relaxed complementary slackness conditions. But this isn't necessarily true. Instead, our goal is to prove that

$$\sum_S y_S * degree_{F'}(S) \leq 2 * \sum_S y_S$$

We will show this using induction on the iterations of the algorithm. Initially, all y_S values are 0, so it's true. At any iteration, we start with F = the collection of connected components. Some are active sets, others are not. There are no other active sets. (An active set cannot be smaller than a connected component, because the cut constraint is satisfied for sets smaller than a connected component. An active set cannot be larger than a connected component because our active sets are the minimal sets that don't satisfy their constraints.)

Suppose in an iteration, we increase y_S for all active S by Δ. The right hand side of the above equation increases by 2Δ times number of active sets. To evaluate the left hand side, we must consider the final forest F. Imagine F with each current connected component collapsed into a single node. In this revised F, none of the inactive connected components will be leaves, because we would have removed edges connecting these components to the rest of the forest during step 8. Therefore, the average degree of all active connected components is the average degree of a subset of the nodes in a tree, including all of the leaves. So the average degree is at most 2. So the left hand side of the above equation also increases by at most 2Δ times the number of active sets, thus completing the proof of the lemma. □

References

[1] M.X. Goemans and D.P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal of Computing*, 24(2): 296-317, 1995.

[2] G. Robins and A. Zelikovsky. Tighter Bounds for Graph Steiner Tree Approximation. *SIAM Journal on Discrete Mathematics*, 19:122-134, 2005.