COM S 783 / OR&IE 634 Approximation Algorithms Fall Semester, 2003

Lecture 17: Nov 5, 2003

Lecturer: Yuval Rabani *Scribe: Xin Qi*

1 k-MST Continued

Last time, we have obtained two trees T_1 of k_1 nodes, and T_2 of k_2 nodes, where $k_1 \leq k \leq k_2$. Now we want to combine them to get a tree of k nodes and bounded cost. We have also got two possibilities μ_1 and μ_2 of choosing between T_1 and T_2, such that the expected cost and number of nodes are both good.

$$\begin{aligned} E[cost] &= \mu_1 \cdot cost(T_1) + \mu_2 \cdot cost(T_2) \leq 2 \cdot OPT_{k-MST} \\ E[\#nodes] &= \mu_1 \cdot k_1 + \mu_2 \cdot k_2 = k \end{aligned}$$

However, what we want is a specific tree of k or more nodes, having good expected cost.

If $\mu_2 \geq \frac{1}{2}$, we can directly take T_2. The cost of T_2 is no more than twice the expected cost, and hence no more than $4 \cdot OPT_{k-MST}$.

The difficulty comes when μ_2 is small. We will first take T_1. Then we take part of T_2, which is a sub-tree of size $k - k_1$, and attach it to the root of T_1 with a shortest path. We hope that the cost of the sub-tree is not too much more than the contribution of T_2 to the expected cost.

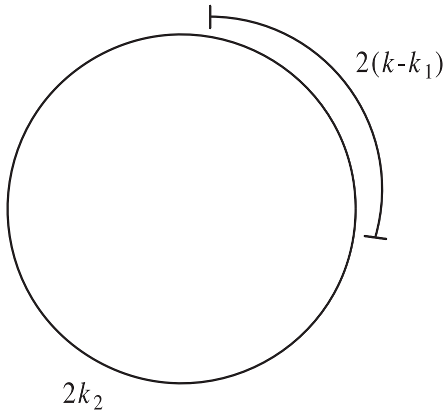
How to take the sub-tree from T_2? First, we double the edges in T_2, and get a (not necessarily simple) cycle with twice the cost of T_2. The length of the cycle is $2k_2$. We can pick a segment of length $2(k - k_1)$ on that cycle to form a sub-tree. If we take a random choice, the expected cost is no more than

$$\frac{k - k_1}{k_2} \times 2cost[T_2]$$

In fact, we can also do this deterministically by choosing the best segment of length $2(k-k_1)$, which will guarantee to have at least $k-k_1$ edges and the cost is no more than $\frac{k-k_1}{k_2} \times 2cost[T_2]$.

What about the cost of the shortest path? To restrict its cost, we need to do preprocessing at the beginning of the algorithm, even before the Primal-Dual part. The idea is to throw away nodes which are too far from the root. In fact, we only have at most $n - k + 1$ possible distance thresholds, which includes the nearest k, $k + 1$, $\ldots$, n nodes separately. We try all these thresholds and pick the best result. We therefore restrict that every node can be connected to the root with a path of cost no more than OPT_{k-MST}.

The above method gives an approximation ratio of 5. With more careful analysis, we can reduce this factor further down to 2.



2 Facility Location - Overview

Basic Problem The Facility Location problem has as input a set of potential facilities F, a set of clients C, costs of opening facilities $f : F \to \mathbb{Q}^+$, and the connection costs $c : F \times C \to \mathbb{Q}^+$. Note that the set of facilities F and the set of clients C are not necessarily disjoint, they might even be identical. The goal of the problem is to find a subset $F' \subset F$, with the following objective function:

$$\min \sum_{i \in F'} f(i) + \sum_{j \in C} \min_{i \in F'} c(i,j)$$

The above general case of Facility Location is at least as hard as Set Cover. However, with the common assumption that c is a metric (or induced metric) on $F \cup C$, we can expect much better result than Set Cover. We will keep this assumption of metric in the following discussion of Facility Location.

Variations of Facility Location Facility Location has a number of variations. Some of them are listed as follows:

- k-Median[1].

  In the k-Median problem, facilities have opening cost of 0

  $$f(i) = 0 \quad \forall i \in F$$

  However, we have the constraint that at most k facilities can be open. Obviously, opening exactly k facilities will be the best choice. The objective function is the same as in Facility Location, without opening cost of course.

Remark The relation between k-Median and Facility Location is quite similar to that between k-MST and Price Collecting Steiner Tree.

- k-Facility Location.

  As in k-Median, at most k facilities can be open, but here the opening costs might not be zero. So we possibly open fewer than k facilities.

- Clients have demand $d : C \to \mathbb{Q}^+$.

  Every clients j has a demand $d(j)$, so the connection cost of client j becomes:

$$d(i) \cdot \min_{i \in F'} c(i,j)$$

  Ordinary Facility Location can be viewed as a special case where each client has a demand of 1.

- Facilities have capacities $u : F \to \mathbb{Q}^+$.

  The capacity of a facility restricts the number of clients or the amount of demands it can serve. In this case, some clients might have to connect to facilities that are far away, because of congestion.

  There are two kinds of capacities:

  - Soft Capacities:

    We can open one facility for several times, and pay opening cost each time.
  - Hard Capacities:[2]

    Every facility can be opened only once.

- Fault Tolerant Version

  In this case, a client might want to connect to more than one facilities to have some kind of "backup". The fault tolerant requirements can be depicted as:

  - Every client need to connect to the first facility, and has to pay that cost.
  - The remaining backup facilities each have a probability to be connected, and such probabilities serve as the weights of costs.

  There is also a simple version of this problem, in which every client connects to a number of different facilities.

In this lecture, we will still focus on the original version of Facility Location.

Now we consider the Integer Programming formulation of this problem. We will use two kinds of 0/1 variables:

- $y_i = 1$ represents that facility i is open, and $y_i = 0$ represents that facility i is not open.

- $x_{ij} = 1$ represents that client j connects to facility i, and $x_{ij} = 0$ represents that client j does not connect to facility i.

The IP is described as follows:

$$\begin{array}{lll}
\min & \sum_{i \in F} f(i) \cdot y_i + \sum_{i \in F} \sum_{j \in C} c(i,j) \cdot x_{ij} & \\
s.t. & \sum_{i \in F} x_{ij} \geq 1 & \forall j \in C \\
 & x_{ij} \leq y_i & \forall i \in F, \forall j \in C \\
 & x_{ij} \in \{0,1\} & \forall i \in F, \forall j \in C \\
 & y_i \in \{0,1\} & \forall i \in F
\end{array}$$

As before, we can relax this IP formulation to a LP, by replacing the integrality constraints to $x, y \geq 0$.

3 Facility Location - Randomized Rounding[3]

The first method that we are considering is Randomized Rounding. However, if we are not careful enough, we might end up with logarithmic guarantee, just as Set Cover, because there is no guarantee to have integral or half-integral solution to this LP, as that in LP for Vertex Cover. We will adopt an idea called "filtering" [4] to obtain a better result.

In optimal solutions, we have

$$\sum_i x_{ij} = 1$$

So we can interpret x_{ij} as the probability that client j connects to facility i. For a specific client j, $\sum_{i \in F} c(i,j) x_{ij}$ is the expected cost of connecting j under distribution x_{ij}.

The idea of "filtering" is to restrict connection cost by only allowing j to connect to the facilities within twice the expectation.

Define

$$r_j = 2 \sum_i c(i,j) x_{ij}$$

which is the upper bound of the distance that client j is allowed to connect to.

Take $c(i, \cdot)$ as the random variable, then according to Markov inequality

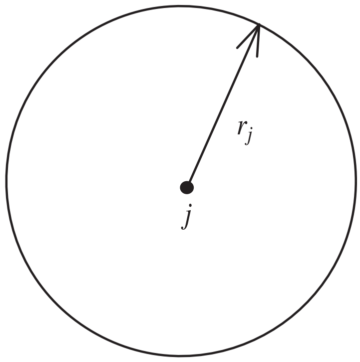
$$X \geq 0 \Rightarrow Pr[X > \alpha E[X]] < \frac{1}{\alpha}$$

we have

$$\forall i \in C \,.\, Pr[c(i,j) > r_j] < \frac{1}{2}$$

So

$$\sum_{i | c(i,j) > r_j} x_{ij} < \frac{1}{2}$$



Draw a ball around each client j, with radius r_j. The total weight x_{ij} of facilities inside the region $\geq \frac{1}{2}$. Remember that $x_{ij} \leq y_i$, so total y_i in the region $\geq \frac{1}{2}$. So we just find the cheapest facility in the region and open it.

It seems to be a 2-approximation algorithm, but it is not, because regions might be overlapping, and therefore a facility might be opened again and again.

To avoid such a problem, we should try to pick a set of non-overlapping balls, but for those clients not chosen, it is possible that no facility is open in its ball. We need to figure out a way to bound the total connection cost for these clients.

Finally, we come up with the following method:

- Sort clients in the ascending order of radii $r_j = \sum_{i \in F} c(i,j) x_{ij}$.
- Consider a lexicographically first maximal non-overlapping sequence of balls, and open a facility in each ball.

In this algorithm, the total facility cost is no more than $2 \sum_{i \in F} f(i) y_i$, since there are no overlapping balls.

Now, let us take a look at a client j' which has no open facilities in its ball. According to our algorithm, the only reason that a ball is not chose is that it overlaps another ball with smaller or equal radius. Suppose it overlaps the ball of client j and $r_j \leq r_{j'}$, and client j chooses facility i to open.

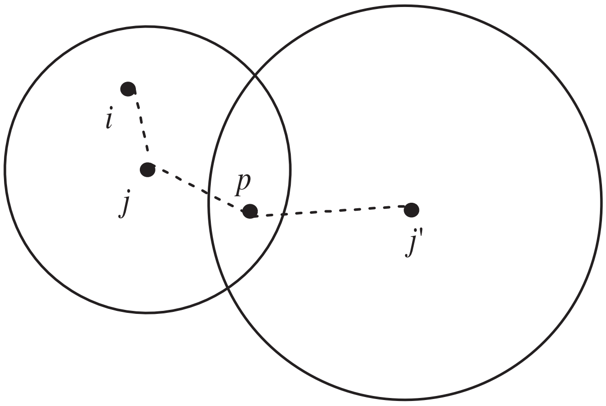
Take an arbitrary point p within the overlapping region of the two balls, then based on the fact that the distance is a metric, we have:

$$c(j', i) \leq r_{j'} + r_j + r_j \leq 3 r_{j'} \leq 6 \sum_{i \in F} c(i, j') x_{ij'}$$

So the total connection cost is at most 6 times of that in optimal solution.

At the beginning of the discussion, we choose $r_j = \alpha \sum_i c(i,j) x_{ij}$ and let $\alpha = 2$. The analysis above shows that it leads to approximation ratio 2 for facility cost, and 6 for connection cost. The difference between these two ratios suggests that we change α for a better balance between them.

We can choose $\alpha = \frac{4}{3}$, and obtain a 4-approximation algorithm.



References

[1] M. Charikar, S. Guha, E. Tardos, and D. Shmoys. A Constant-Factor Approximation Algorithm for the k-Median Problem. STOC '99.

[2] M. Pal, E. Tardos, and T. Wexler. Facility location with nonuniform hard capacities. FOCS '01.

[3] D. Shmoys, E. Tardos, and K. Aardal. Approximation algorithm for facility location problems. STOC '97.

[4] J.-H. Lin and J. S. Vitter. ϵ-approximations with minimum packing constraints violation. STOC '92.