# The Uncapacitated Facility Location Problem (UFLP) : Greedy Algorithm

1805092- Ishika Tarin

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology

December 2023

# Outline

1. Normal Greedy Approach for uncapacitated facility location problem
2. Two proposed modification for further improvements
3. Dual fitting algorithm for the uncapacitated facility location problem
4. Important lemmas
5. An Important Theorem

# Next Topic

1. Normal Greedy Approach for uncapacitated facility location problem
2. Two proposed modification for further improvements
3. Dual fitting algorithm for the uncapacitated facility location problem
4. Important lemmas
5. An Important Theorem

# Normal Greedy Approach for uncapacitated facility location problem

- **Initialization:**
  - Start with all facilities closed and no customers served.
  - Initialize an empty set of open facilities.

# Normal Greedy Approach for uncapacitated facility location problem

- **Initialization:**
  - Start with all facilities closed and no customers served.
  - Initialize an empty set of open facilities.
- **Iteration:**
  - For each customer, assign it to the nearest open facility.

# Normal Greedy Approach for uncapacitated facility location problem

- **Initialization:**
  - Start with all facilities closed and no customers served.
  - Initialize an empty set of open facilities.
- **Iteration:**
  - For each customer, assign it to the nearest open facility.
- **Open Facility:**
  - Open the facility that minimizes the total cost of opening and serving the assigned customers.

# Normal Greedy Approach for uncapacitated facility location problem

- **Initialization:**
  - Start with all facilities closed and no customers served.
  - Initialize an empty set of open facilities.
- **Iteration:**
  - For each customer, assign it to the nearest open facility.
- **Open Facility:**
  - Open the facility that minimizes the total cost of opening and serving the assigned customers.
- **Update:**
  - Update the set of open facilities.

# Normal Greedy Approach for uncapacitated facility location problem

- **Initialization:**
  - Start with all facilities closed and no customers served.
  - Initialize an empty set of open facilities.
- **Iteration:**
  - For each customer, assign it to the nearest open facility.
- **Open Facility:**
  - Open the facility that minimizes the total cost of opening and serving the assigned customers.
- **Update:**
  - Update the set of open facilities.
- **Repeat:**
  - Repeat steps 2-4 until all customers are served.

# Normal Greedy Approach for uncapacitated facility location problem

- **Initialization:**
  - Start with all facilities closed and no customers served.
  - Initialize an empty set of open facilities.
- **Iteration:**
  - For each customer, assign it to the nearest open facility.
- **Open Facility:**
  - Open the facility that minimizes the total cost of opening and serving the assigned customers.
- **Update:**
  - Update the set of open facilities.
- **Repeat:**
  - Repeat steps 2-4 until all customers are served.
- **Output:**
  - The open facilities and the assignment of customers to facilities constitute the solution.

To be more precise,
let,
$X =$ the set of facilities opened so far,
$S =$ the set of clients that are not connected to facilities in $X$ so far.

To be more precise,
let,
$X =$ the set of facilities opened so far,
$S =$ the set of clients that are not connected to facilities in $X$ so far.
Some $i \in F - X$ and $Y \subseteq S$ will be picked in every iteration that minimizes the ratio

$$\frac{f_i + \sum_{j \in Y} c_{ij}}{|Y|}.$$

To be more precise,
let,
$X =$ the set of facilities opened so far,
$S =$ the set of clients that are not connected to facilities in $X$ so far.
Some $i \in F - X$ and $Y \subseteq S$ will be picked in every iteration that minimizes the ratio

$$\frac{f_i + \sum_{j \in Y} c_{ij}}{|Y|}.$$

After every iteration, $i$ will be added to $X$, $Y$ will be removed from $S$, and it will repeat.

To be more precise,
let,
$X$ = the set of facilities opened so far,
$S$ = the set of clients that are not connected to facilities in $X$ so far.
Some $i \in F - X$ and $Y \subseteq S$ will be picked in every iteration that minimizes the ratio

$$\frac{f_i + \sum_{j \in Y} c_{ij}}{|Y|}.$$

After every iteration, $i$ will be added to $X$, $Y$ will be removed from $S$, and it will repeat.
To find the appropriate set $Y \subseteq S$, for any given facility $i$, the clients are sorted in $S$ by their distance from $i$, from nearest to farthest, and the set $Y$ minimizing the ratio for $i$ will be some prefix of this ordering.

# Next Topic

1. Normal Greedy Approach for uncapacitated facility location problem
2. Two proposed modification for further improvements
3. Dual fitting algorithm for the uncapacitated facility location problem
4. Important lemmas
5. An Important Theorem

# Two proposed modification for further improvements

- Setting facility cost $f_i$ to 0 instead of removing it

# Two proposed modification for further improvements

- Setting facility cost $f_i$ to 0 instead of removing it
- Switching assignments to other facilities that may be opened later instead of fixing the assignment of clients to a facility once made.

# Two proposed modification for further improvements

$S \leftarrow D$
$X \leftarrow \varnothing$
$S \neq \varnothing$

Choose $i \in F$ and $Y \subseteq D - S$ minimizing

$$\left( \frac{f_i - \sum_{j \in \overline{S}} (c(j, X) - c_{ij})_+ + \sum_{j \in Y} c_{ij}}{|Y|} \right)$$

$f_i \leftarrow 0$
$S \leftarrow S - Y$
Open all facilities in $X$ and assign each client $j$ to the closest facility in $X$.

# Next Topic

# Dual fitting algorithm for the uncapacitated facility location problem

$\alpha \leftarrow 0$
$S \leftarrow D$
$X \leftarrow \emptyset$
$\hat{f}_i \leftarrow 2f_i$ for all $i \in F$
while $S \neq \emptyset$ // While not all clients neighbor a facility in X
    Increase $\alpha_j$ for all $j \in S$ uniformly until $[\exists j \in S, i \in X$ such that $\alpha_j = c_{ij}]$ or $[\exists i \in F - X : \sum_{j \in S} (\alpha_j - c_{ij})^+ + \sum_{j \in \overline{S}} (c(j, X) - c_{ij})^+ = \hat{f}_i]$
    if $\exists j \in S, i \in X$ such that $\alpha_j = c_{ij}$ then
        $S \leftarrow S - \{j\}$ // $j$ becomes a neighbor of an existing facility $i$ in $X$
    else
        $X \leftarrow X \cup \{i\}$ // Facility $i$ is added to $X$
        for all $j \in S$ such that $\alpha_j \geq c_{ij}$
        do $S \leftarrow S - \{j\}$
Open all facilities in $X$ and assign each client $j$ to the closest facility in $X$.

# Next Topic

## lemma 1

Let $\alpha$ be the final set of bids, and let $X$ be the set of facilities opened by the algorithm. The first lemma says that the total bids of all clients equal the cost of the solution with facility costs $\hat{f}$.

$$\sum_{j \in D} \alpha_j = \sum_{j \in D} c(j, X) + 2 \sum_{i \in X} f_i.$$

# proof of lemma 1

**Initialization:**
The equality $\sum_{j \in D} \alpha_j = \sum_{j \in D} c(j, X) + 2 \sum_{i \in X} f_i$ is initially true since $S = D$ and $X = \emptyset$.
**Loop Execution:**
In each iteration of the loop, one of two cases occurs:
**Case 1: Connecting to Existing Facility:** If some $j \in S$ is connected to a facility $i$ already in $X$, then $\alpha_j = c(j, X)$. The equality holds, and $j$ is removed from $S$.
**Case 2: Opening a New Facility:** If a new facility $i$ is opened in $X$, then the algorithm removes from $S$ all $j \in S$ such that $\alpha_j - c_{ij} \geq 0$. Let $S'$ represent this subset of $S$.

# proof of lemma 1

**Change in Cost:**
The change in cost for the left-hand side is $\sum_{j \in S'} \alpha_j$. For the right-hand side, the change is calculated while factoring facility cost with scale of 2 for making it feasible as $2f_i + \sum_{j \in S: \alpha_j \geq c_{ij}} c_{ij} - \sum_{j \in \overline{S}} (c(j, X) - c_{ij})^+$.
**Analysis:**
The algorithm maintains the equality by ensuring that the changes in cost for both sides are the same. The key observation is that $2f_i = \hat{f}_i = \sum_{j \in S: \alpha_j \geq c_{ij}} c_{ij} + \sum_{j \in \overline{S}} (c(j, X) - c_{ij})^+$, where $\hat{f}_i$ is the doubled facility cost.
**Conclusion:**
The algorithm's correctness is supported by the consistent maintenance of the equality throughout its execution, ensuring that the dual solution remains valid at each step.

**lemma 2**

Consider the time $\alpha_j$ at which $j$ first connects to some facility. Then the bid of client $k$ on facility $i$ at that time, for any client $k$ such that $\alpha_k \leq \alpha_j$, is at least $\alpha_j - c_{ij} - 2c_{ik}$.

# Proof of lemma 2

If $k$ connects to a facility at the same time as $j$, then $\alpha_j = \alpha_k$, and at time $\alpha_j$, its bid on facility $i$ is $(\alpha_k - c_{ik})^+ = (\alpha_j - c_{ik})^+ \geq \alpha_j - c_{ij} - 2c_{ik}$.

# Proof of lemma 2

If $k$ connects to a facility at the same time as $j$, then $\alpha_j = \alpha_k$, and at time $\alpha_j$, its bid on facility $i$ is $(\alpha_k - c_{ik})^+ = (\alpha_j - c_{ik})^+ \geq \alpha_j - c_{ij} - 2c_{ik}$.
Now suppose $k$ connects to a facility at an earlier time than $j$. Let $h$ be the facility that client $k$ is connected to at time $\alpha_j$.
Then at time $\alpha_j$, the bid that $k$ offers facility $i$ is $(c_{hk} - c_{ik})^+$.

# Proof of lemma 2

If $k$ connects to a facility at the same time as $j$, then $\alpha_j = \alpha_k$, and at time $\alpha_j$, its bid on facility $i$ is $(\alpha_k - c_{ik})^+ = (\alpha_j - c_{ik})^+ \geq \alpha_j - c_{ij} - 2c_{ik}$.
Now suppose $k$ connects to a facility at an earlier time than $j$. Let $h$ be the facility that client $k$ is connected to at time $\alpha_j$.
Then at time $\alpha_j$, the bid that $k$ offers facility $i$ is $(c_{hk} - c_{ik})^+$.
By the triangle inequality, we know that $c_{hj} \leq c_{ij} + c_{ik} + c_{hk}$. Furthermore, since $j$ first connects to a facility at a time later than $\alpha_k$, it must be the case that $j$ did not earlier connect to $h$, and so $\alpha_j \leq c_{hj}$. Thus, we have $\alpha_j \leq c_{ij} + c_{ik} + c_{hk}$. So the bid of client $k$ on facility $i$ at time $\alpha_j$ is $(c_{hk} - c_{ik})^+ \geq c_{hk} - c_{ik} \geq \alpha_j - c_{ij} - 2c_{ik}$, as claimed.

## lemma 3

Let $A \subseteq D$ be any subset of clients. Reindex the clients of $A$ so that $A = \{1, \ldots, p\}$ and $\alpha_1 \leq \ldots \leq \alpha_p$. Then for any $j \in A$,

$$\sum_{k=1}^{j-1} (\alpha_j - c_{ij} - 2c_{ik})^+ + \sum_{k=j}^{p} (\alpha_j - c_{ik}) \leq \hat{f}_i.$$

# Proof of lemma 3

We know that at any time, the sum of the bids on facility $i$ is at most the facility cost $\hat{f}_i$. By Lemma 2, at time $\alpha_j$, for all clients $k$ with $k < j$, the bid of $k$ for facility $i$ is at least $\alpha_j - c_{ij} - 2c_{ik}$. For all clients $k \geq j$, since $\alpha_k \geq \alpha_j$, at any time just before $\alpha_j$, they have not connected to a facility, so their bid on facility $i$ at time $\alpha_j$ is $(\alpha_j - c_{ik})^+ \geq \alpha_j - c_{ik}$. Putting these together gives the lemma statement.

## lemma 4

Let $v_j = \frac{\alpha_j}{2}$, and let $w_{ij} = (\max(v_j - c_{ij}, 0))$. Then $(v, w)$ is a feasible solution to the dual.

$$\sum_{j \in D} \alpha_j = \sum_{j \in D} c(j, X) + 2 \sum_{i \in X} f_i.$$

# Proof of lemma 4

The algorithm starts at time 0, and uniformly increases all $\alpha_j$ with $j \in S$. At time $t$, any client $j$ not yet connected to a facility (and thus $j \in S$) has $\alpha_j = t$.

# Next Topic

**Theorem 1**

The last proposed algorithm is a 2-approximation algorithm for the uncapacitated facility location problem.

# Proof of theorem 1

Combining all Lemmas, we have that

$$\sum_{j\in D} c(j,X) + \sum_{i\in X} f_i \leq \sum_{j\in D} c(j,X) + 2\sum_{i\in X} f_i = \sum_{j\in D} \alpha_j = 2\sum_{j\in D} v_j \leq 2\mathrm{OPT},$$

where the final inequality follows since $\sum_{j\in D} v_j$ is the dual objective function, and by weak duality is a lower bound on the cost of the optimal integer solution.
We actually prove that

$$\sum_{j\in D} c(j,X) + 2\sum_{i\in X} f_i \leq 2\sum_{j\in D} v_j$$

for the feasible dual solution $(v, w)$. Thus, the algorithm is Lagrangean multiplier preserving a $2(2+\varepsilon)$-approximation algorithm . .

# Thank You