College of Computer & Information Science	Fall 2009
Northeastern University	13 October 2009
CS7880: Algorithmic Power Tools	Scribe: San Tan and Bishal Thapa

Lecture Outline:

- Uncapacitated Facility Location:
  - NP-hardness and randomized rounding
  - Linear Programming Rounding Approach : Constant (6)-Approximation

In this lecture, we first introduce the class of facility location problems, and then focus on the uncapacitated facility location problem (UFL). We show that UFL is NP-hard. We then consider a randomized rounding approach that yields a logarithmic-approximation to the problem. Finally, we present a deterministic rounding approach that yields a constant-factor approximation.

1 (metric) Uncapacitated Facility Location

Problem 1. We are given a set V of demand locations (or clients), a set F of potential facility locations, a cost function $c : (V \bigcup F) \times (V \bigcup F) \rightarrow Q^+$, cost f_j of opening facility at $j \in F$, and demand d_i at client $i \in V$. The goal is to determine locations $S \subseteq F$ where to open facilities and $\sigma : V \rightarrow S$ to minimize $\sum_{j \in S} f_j + \sum_{i \in V} c_{i\sigma_i} d_i$.

The notation is summarized in the following table.

V	Set of demand points (clients)
F	Set of possible facility locations
d_i	Demand for the service at demand point i
c	Metric on $V \cup F$
c_{ij}	Cost for assigning $i \in V$ to facility $j \in F$
σ	Assignment function, $\sigma : V \rightarrow F$
f_j	Cost of opening a facility at location $j \in F$
FacilityCost	$= \sum_{j \in F} f_j$
ServiceCost	$= \sum_{i \in V} d_i \times c_{i\sigma(i)}$
TotalCost	= FacilityCost + ServiceCost

There are different variants of problem 1

- Metric or non-metric: metric means the cost function of problem 1 satisfies symmetry and triangle inequality,
$$c_{ij} = c_{ji};\ c_{ij} \leq c_{ik} + c_{kj};\ c_{ii} = 0.$$
- Capacity of facility: There is a bound on the total demand (or number of clients) that a facility can serve

- There is a bound on the number of facilities (or cost of facilities) that can be opened
- Facility costs

In this class, we will focus on the uncapacitated metric facility location (UFL). Example applications of facility locations include

- Hub and spoke scheduling;
- Locating concentrators in a routing network;
- Locating servers in a content-delivery network.

Theorem 1. *UFL is NP-Hard.*

Proof. Reduction from set cover. Use notations defined in set cover problem and problem 1. Construct a UFL problem as follows from set cover.

- $\mathcal{U} \to V$,
- $S \to F$,
- $c_{ij} = \begin{cases} 1 & \text{if } e_i \in s_j; \\ 3 & \text{otherwise;} \end{cases}$
- $f_j = 1$,
- $d_i = 1$.

It is easy to check that the cost function is a metric. We now argue that there exists a set cover of cost C if and only if there exists a facility location solution of cost at most $C + n$.

One direction is trivial. If there is a set cover of cost C, then the sets form the facilities, with a total cost of $C + n$.

We now consider the other direction. Suppose we have a facility location solution of cost $C + n$.

- case 1: The service cost is equal to n. The selected facilities yield the collection of sets of cost C for the set cover problem.
- case 2: The service cost is greater than n. For each client i that is paying service cost at least 3, add a facility j such that $j \notin \{\text{selected facilities}\}$ and $c_{ij} = 1$. Since $1 + c_{ij} \leq 3$, the total cost doesn't increase. Then go to case 1.

□

A possible approximation algorithm for UFL is to reduce it to set cover problem, then solve it using LP-rounding. For each facility, $(2^{|V|} - 1)$ sets are generated, with cost of $c = x_{js}, s \subseteq V$, in which $x_{js} = f_j + \sum_{i \in s} c_{ij}$. However, this approach has exponential-complexity. It's not hard to reduce the solution to polynomial time by identifying only those sets that could possibly be in an optimal solution.

2 Randomized rounding approach for UFL

Consider the following integer linear program for UFL. For any $j \in F$, define

$$y_j = \begin{cases} 1 & \text{if facility is opened at } j, \\ 0 & \text{otherwise.} \end{cases}$$

Obviously clients should be assigned to nearest facility. For $i \in V$, $\sigma(i) =$ nearest j such that $y_j = 1$. Define

$$x_{ij} = \begin{cases} 1 & \text{if } i \text{ is served by } j, \\ 0 & \text{otherwise.} \end{cases}$$

Then, the integer linear program is the following.

$$\begin{array}{lll} min & \sum_{j \in F} f_j \times y_j + \sum_{i \in V} x_{ij} \times c_{ij} & \\ s.t. & x_{ij} \le y_j & \forall i \in V, j \in F \\ & \sum_j x_{ij} \ge 1 & \forall i \in V \\ & x_{ij}, y_j \in \{0,1\} & \forall i \in V, j \in F \end{array}$$

By relaxing the integrality constraints we get the linear program:

$$\begin{array}{lll} min & \sum_{j \in F} f_j \times y_j + \sum_{i \in V} x_{ij} \times c_{ij} & \\ s.t. & x_{ij} \le y_j & \forall i \in V, j \in F \\ & \sum_j x_{ij} \ge 1 & \forall i \in V \\ & x_{ij}, y_j \ge 0 & \forall i \in V, j \in F \end{array}$$

Suppose that an optimal solution for the above LP is $\{x^*\}$ and $\{y^*\}$. Here is a randomized rounding approach.

- open facility at j with probability y_j^*. so

$$E[\text{facility cost so far}] = \sum_j y_j^* f_j.$$

- if distance of i to nearest open facility is at most $r_i = 2\sum_j x_{ij}^* c_{ij}$, then assign i to nearest open facility; otherwise don't assign. (Note that if we do not have a filter like this, the client costs could be arbitrarily high.)

Lemma 1. $\forall i \in V, \Pr[i \textit{ is assigned}] \ge 1 - \frac{1}{\sqrt{e}}$.

Proof. Supose B is a ball centering i with radius $2\sum_j x_{ij}^* c_{ij}$. We have

$$\begin{aligned}\sum_{j\in B} y_j^* &\geq \sum_{j\in B} x_{ij}^* \\ &\geq \frac{1}{2},\end{aligned}$$

because

$$\begin{aligned}\sum_{j\notin B} x_{ij}^* &= \frac{r_i \sum_{j\notin B} x_{ij}^*}{r_i} \\ &\leq \frac{\sum_j x_{ij}^* c_{ij}}{r_i} \\ &= \frac{\frac{1}{2} r_i}{r_i} \\ &= \frac{1}{2}.\end{aligned}$$

So

$$\begin{aligned}\Pr[i \text{ is assigned}] &= 1 - \prod_{j\in B}(1 - y_j^*) \\ &\geq 1 - (1 - \frac{1}{2|F|})^{|F|} \\ &\geq 1 - \frac{1}{\sqrt{e}}.\end{aligned}$$

□

Then, if we repeat the above round step t times and combine the rounding result in a way as we did in set cover, we have

$$\forall \text{ client } i, \Pr[\text{client } i \text{ is not assigned}] \leq (\frac{1}{\sqrt{e}})^t.$$

Set $t = \log_{\sqrt{e}}(4n) = O(\log n)$, and obtain

$$\Pr[\text{client } i \text{ is not assigned}] \leq \frac{1}{4n};$$

$$\Pr[\text{some client is not assigned}] \leq \frac{1}{4};$$

$$E[\text{cost of solution}] \leq 2t \cdot OPT_{LP}.$$

By Markov's inequality, $\Pr[\text{cost } \geq 8t \cdot OPT_{LP}] \leq \frac{1}{4}$.

Putting these together with $t = \log_{\sqrt{e}}(4n) = O(\log n)$, we obtain that with probability at least 1/2, we have a feasible facility location solution that has cost $O(\log n) OPT_{LP}$. As we did for the set cover problem, we can improve our success probability by repeating the above process until we are guaranteed a feasible solution with the desired approximation ratio.

3 A constant-approximation deterministic rounding algorithm

The logarithmic-approximation ratio of the randomized rounding approach is primarily due to the fact that in each iteration of the algorithm, the probability that a client is served by a "nearby" facility is only a constant fraction (less than 1). Note, however, that the randomized algorithm does not make use of the metric property anywhere. Can we do better by a more careful selection of facilities and processing of the clients so as to obtain a constant-factor approximation?

Algorithm 1 is a deterministic rounding algorithm that achieves a solution of cost within a factor of 6 to the optimal solution.

Algorithm 1: Rounding the fractional LP solution

1. Solve the fractional LP. Let $(< x^*_{ij} >, < y^*_j >)$ be the optimum fractional solution.
2. **for** *each client* $i \in V$ **do**
   - define:
     - $r_i = 2 \sum_j x^*_{ij} \times c_{ij}$
     - $B_i = \{j \in F | c_{ij} \leq r_i\}$
3. Sort r_i in the non-decreasing order. WLOG, assume $r_1 \leq r_2 \leq \cdots \leq r_n$.
4. Let $V' := \{r_1, r_2, \cdots r_n\}$ s.t. the index-set of V' represents the clients. Call the set $I[V']$.
5. Let $i := 1$, $F' := \emptyset$.
6. **for** *client* $i \in I[V']$ **do**
   - let $j \in B_i$ with $x^*_{ij} > 0$ and f_j minimum. Set $F' = F' \cup \{j\}$.
   - let $N_i = \{r_k \in V' | B_i \cap B_k \neq \emptyset\}$.
   - **for** *each client* k *s.t.* $r_k \in N_i$ **do**
     - Set $\sigma(k) \leftarrow j$.
   - **Note**: $i \in N_i$ vacuously.
   - Set $V' = V' \setminus N_i$ while keeping the sorted order among the remaining elements as before.
   - Re-label the elements in the new V' starting from index 1. (new $I[V'] \subset$ old $I[V']$.)
   - If $V' = \emptyset$, break.
7. Output F' and σ.

Theorem 2. *The running time for Algorithm 1 is polynomial in the size of input.*

Proof. It is clear that the algorithm terminates. The algorithm consists of three processes: LP solving, Sorting and Comparing. We know that the linear programming can be solved in polynomial time using the ellipsoid or interior point methods. Sorting can be done very efficiently as well, using a merge sort or heap sort at the cost of $O(n \log n)$ running time. Finally, the comparing procedure involved in step 6 of Algorithm 1 can also be completed in polynomial-time. □

Claim 1. $\forall i \in V, \sum_{j \in B_i} y^*_j \geq \frac{1}{2}$

Proof. (Proof by Contradiction) Let LP_i denote the service cost of client $i \in V$ in the LP solution. So, $LP_i = \sum_j x^*_{ij} \times c_{ij}$. Assume that the claim is not true. Then, $\sum_{j \in B_i} y^*_j < \frac{1}{2}$ [hypothesis].
We have,

$$LP_i \geq \sum_{j \notin B_i} x^*_{ij} \times c_{ij}$$

because, the right hand side doesn't include j's inside the ball.

$$\geq \sum_{j \notin B_i} x^*_{ij} \times r_i \left(= r_i \times \sum_{j \notin B_i} x^*_{ij} \right)$$

because, $c_{ij} > r_i \ \forall j \notin B_i$.

$$> \frac{1}{2} \times r_i$$

because of the hypothesis and the fact that,

$$\sum_{j \in B_i} x^*_{ij} \leq \sum_{j \in B_i} y^*_j$$

thus, $\sum_{j \in B_i} x^*_{ij} < \frac{1}{2}$, which further implies,

$$\sum_{j \notin B_i} x^*_{ij} > \frac{1}{2}$$

Hence, we get

$$LP_i > \frac{1}{2} \times r_i = LP_i \ (contradiction)$$

because of the definition of LP_i. □

Lemma 2. *Let LP_F denote the facility cost of the LP solution. Then, $\sum_{j \in F'} f_j \leq 2 \times LP_F$.*

Proof. We know $LP_F = \sum_{j \in F} f_j \times y^*_j$. For each $j \in F'$, let i_j denote the client in $I[V']$ that was considered, when j was added to F'. Consider a ball B_{i_j}. Then,

$$\sum_{l \in B_{i_j}} y^*_l \times f_l \geq \sum_{l \in B_{i_j}} y^*_l \times f_j \left(= f_j \times \sum_{l \in B_{i_j}} y^*_l \right)$$

because, we pick $j \in B_{i_j}$ s.t. f_j is minimum (refer to Step 6, Algorithm 1). Hence,

$$\sum_{l \in B_{i_j}} y^*_l \times f_l \geq \frac{1}{2} \times f_j$$

by previous equation and Claim 1. Adding over all $j \in F'$,

$$\frac{1}{2} \times \sum_{j \in F'} f_j \leq \sum_{l \in B_{i_j}} y^*_l \times f_l \leq \sum_{j \in F} f_j \times y^*_j$$

because, all the facilities are selected from non-overlapping balls. Hence,

$$\sum_{j \in F'} f_j \leq 2 \times LP_F$$

□

Lemma 3. *Let LP_i be the service cost of client $i \in V$ in the LP solution. Then, $\forall i \in V$, $c_{i\sigma(i)} \leq 6 \times LP_i$.*

Proof. Consider two possible cases for a client i:

1. Client i is such that some $j \in B_i$ was selected in step 6 of the algorithm 1. Then,

$$c_{i\sigma(i)} \leq r_i = 2 \times LP_i$$

   by definition of B_i.

2. Client i is removed at some iteration in Step 6, due to the overlap of B_i with B_k for some k. Then, $r_i \geq r_k$ and there exists $l \in B_i \cap B_k$. Assuming, we included $j \in B_k$ in F', we have $\sigma(i) = \sigma(k) = j$. Therefore,

$$c_{ij} \leq c_{il} + c_{lk} + c_{kj} \leq r_i + 2 \times r_k \leq 3 \times r_i,$$

   by triangle inequality, leading to $c_{ij} \leq 6 \times LP_i$.

In either case, $c_{i\sigma(i)} \leq 6 \times LP_i$. □

Let LP_S denote the service cost of the LP solution. Then, Lemma 3 gives us,

$$\sum_{i \in V} c_{i\sigma(i)} \leq 6 \times LP_S$$

.

Theorem 3. $\textsc{TotalCost} \leq 6 \times (LP_F + LP_S)$

Proof. This follows directly from Lemmas 2 and 3 and the definition of $\textsc{TotalCost}$. Since $\textsc{ServiceCost} \leq 6 \times LP_S$ and $\textsc{FacilityCost} \leq 2 \times LP_F$, we have $\textsc{TotalCost} \leq 6 \times (LP_F + LP_S)$. □