Scribe: Bishal Thapa

Lecture Outline:

- Uncapacitated Facility Location (contd..)
  - Analysis of Combinatorial "Greedy" Approach : Constant (6)-Approximation
  - Analysis of a Combinatorial Greedy Algorithm using Dual Fitting: Constant (3)-Approximation

In this lecture, we first complete the analysis of "Greedy" approach we introduced in the last lecture, that gives a constant (6)-approximation polynomial-time solution for the Uncapacitated Facility Location problem. Then, we present a primal-dual approach of solving the problem, that yields a better approximation solution of cost within a factor of 3 of the optimal. The latter has a polynomial-time complexity as well.

1 (metric) Uncapacitated Facility Location

We previously described the Uncapacitated Facility Location problem as follows:

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 $j \in F$
$Facility\,Cost$	$= \sum_{j \in F} f_j$
$Service\,Cost$	$= \sum_{i \in V} d_i \times c_{i\sigma(i)}$
$Total\,Cost$	$= Facility\,Cost + Service\,Cost$

Our goal is to minimize the total cost. Notice that we are considering the uncapacitated metric facility problem. So, the *metric* service costs satisfy this property (triangle inequality):

$$c_{ij} + c_{jk} + c_{kl} \geq c_{il} \;\forall\, i, j, k, l \in V \cup F$$

1.1 Combinatorial "Greedy" Approach

This algorithm takes a greedy approach on the UNCAPACITATED FACILITY LOCATION problem. The algorithm doesn't make an explicit use of primal-dual schema however, it substantially derives its intuition from the dual fitting approach. Algorithm 1 runs on polynomial-time and achieves a solution of cost within a factor of 6 of the optimal solution.

Algorithm 1: Greedy Approach

1. **for** *each facility* $j \in F$ **do**
   define:
   $r_j = r, r > 0$ at which $\sum_{i:c_{ij} \leq r}(r - c_{ij}) = f_j$
   $B_j = \{i \in V | c_{ij} \leq r_j\}$
2. Sort r_j in the non-decreasing order. WLOG, assume $r_1 \leq r_2 \leq \cdots \leq r_m$.
3. Let $F' := \{r_1, r_2, \cdots r_m\}$ s.t. the index-set of F' represents the facilities. Call the set $I[F']$.
4. Let $j := 1, and\ X := \emptyset$.
5. **for** *facility* $j \in I[F']$ **do**
   Set $X = X \cup \{j\}$.
   let $M_j = \{r_l \in F' | B_j \cap B_l \neq \emptyset\}$.
   Set $F' = F' \setminus M_j$ with keeping the sorted order among the remaining elements as before.
   Note: $j \in M_j$ vacuously.
   Re-label the elements in the new F' starting from index 1. (new $I[F'] \subset$ old $I[F']$.)
   If $F' = \emptyset$, break.
6. Assign each client to the nearest facility in X. Let $\sigma : V \to X$ denote the mapping.
7. Output X and σ.

Theorem 1. *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: (a) Finding the associated radius for each facility [step 1], (b) Sorting [step 2] and (c) Comparing [finding intersecting sets, step 5]. For each facility $j \in F$, (a) the associated radius can be computed in $O(nlgn)$ time. This is essentially a weighted selection problem, and given the cost f_j are sorted, this can be done in $\Theta(n)$ time. (b) Sorting n radii of step [2] can be accomplished in $O(nlgn)$ time. And finally, (c) each iteration of step [5] of comparing and finding intersecting sets doesn't exceed linear time [$O(n)$]. Hence, (c) can be implemented for a total of $O(n^2)$ time. □

Now, consider the cost of the "greedy" solution for a client $i \in V$, denoted *cost*(i).

$$cost(i) = \begin{cases} c_{i\sigma(i)} + (r_j - c_{ij}) & \text{if } i \in B_j,\ j \in X \\ c_{i\sigma(i)} & \text{if } i \notin B_j,\ \forall j \in X \end{cases}$$

Lemma 1.

$$cost(i) \leq \begin{cases} r_j & if\ i \in B_j,\ j \in X \\ c_{i\sigma(i)} & otherwise \end{cases}$$

Proof. Recall the definition of *cost*(i). It is sufficient to show that for a client i,

$$cost(i) \leq r_j \quad if\ \ i \in B_j\ \ and,\ \ j \in X.$$

This is straightforward if $\sigma(i) = j$, since then,

$$cost(i) = c_{ij} + (r_j - c_{ij}) = r_j.$$

Assume otherwise. Let $\sigma(i) = k$ s.t. $k \in X$. Then,

$$cost(i) = c_{ik} + (r_j - c_{ij}).$$

This is because, given $i \in B_j$ and $j \in X$, $i \notin B_k \;\; \forall k \in X,\; k \neq j$. Now,

$$\sigma(i) = k \;\Rightarrow c_{ik} \leq c_{ij}.$$

Hence,

$$cost(i) = c_{ik} + (r_j - c_{ij}) \leq c_{ij} + (r_j - c_{ij}) = r_j$$

□

Consider a client $i \in V$. Let $OPT(i)$ be the optimal cost charged to the client i in the optimal solution, OPT. Also, assume Z^* be the facility-set opened in OPT. Then,

$$OPT(i) = c_{i\sigma^*(i)} + \sum_{j \in Z^* | i \in B_j} (r_j - c_{ij})$$

Lemma 2. *Let $i \in V$. Suppose $l = \sigma^*(i)$, the nearest facility location to i. Then,*

$$OPT(i) \geq \begin{cases} r_l & \text{if } i \in B_l \\ c_{il} & \text{if } i \notin B_l \end{cases}$$

Proof. Consider each of the possible cases:

1. Assume $i \in B_l$. Then using hypothesis and by the definition of $OPT(i)$,

$$OPT(i) \geq c_{i\sigma^*(i)} + (r_l - c_{il}) = c_{il} + (r_l - c_{il}) = r_l.$$

2. Assume $i \notin B_l$. Then, the lemma follows directly from the definition of $OPT(i)$,

$$OPT(i) \geq c_{i\sigma^*(i)} = c_{il}.$$

□

Theorem 2. *$\forall i \in V$, $cost(i) \leq 6 \times OPT(i)$.*

Proof. Let us consider all possible cases :

1. Assume $i \in B_l$ and, $l \in X$. This implies,

$$cost(i) = c_{il} + (r_l - c_{il}) = r_l \leq OPT(i)$$

   from the definition of l, $cost(i)$ and Lemma 2.

2. Assume $i \in B_l$ but, $l \notin X$. This implies,

$$\exists j \in X, \;\; r_j \leq r_l \; s.t. \; B_l \cap B_j \neq \emptyset.$$

   Then there are two possibilities:

(a) $i \notin B_k$ for any $k \in X$. This implies, $cost(i) = c_{i\sigma(i)} \le c_{ij}$. But,

$$c_{ij} \le_1 c_{il} + c_{lj} \le_2 r_l + (r_l + r_j) \le_3 3 \times r_l \le_4 3 \times OPT(i).$$

The first inequality is because of the triangle inequality induced by the *metric c*. The second inequality is due to the fact that $B_j \cap B_l \neq \emptyset$. So, the distance between the centers of two balls is less than the sum of two radii. Third inequality is from the hypothesis and the last follows directly from Lemma 2.

(b) $i \in B_k$ for some $k \in X$. This implies, $cost(i) = c_{ij} + (r_k - c_{ik})$. Two possibilities:

- If $j = k$, then by the hypothesis, $cost(i) = c_{ij} + (r_j - c_{ij}) = r_j \le r_l$.

- If $j \neq k$, then $B_j \cap B_k = \emptyset$. This implies,

$$r_j + r_k \le_1 c_{jk} \le_2 c_{jl} + c_{il} + c_{ik} \le_3 r_j + r_l + c_{il} + c_{ik}$$

$$\implies r_k \le_4 r_l + c_{il} + c_{ik}$$

$$\implies (r_k - c_{ik}) \le_5 r_l + c_{il} \le_6 2 \times r_l.$$

The first inequality is because of the hypothesis, that $B_j \cap B_k = \emptyset$. The second inequality is due to triangle inequality induced by *metric c*. The third inequality is due to the fact that $\{i\} \subseteq B_k \cap B_l$. We get the fourth relation by subtracting r_j from both sides of the inequality. And, we get the final inequality by simply moving term from right to left and using the definition of a ball, B_l. Hence, the cost becomes:

$$cost(i) = c_{ij} + (r_k - c_{ik}) \le c_{ij} + 2 \times r_l \le_7 3 \times r_l + 2 \times r_l.$$

The inequality [7] simply follows from the same argument of case (a): $i \notin B_j$ for any $j \in X$.

In either of the two sub-cases, $cost(i) \le 5 \times r_l \le 5 \times OPT(i)$. [From Lemma 2]

3. Assume $i \notin B_l$. The theorem holds trivially for the case of $l \in X$, because, then

$$cost(i) = c_{il} \le OPT(i).\ [From\ Lemma\ 2]$$

Hence, assume $l \notin X$. Then, we know that $\exists j \in X$, $r_j \le r_l$, $s.t.\, B_j \cap B_l \neq \emptyset$. This implies,

(a) If $i \notin B_k$ for any $k \in X$, then

$$cost(i) \le_1 c_{ij} \le_2 c_{il} + c_{lj} \le_3 c_{il} + r_l + r_j \le_4 c_{il} + 2 \times r_l \le_5 3 \times c_{il}.$$

The first inequality is due to the hypothesis that implies, $cost(i) = c_{i\sigma(i)} \le c_{ij}$. The second inequality is due to the triangle inequality induced by *metric c*. The third inequality follows from the fact that $B_j \cap B_l \neq \emptyset$. The fourth and the fifth inequalities directly follow from the hypothesis, $r_j \le r_l$ and $i \notin B_l$.

(b) If $i \in B_k$ for some $k \in X$, then

- If $k = j$, $cost(i) = c_{i\sigma(i)} + (r_j - c_{ij}) \leq c_{ij} + (r_j - c_{ij}) = r_j \leq r_l \leq c_{il}$. This is because of the hypothesis that implies $cost(i) = c_{i\sigma(i)} \leq c_{ij}$ and $r_j \leq r_l$, $i \notin B_l$.
- If $k \neq j$, $cost(i) = c_{i\sigma(i)} + (r_k - c_{ik}) \leq c_{ij} + (r_k - c_{ik})$. Now, it is sufficient to show that $(r_k - c_{ik}) \leq c_{ij}$. If so, then

  $$cost(i) \leq 2 \times c_{ij} \leq_1 2 \times (3 \times c_{il}) = 6 \times c_{il}.$$

  The first inequality follows from the similar argument as that of the case $3(a)$.

  Claim 1. $(r_k - c_{ik}) \leq c_{ij}$.

  Proof. Since both $j, k \in X$, we know that $r_k \leq c_{jk}$. But, from triangle inequality, we have $c_{jk} \leq c_{ij} + c_{ik}$. Therefore, $r_k \leq c_{ij} + c_{ik}$. This implies $(r_k - c_{ik}) \leq c_{ij}$. □

In either case, $cost(i) \leq 6 \times c_{il} \leq 6 \times OPT(i)$. [From Lemma 2]

Therefore, $cost(i) \leq 6 \times OPT(i)$ for all $i \in V$.

□

Theorem 3. *Total Cost* $\leq 6 \times OPT$.

Proof. This follows directly from Theorem 2 and the fact that

- $\sum_i OPT(i) = OPT$.
- $\sum_i cost(i) = Total\,Cost$.

□

1.2 An alternative Analysis using dual-fitting

We now analyze a slight variant of Algorithm 1, using a dual-fitting approach. The difference is in step [5] of the algorithms. In Algorithm 1, we pick the smallest ball among all the facility balls, and remove the ones that intersect the chosen ball. In Algorithm 2, we pick the smallest ball among all the facility balls, and remove the ones whose twice the size contains the center (facility) of the chosen ball. It is clear that the latter removes more facilities from the search space compared to the former method. In the original paper of Mettu-Plaxton, the analysis of the algorithm 2 is quite similar to what we have done for Algorithm 1. The approximation factor for this method is 3, and the algorithm 2 runs in polynomial time of the input.

The integer linear program for the problem is as follows:

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

Algorithm 2: Mettu-Plaxton Approach

1. **for** *each facility* $j \in F$ **do**
   define:
   $r_j = r, r > 0$ at which $\sum_{i:c_{ij} \leq r}(r - c_{ij}) = f_j$
   $B_j = \{i \in V | c_{ij} \leq r_j\}$
2. Sort r_j in the non-decreasing order. WLOG, assume $r_1 \leq r_2 \leq \cdots \leq r_m$.
3. Let $F' := \{r_1, r_2, \cdots r_m\}$ s.t. the index-set of F' represents the facilities. Call the set $I[F']$.
4. Let $j := 1, and\ X := \emptyset$.
5. **for** *facility* $j \in I[F']$ *in order* **do**
   Set $X = X \cup \{j\}$.
   let $M_j = \{r_l \in F' | c_{jl} \leq 2 \times r_l\}$.
   Set $F' = F' \setminus M_j$ with keeping the sorted order among the remaining elements as before.
   Note: $j \in M_j$ vacuously.
   Re-label the elements in the new F' starting from index 1. (new $I[F'] \subset$ old $I[F']$.)
   If $F' = \emptyset$, break.
6. Assign each client to the nearest facility in X. Let $\sigma : V \to X$ denote the mapping.
7. Output X and σ.

$$\sum_j x_{ij} \geq 1 \qquad \forall i \in V$$

$$x_{ij}, y_j \in \{0, 1\} \qquad \forall i \in V, j \in F$$

The LP-relaxation of this program gives:

$$\begin{aligned} min \qquad & \sum_{j \in F} f_j \times y_j + \sum_{i \in V} x_{ij} \times c_{ij} \\ s.t. \qquad & x_{ij} \leq y_j \qquad \forall i \in V, j \in F \\ & \sum_j x_{ij} \geq 1 \qquad \forall i \in V \\ & x_{ij}, y_j \geq 0 \qquad \forall i \in V, j \in F \end{aligned}$$

The dual program is:

$$\begin{aligned} max \qquad & \sum_{i \in V} v_i \\ s.t. \qquad & \sum_{i \in V} w_{ij} \leq f_j \qquad \forall j \in F \\ & v_i - w_i j \leq c_{ij} \qquad \forall i \in V,\ j \in F \\ & w_{ij}, v_i \geq 0 \qquad \forall i \in V,\ j \in F \end{aligned}$$

Theorem 4. *The running time for Algorithm 2 is polynomial in the size of input.*

Proof. Left to the reader. [Very similar to the proof of Thm.1] □

Now, we obtain a **dual solution** from Algorithm 2 as follows:

1. For each client $i \in V$, facility $j \in X$, set

$$w_{ij} = \begin{cases} r_j - c_{ij} & \text{if } i \in B_j \\ 0 & \text{otherwise} \end{cases}$$

2. For each client $i \in V$, set $v_i = min_j(w_{ij} + c_{ij})$.

Checking the **feasibility** of the dual solution:

1. For each facility $j \in X$, $\sum_{i \in B_j}(r_j - c_{ij}) = f_j$.
2. For each client $i \in V$, facility $k \in X$, $v_i = min_j(w_{ij} + c_{ij}) \leq (w_{ik} + c_{ik})$.

Consider the *Total Cost* of the solution:

$$Total\,Cost = (a) \sum_{j \in X} \sum_{i \in B_j} (w_{ij} + c_{ij}) + (b) \sum_{i \notin B_j | j \in X} c_{i\sigma(i)}.$$

Notice that the clients that contribute to (a) are disjoint from the clients that contribute to (b). Now, consider any client $i \in V$. Let $v_i = w_{ik} + c_{ik}$, where $k \in F$ is some facility that minimizes the client contribution. Let $cost(i)$ denote the cost for the client i in our solution.

Theorem 5. *$\forall i \in V$, $cost(i) \leq 3 \times v_i$.*

Proof. We will prove this by considering all cases. [From Algorithm 2] For all cases, we have: $\exists p \in X$ s.t. $r_p \leq r_k$ (p could be equal to k) and $c_{pk} \leq 2 \times r_k$.

1. Suppose, $\exists j \in X$ s.t. $i \in B_j$.
   (a) If $j = p$, then $cost(i) = (r_j - c_{ij}) + c_{ij} = r_j \leq r_k$. Now, we show that $v_i \geq r_k$. There are two possibilities:
      - If $i \in B_k$, then $v_i = (r_k - c_{ik}) + c_{ik} = r_k$. Hence, $v_i \geq r_k$.
      - If $i \notin B_k$, then $v_i = c_{ik} \geq r_k$. Hence, $v_i \geq r_k$.

      In either of the two sub-cases, we see that $v_i \geq r_k$.
   (b) If $j \neq p$, then $cost(i) = (r_j - c_{ij}) + c_{ij} = r_j \leq_1 c_{ip} \leq_2 c_{ik} + c_{kp} \leq_3 c_{ik} + 2 \times r_k$. The first inequality is due to the claim 2. The second inequality follows from triangle inequality induced by *metric* c, and the final inequality follows from the hypothesis.

   Claim 2. *For distinct $p, j \in X$ and $i \in V$, $(r_j - c_{ij}) + c_{ij} \leq c_{ip}$, given $i \in B_j$.*

Proof. First of all, for any distinct $p, j \in X$, $B_p \cap B_j = \emptyset$. Hence, $i \notin B_p$. Assume $\sigma(i) = p$. Then $cost(i) = c_{ip} \leq (r_j - c_{ij}) + c_{ij} = r_j$. Let $r_p \leq r_j$. Otherwise, $c_{ip} > r_p > r_j$ Then distinct $p, j \in X$ also implies $c_{pj} > 2 \times r_j$ [from our construction]. This further implies, $c_{ip} + c_{ij} > 2 \times r_j$. Hence, $c_{ip} > r_j$ (contradiction). So, $\sigma(i) = j$. □

We are left to show $3 \times v_i \geq c_{ik} + 2 \times r_k$ for case (b) to satisfy the theorem. There are two possibilities:

- If $i \in B_k$, then $v_i = r_k \geq c_{ik}$. Therefore, $3 \times v_i = 3 \times r_k \geq c_{ik} + 2 \times r_k$.
- If $i \notin B_k$, then $v_i = c_{ik} \geq r_k$. Therefore, $3 \times v_i = 3 \times c_{ik} \geq c_{ik} + 2 \times r_k$.

The claim holds true for both cases.

Hence, for both sub-cases (a) and (b), the inequality, $cost(i) \leq 3 \times v_i$ holds.

2. Suppose $i \notin B_j$ for any $j \in X$. Then $cost(i) = c_{i\sigma(i)} \leq c_{ip}$ since $p \in X$. It follows that $cost(i) \leq c_{ip} \leq 3 \times v_i$ from the similar argument as that of case 1(b).

□

Now, the total cost of our solution is given by $Total\,Cost = \sum_i cost(i) \leq \sum_i (3 \times v_i) = 3 \times \sum_i v_i$. Let OPT denote the optimal primal integral solution.

Theorem 6. *$Total\,Cost \leq 3 \times OPT$.*

Proof. Let $cost(v, w)$ represent the dual feasible solution that we achieve from the algorithm 2. By weak-duality, we know that a feasible dual solution gives a lower bound to the optimal primal solution, which is, in turn, a lower bound for optimal primal integral solution. Hence,

$$Total\,Cost \leq 3 \times cost(v, w) \leq 3 \times LP_{Dual} \leq 3 \times LP_{Primal} \leq 3 \times OPT.$$

□