CS 684: Approximation and Network Algorithms	11/07/01
Access network design #1	Scribe: Gavin Hurley

1 Basic definitions.

In the access network design problem, we are given a graph $G = (V, E)$, a specified node $a \in V$ and a subset $S \subseteq V$. The set S are demand points or customers; each wants to be connected to a. We will have to pay for the paths used to connect each $s \in S$ to a. We can either "rent" capacity on a edge and pay c_e per path using that edge (so if x_e paths use the edge e then we pay $x_e c_e$) or "buy" the edge, paying αc_e where α is a fixed integer.

The goal is, given $G = (V, E)$, $a \in V$, $S \subseteq V$ and $c_e \geq 0\ \forall e \in E$ and α a fixed integer, to find a path from each $s \in S$ to a that minimizes the cost, where the cost of a solution is given by

$$\sum_e \min(\alpha, x_e) c_e$$

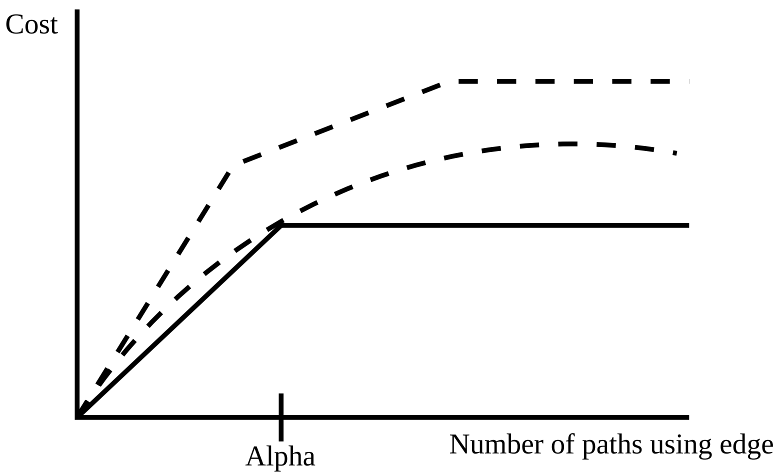

Figure 1: Different possible cost structures

In Figure 1 we give some examples of how the amount paid for an edge could vary as a function of the number of paths using the edge. The solid line corresponds to the case outlined above; there is a linear increase up to the time when α paths use an edge (the "renting" phase) and after that the cost remains constant (as we have "bought" the edge).

Although we will not give details of this, the method we will describe below to solve this problem can be applied to more general cost structures. Two examples of such generalized cost structures are given in Figure 1 in dashed lines. Two things are required for this generalization to work. The cost functions must be concave (this corresponds to an economies-of-scale type condition) and they must be the "same shape" for each edge.

As the solution we will give will be an approximation algorithm, we should note that this problem is NP-hard. To see this, consider the case when $\alpha = 1$. For this we ask for a minimum cost subgraph of G such that it contains S and a and is connected. This is the Steiner tree problem, which is NP-hard.

2 Solution structure

It is natural to think of a solution to this problem as being made up of two parts: edges we buy and edges we rent. For definiteness, suppose that we only buy an edge if strictly more than α paths use it. We can make the following statement about the edges we should buy.

Lemma 2.1 *The edges bought by an optimal solution* (OPT) *to the problem will form a connected tree containing a.*
Proof:

Once an edge is bought any number of paths can use it. Thus, OPT will not buy a cycle as at least one of the edges is not necessary (we could route the paths using that edge around the other edges in the cycle).

Thus the edges that OPT buys will form a forest. We now show that they are in fact a single tree containing a. Suppose not, namely that an edge e is bought which is not part of the component containing a. We bought the edge, so it must have strictly more than α paths on it. Each of these paths must rent some edges to reach the component containing a. But the cost of our solution can only improve is we choose the cheapest of these paths, buy the edges on it and route all the paths using e along these edges.

$\square$

Thus we can think of our solution as containing a tree consisting of bought edges. Each node $s \in S$ then rents along the shortest path to the bought edges.

At a high-level we should therefore solve the problem as follows: Firstly, we should find suitable "gathering points" at which paths from α or more $s \in S$ can join up. We then place a Steiner tree on those gathering points and a.

3 Finding Gathering Points

We will find (approximately) suitable gathering points using an application of the facility location problem. Recall that in this problem we are given a graph, with a subset of the nodes corresponding to a set of clients and a set of potential facilities. Each facility can be opened at a cost of f_c. Further, each client must be connected to a facility and there is a cost c_{ij} associated with assigning client i to facility j. We assume that the c_{ij}'s form a metric. In a previous lecture we gave a 3-approximation algorithm for this problem.

Note that the gathering problem we wish to solve is very similar. The set of customers S can be viewed as the clients. We will consider every node $v \in V$ as a potential facility. There is no cost for opening a facility but we require that each open facility serves $\geq \alpha$ clients. For a client i and a facility j the cost of assigning i to j is simply the length of the minimum cost path from i to j. This clearly forms a metric.

We will solve an approximate version of the gathering problem. Specifically, we will solve a facility location problem such that the cost of the solution C satisfies $C \leq 6\text{OPT}$ where OPT here corresponds to the cost of an optimal solution to the gathering problem. The solution will also be approximate in the sense that it will only guarantee that at least $\alpha/3$ paths are gathered at each facility (gathering point).[1]

In order to be able to apply a facility location algorithm, we must invent a facility cost for each of the potential facilities (here all of V). For a node $i \in V$ let K_i be the set of α closest clients (elements of S). Then define

$$f_i = \sum_{j \in K_i} c_{ij}$$

Now we simply run our approximation algorithm for facility location. As it is an approximation algorithm, we cannot be sure that each client is assigned to the facility closest to it. However, a simple post-processing layer (disconnect all the clients, keep the same facilities and assign each client to its closest facility) will fix this and so from now on we assume that each client is indeed assigned to the facility closest to it.

Lemma 3.1 *The minimal cost of the facility location problem ≤ 2 times minimal cost of the original gathering problem.*

Proof:

The optimal solution to the gathering problem clearly also gives a solution to the facility location problem. The assignments costs for both these solutions will be the same and so the only difference in cost for these will be the extra charge for facilities in the facility location problem.

However, this will not be too much extra. Consider a facility i which is open in the optimal solution to gathering problem. It must serve $\geq \alpha$ clients. These must clearly be at least as far away as the α closest clients. Thus i's clients connection costs are $\geq f_i$.

Thus the extra cost due to the charges to open facilities is at most the total assignment costs and the result follows.

□

This result, combined with the fact that we are using a 3-approximation algorithm for facility location, gives us our first guarantee about our approximate solution to the gathering problem, namely that it has cost no more than six times the optimal.

[1] We have given specific numbers here. However, if you have an r-approximation algorithm for facility location and wish to guarantee that at least α/q paths are gathered at each facility then the same algorithm and analysis below shows that this can be done with a cost not greater than $r\frac{q+1}{q-1}\text{OPT}$. In our case $r = 3$ (the algorithm we gave for facility location in a previous lecture is a 3-approximation) and $q = 3$.

The following lemma will guarantee the other claim, namely that each gathering point will serve at least $\alpha/3$ clients.

Lemma 3.2 *If facility i is open in the facility location solution and it serves $< \alpha/3$ clients, then closing i and reassigning its clients will improve the cost.*

Proof:

Client i serves $< \alpha/3$ clients. Thus, there exists a client j among the $\alpha/3$ closest to i which is being served by some other facility i', say. As we have assumed that each client is served by its closest facility we have $c_{ji'} \leq c_{ji}$.

Consider the change in cost arising from closing i and reassigning all its clients to i'. It will be $-f_i$ (we no longer need to pay to open i) plus the increase in assignment costs.

Client j is among the $\alpha/3$ closest clients to i. Thus there are at least $2\alpha/3$ among the α closest clients to i which are at least c_{ij} away from i. Thus we have

$$f_i > \frac{2\alpha}{3} c_{ij}$$

and hence

$$c_{ij} < \frac{3f_i}{2\alpha}$$

Each of i's clients being reassigned to i' will have to pay at most $c_{ii'}$ extra. Note

$$c_{ii'} \leq c_{ji} + c_{ji'} \leq 2c_{ij} < \frac{3}{\alpha} f_i$$

using the bound on c_{ij} above.

We know that strictly less than $\alpha/3$ clients will have to pay this extra assignment costs and so the overall change in the cost of the solution, Δ satisfies

$$\Delta < -f_i + \frac{\alpha}{3} c_{ii'} < -f_i + \frac{\alpha}{3} \frac{3f_i}{\alpha} = 0$$

i.e. the overall cost does not increase.

□

Thus to complete our solution we will run this algorithm to find gathering points and then place an (approximate) Steiner tree on these points. The full details and analysis of this second phase are in the next lecture.