Network Design Foundation
Fall 2011
Lecture 9

Instructor: Mohammad T. Hajiaghayi
Scribe: Vaibhav Singh

October 26, 2011

1 Overview

We study the Primal Dual approach in approximation algorithms for the NP hard problems of: STEINER TREE PROBLEM andPRIZE COLLECTING STEINER TREE PROBLEM.

2 STEINER TREE PROBLEM

(1)**Definition**

INPUT : A graph $G = (V, E)$, the cost function for each edge C:E$\rightarrow \mathbb{R}^+$ and τ a set consisting of terminal nodes such that $\tau \subseteq V$

GOAL: Find a minimum cost $T \subseteq E$ such that it connects each of the terminal nodes.
The problem is equivalent to connecting all the terminal nodes to the root node r. A generalization of this problem is the Steiner forest problem and dual of the Steiner tree problem is the Multi-cut problem where our goal is to delete the edges in a given graph so as to make terminal nodes disconnected.

(2) **Simple 2-Approximation Algorithm**
Without loss of generality, we assume that G is always a complete graph, moreover, C obeys triangle inequalities.
Algorithm:- Construct the subgraph including τ (terminal nodes) and excluding all the Steiner nodes after metric completion and find the Minimum Spanning Tree on this subgraph.

Analysis:-Since OPT is a tree, one could do the depth-first search on it from

an arbitrary node, which constructs a Eulerian tour on the nodes spanned by OPT. Note the cost is doubled for the tour compared to OPT. Then revise the tree by skipping all of the Steiner nodes and joining consecutive terminal nodes according to the tour, which returns a spanning tree on T. By triangle inequalities on C the cost of this algorithm is less than or equal to two times the cost of the optimum. Hence this is a 2-approximation algorithm.
Considering a graph centered at D in figure1 where node D is the Steiner node and the nodes A,B,C are the terminal nodes. It is easy to see that the star graph is the Steiner tree of total cost =3 and if we just consider the maximum spanning tree for the terminal nodes we have to pay a cost of 4-2k. Extending this example to the n terminal nodes would give us the lower bound $(2-k)(n-1)/n = 2-2/n$ approximation factor of the above algorithm where $k \to 0$.

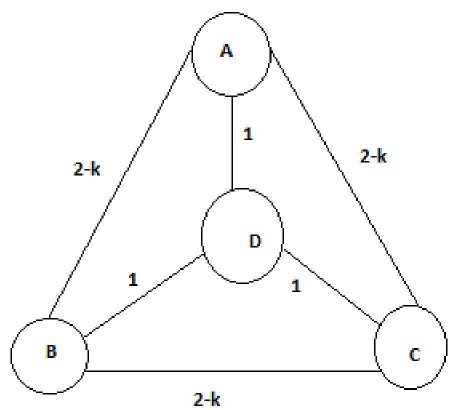

Figure 1: Steiner tree

3 Approximation via Primal Dual Algorithm

The notations are :-

1) $S \subseteq V$, $\delta(S) = e_{u,v} \in E, u \notin S, v \in s$

2) We denote S as a good set if $S \subseteq V : S \bigcap \tau \neq \phi, \neq \tau$

3)The cost C(F) where $F \subseteq E$, is equal to $\sum_{e \in F} C_e$, C_e is the cost needed to buy the edge e

We can formulate the steiner tree problem as an Integer program in the following fashion:-

$$\begin{array}{lr} \text{minimize:} & \sum_{e \in E} C_e x_e \\ \text{subject to} & \sum_{e \in \delta(S)} x_e \geq 1, \forall S \subseteq goodset \\ & x_e \in \{0, 1\}| \end{array} \tag{1}$$

Here $x_e = 1$ denotes that edge e is included in our solution otherwise $x_e = 0$. The first constraint is the necessary and sufficient condition for T to span the set of terminal nodes τ which can be seen by the application of max flow-min cut theorem

The above IP can be used to obtain the LP by relaxing the second constraint and allowing x_e to take non-negative real values. If we have a look at the dual of this LP it is quite similar to the set cover problem LP and hence the dual in this case is called covering LP. The dual LP is given as :-

$$\begin{array}{lr} \text{maximize:} & \sum_{S \in goodset} y_s \\ \text{subject to:} & \sum_{S: e \in \delta(S)} y_s \geq C_e, \forall e \in E \\ & y_s \geq 0 \end{array} \tag{2}$$

We use the primal-dual approximation algorithm technique as in some cases it is easier to find a feasible solution to the dual problem. In this particular case both the primal and the dual programs are feasible. So the complementary slackness constraints are satisfied for the pair of optimal solutions.

$$\forall e \in E, x_e > 0 \Rightarrow \sum_{S: e \in \delta(S)} y_s = C_e$$

thus for every edge, the primal buys exactly what the dual pays for it moreover each cut pays exactly for one edge.We know that the solution of the Primal LP is the same as Dual LP moreover a feasible solution to the dual problem is less than or equal to the OPT of the dual and hence less or equal to the OPT of the primal LP.

scribe: Vaibhav Singh
Lecture 9 **Date:** 10/26/2011

Agrawal-Klein-Ravi Algorithm for Steiner Tree

F is null
repeat
repeat

- Grow at unit rate y_s for every component induced by F that contains a terminal.

  Until

- some edge e goes tight (is paid for by the cut it crosses)[Break ties arbitrarily].

- $F \leftarrow F + e$[This will never create a cycle because edge (u; v) stops being paid off once u and v are in the same component].

- Add edges without creating cycles and ties are broken arbitrarily. until F spans τ.

  Pruning: iteratively delete edges to steiner nodes that are leaves

Now we look at the graph in Figure 2, where $\{a, b, e, f\}$ are the terminal nodes and $\{c, d, g\}$ are the Steiner nodes.
We start at the terminal nodes and grow the radius of the balls centered at these vertices at rate=1.
Grow y_a, y_b, y_e, y_f at rate=1

Time=1 :- After one unit of time we merge ac,bd,eg.
Grow $y_{a,c}, y_{b,d}, y_{e,g}, y_f$ at rate=1
Time =2:- we merge aceg,bdf

.... we continue till each of the terminal nodes have been merged and hence a laminar family is formed.

Finally after getting the solution we prune the solution to remove the unnecessary Steiner edges. In this example we prune(remove) the edge eg after

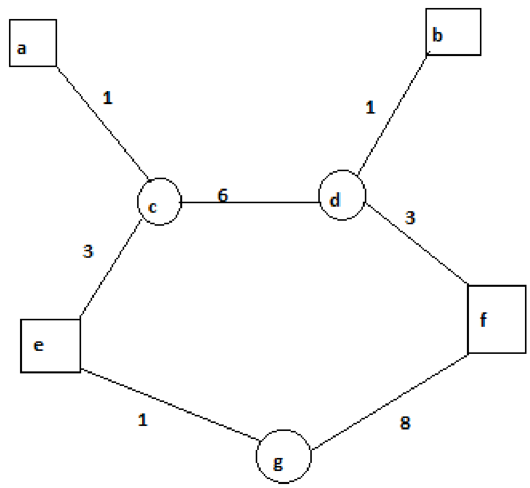

Figure 2: AKR Algorithm example

obtaining the solution .A similar kind of algorithm can also be used to give a 2-approximation for the Steiner Forest problem

Theorem 1
F output by the algorithm satisfies $C(F) \leq 2(1-1/n)\sum y_s$, where y_s values untouched by the algorithm are implicitly set to 0.(n is the number of terminal nodes)

Proof
The final solution is given in Fig.3 where A denotes the active component of the graph (containing at least 1 terminal nodes) and I denotes the components containing steiner nodes hence inactive.
Let us assume the growth in the length of the edge in ϵ time interval (assuming unit rate growth) is ϵ units.
For the primal LP we need to pay the following in ϵ time interval :-

$$\left(\sum_{v\in A} \mathsf{degree}(v)\right)\epsilon$$

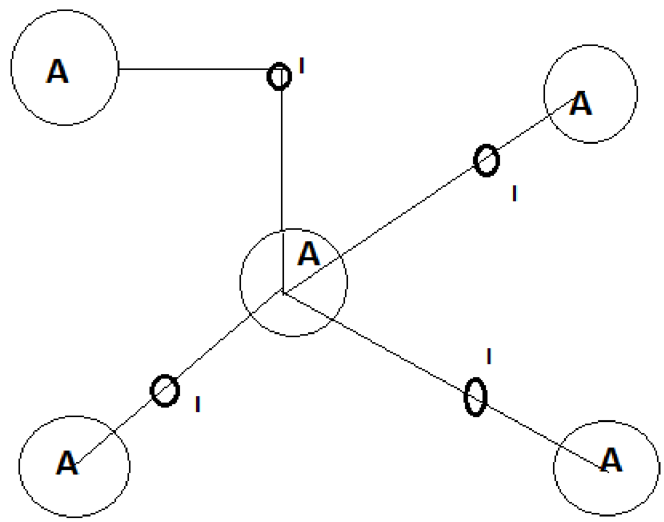

Figure 3: Analysis Theorem1

for the dual LP we need to pay $\epsilon|A|$

$$(\sum_{v \in A \cup I} degree(v)) = 2(|A| + |I| - 1)$$

$$(\sum_{v \in I} degree(v)) \geq 2|I|$$

$$\text{hence } (\sum_{v \in A} degree(v)) \leq 2(|A| - 1)$$

therefore, the rate of increase of cost of primal solution=2 X the rate of increase of cost of dual solution
by induction argument we get $C(F) \leq 2(1 - 1/n) \sum y_s$ and an approximation factor of 2.
Similar analysis can be applied to the Steiner Forest Problem

4 Prize Collecting Steiner Tree

In conjugation to the Steiner tree problem this problem has a prize function $\pi : V \to R_+$ which assosciates a non-negative value to each of the node once

reached and also a root $r \in V$ from which we build the Steiner tree. The goal is to construct a r rooted tree F so as to maximize the quantity :-
$\pi(F) - C(F)$;
however it is NP-hard to determine whether the optimum value of this quantity is positive so we need to massage it to design an efficient approximation algorithm. Note that maximizing the above quantity is equivalent to :-
maximize:$\pi(F) - C(F) - \pi(V)$
which is equivalent to:-
minimize:$\pi(\bar{F}) + C(F)$ which can be interpreted as minimizing the total penalty of not covering a particular terminal node added with the cost of connecting the remaining terminal nodes. This problem can be mapped to the real world problem in which the service provider (example ATT), wants to connect a set of locations but has an extra degree of freedom of not connecting to some of the locations which have a very high deployment cost (example challenging terrain) as compared to the penalty incurred by not covering them. ATT saved millions of dollars by modelling the laying of the optical fibers as the Prize collecting Steiner Tree problem.

4.1 Primal Dual LP Solution

We can define the following IP from the above problem statement:-

$$\begin{array}{rr} \text{minimize:} & \sum_{e \in E} C_e x_e^* + \sum_{t \subseteq V-r} z_t^* \pi_t \\ \text{subject to :} & \sum_{e \in \delta(S)} x_e^* + z_t^* \leq 1, \forall t \in \tau, \forall S \subset V, r \notin S, t \in S. \\ & x_e^* \geq 0; z_t^* \geq 0 \end{array} \tag{3}$$

$z_t^* = 1$when we do not cover the terminal node t and therefore have to pay a penalty otherwise $z_t^* = 0$.
The first constraint signifies that one can be free once he is willing to pay the penalty by not covering a particular terminal node. The above IP can be relaxed to LP but the number of constraints are not polynomial . However, we can use a separation time oracle to solve polynomial time using ellipsoid.
A 3-Approximation algorithm
We solve the LP $(x, z)$and if the value of $z_t \geq 1/3$ we have to pay a penalty otherwise connect the terminal nodes using the Steiner tree algorithm discussed earlier.
Analysis:-
If $z_t < 1/3$ then $\sum_{e \in S} x_e \geq 2/3$

$$3/2 \sum_{e \in S} x_e \geq 1$$

Now we know for the Steiner tree problem that 2 times feasible dual cost $\leq$ 2 times optimal dual solution =2*optimal primal cost$\leq$ 2*feasible primal cost So we pay 2 times the feasible solution for the Steiner tree problem

the total cost $\leq 3/2 * 2 * (\sum_{e \in E} C_e x_e) + 3 * (\sum_{t \subseteq V-r} z_t \pi_t)$
hence we get an approximation factor of 3. Instead if we choose the factor (1/3) from some distribution we can obtain an approximation factor of 2.54 which is the best known approximation factor for the prize collecting Steiner forest problem.

Some best known approximation algorithm factors TSP:- 1.50 ; Prize collecting TSP:-1.91
Steiner Tree:-1.39 ; Prize collecting Steiner Tree:-1.96;
Steiner Forest:-2.00 ; Prize collecting Steiner forest:-2.54

5 References

[1] A.Agrawal , P.Klein, R.Ravi 'When trees collide:an approximation algorithm algorithm for the generalized Steiner problem on network'.STOC'91, Proceedings of the twenty-third annual ACM symposium on Theory of computing.