Network Design and Game Theory
Spring 2008
Lecture 6

Instructor: Mohammad T. Hajiaghayi
Scribe: Mangesh Gupte

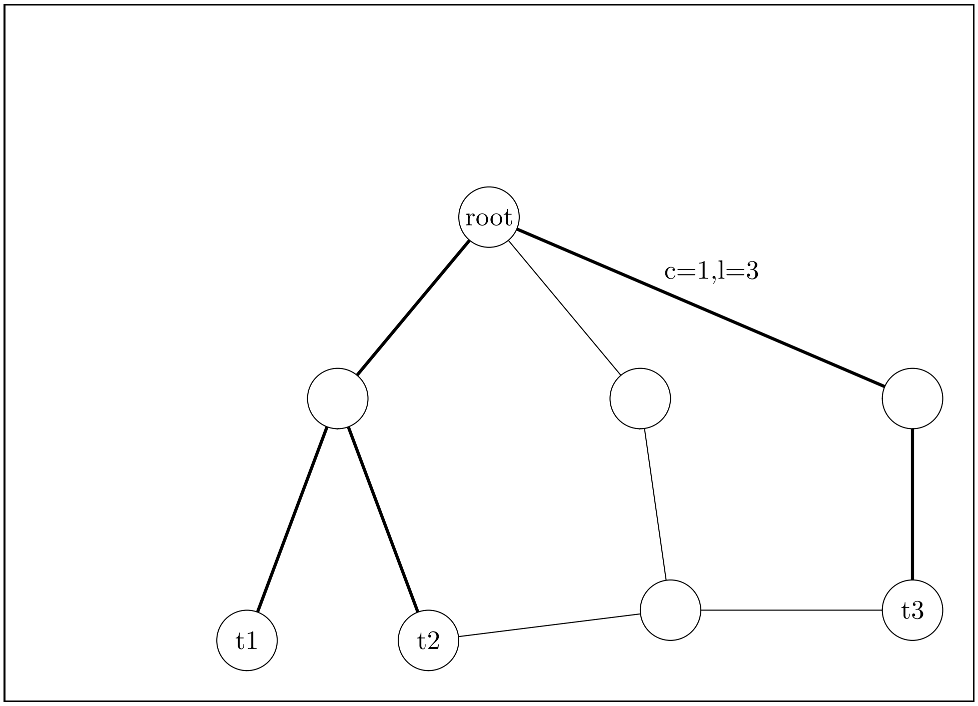
March 24, 2008

1 Cost-Distance Network Design

The COST-DISTANCE NETWORK DESIGN problem is very similar to the RENT-BUY and BUY-AT-BULK problems that we considered in previous lectures. We want to buy a set of edges in a graph, such that the sum of the cost of buying these and the total weighted distance in the subgraph of bought edges of a given set of terminal nodes from the root is minimized. In this lecture we see how to obtain a $\mathcal{O}(\log |S|)$ approximation to this problem, where $|S|$ is the number of terminals. We start with a formal definition of the problem.

Problem Definition : We are given an undirected graph $G = (V, E)$ with a designated root vertex r and a set of terminals(demands) S. On the edge set E, there are two non-negative real-valued functions, namely a cost(buy) function c and a length(rent) function l and a non-negative weight function w on S.

The objective is the construct a solution T (tree) that connects the terminals in S to the root and minimizes $\sum_{e\in T} c_e + \sum_{t\in S} w(t) \cdot l_T(r, t)$, where $l_T(r, t)$ is the length of the path in T from t to r (essentially the rent cost).



For each $t \in S$ let P_t denote the set of directed paths from the root r to t. In general P_t will be exponential sized. Also note that $P_t \cap P_{t'} = \phi$.

The following Integer Program exactly models this problem. For each edge e, $x(e)$ is the amount of edge e we buy, and for each path p the (fraction of) flow through it is $f(p)$.

$$
\begin{aligned}
\min \sum_{e \in E} c(e)\, x(e) + \sum_{t \in S} w(t) \sum_{p \in P_t} l(p)\, f(p) & \\
\text{s.t.} \sum_{p \in P_t : e \in p} f(p) \leq x(e) & \qquad \forall e \in F, t \in S \\
\sum_{p \in P_t} f(p) \geq 1 & \qquad \forall t \in S \\
f(p), x(e) \in \{0, 1\} & \qquad \forall e \in E, t \in S, p \in P_t
\end{aligned}
$$

We relax the last constraint to $f(p), x(e) \geq 0$ to get a LP.

Remark 1 (Flow LP) *Although this LP has an exponential number of variables (the $f(p)$'s), it does not require this. We can rewrite this in the form of a flow LP. In that we only use variables of the form $f^t(e)$ for every terminal t and edge e. Since there are only a polynomial number of terminals and edges the number of these variables is also polynomial.*

We are now ready to describe the main algorithm, for this problem. Consider a complete graph H on $S \cup \{r\}$, with weight function b on its edges s.t. $b(u, v)$ is the weight of the shortest path in G connecting u and v with length $m(e) = c(e) + \min\{w(u), w(v)\} \cdot l(e)$.

Algorithm 1: Cost-Distance Network Design

Input: Set of terminals $|S|$, distance function b on it.

1 If $S = \{r\}$, return tree $\{r\}$
2 Compute the shortest path (u, v) for $u, v \in S$, in graph H
3 With probability $\frac{w(u)}{w(u)+w(v)}$ set the demand of u to $w(u) + w(v)$ and $S \leftarrow S - \{u\}$
4 With probability $\frac{w(v)}{w(u)+w(v)}$ set the demand of v to $w(u) + w(v)$ and $S \leftarrow S - \{v\}$
5 Recursively obtain a solution to the reduced problem
6 Connect v(if v is deleted) to the root via u using the shortest path.

Lemma 1 *For any instance of the problem there is a shortest path of length at most* $2\ \mathsf{OPT}_{\mathsf{LP}}/|S|$

Lemma 2 $\mathsf{E}\left[\mathsf{OPT}_{\mathsf{LP}}(I')\right] \leq \mathsf{OPT}_{\mathsf{LP}}(I)$ *where* I *is the original LP and* I' *is the reduced instance in step 4 of the above algorithm.*

Proof: Let (x^*, f^*) be the solution for LP in instance I. Then (x^*, f^*) is a solution for LP in instance I'.
In I we pay amount

$$P(I) = w(u) \cdot \sum_{p \in P_u} l(p)\, f(p) + w(v) \sum_{p \in P_v} l(p)\, f(p)$$

In I' we pay

$$\begin{aligned} P(I') =& (w(u) + w(v)) \sum_{p \in P_u} l(p)\, f(p) \frac{w(u)}{w(u) + w(v)} \quad + \\ & (w(v) + w(u)) \sum_{p \in P_v} l(p)\, f(p) \frac{w(v)}{w(u) + w(v)} \\ =& w(u) \cdot \sum_{p \in P_u} l(p)\, f(p) + w(v) \sum_{p \in P_v} l(p)\, f(p) \end{aligned}$$

■

Lemma 3 *In step 5 of the above algorithm, the expected cost of routing is at most twice the length of the path* $b(u, v)$ *in* H.

Proof: The expected cost of routing due to step 5 is,

$$\begin{aligned}
\mathsf{cost} &= \frac{w(u)}{w(u)+w(v)} w(u) l(p^*) \quad + \quad \frac{w(v)}{w(u)+w(v)} w(u) l(p^*) \\
&= \frac{2w(u)w(v)}{w(u)+w(v)} \sum_{e \in p^*} l(e) \\
&\leq 2 \min\{w(u), w(v)\} \sum_{e \in p^*} l(e)
\end{aligned}$$

■

Theorem 1 *There is a randomized algorithm with* $\mathcal{O}(\lg|S|)$ *approximation for the* Cost-Distance Network Design *problem.*

Proof: Using induction, we prove that the cost produced by the agorithm is bounded by $4\,\mathsf{H}_{|S|}\,\mathsf{OPT}_{\mathsf{LP}}(I)$. By linearity of expectation,

$$\begin{aligned}
\mathsf{cost} = \quad & \underbrace{4\,\mathsf{H}_{|S|-1}\,\mathsf{OPT}_{\mathsf{LP}}(I)}_{\text{Induction Hypothesis}} + \underbrace{2 \cdot l^*(p^*)}_{\text{Lemma 3}} \\
&\leq 4\,\mathsf{H}_{|S|-1}\,\mathsf{OPT}_{\mathsf{LP}}(I) + 2 \cdot \frac{2 \cdot \mathsf{OPT}_{\mathsf{LP}}(I)}{|S|} \\
&= 4\,\mathsf{H}_{|S|}\,\mathsf{OPT}_{\mathsf{LP}}(I)
\end{aligned}$$

■

References

[1] Adam Meyerson, Kamesh Munagala, and Serge Plotkin. Cost-Distance: Two-Metric Network Design. In *Proc. of IEEE Symposium on Foundations of Computer Science (FOCS)*, 2000.

[2] C. Chekuri, S. Khanna, S. Naor. A deterministic algorithm for the cost-distance problem. In the *Proc. of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2001, pages 232-233.

[3] C. Chekuri, M.T. Hajiaghayi, G. Kortsarz, M. R. Salavatipour. Approximation algorithms for node-weighted buy-at-bulk networks. In *Procc of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2007, pages 1265–1274.