J. Electrical Systems 20-7s (2024): 461-468

Manisha Bansal[1]

Seema Aggarwal[2]

Geeta Aggarwal[3*]

Neelima Gupta[4]

A 3-Approximation Algorithm for Uniform Capacitated Facility Location Problem with Penalties

Journal of Electrical Systems

Abstract

We introduce an algorithm based on local search for "*Uniform Capacitated Facility Location Problem with Penalties (UCFLPP)" and* analyze the same. The algorithm is same as given by Korupolu et al. for the "*Uniform Capacitated Facility Location Problem (UCFLP*)". Aggarwal et al. in their work proved that the algorithm given by Korupolu et al. is a 3-factor approximation algorithm for UCFLP. We extend idea of Aggarwal et al. to show that the same algorithm works for the problem with penalty incorporated. It has the same approximation guarantee for UCFLPP, as given by them. This improves upon the current best of 5.732 factor for the problem, which is an LP based algorithm by Lv and Wu.

Keywords: Facility Location, Penalties, Local Search, Approximation Algorithm

1. Introduction

Many attempts have been made to study facility location problem in the various fields of Computer Science and Operations Research. Its application lies in logistics and supply chain management. In classical facility location problem (FLP), our input includes n facilities $F = \{1, . . , n\}$; m clients $C = \{1, . . , m\}$ and a metric c_{ij} which is the cost associated with assigning client j's unit demand to facility *i.*

A client *j* belonging to set *C* has a demand d_j to be served. Our goal is to identify *a* subset *S* of facilities *F* that serve all clients while optimizing both the overall client service cost and the total facility cost of all the facilities that make up *S*. This problem has been explored to almost close the gap between the hardness result [1] and the best approximation factor achieved so far [2].

There is another variant of FLP in which we may choose to leave some of the clients unserved. For each client that remains unserved, a penalty cost p_j per unit of the demand unserved for client j is added to the solution cost. The problem is named as "uncapacitated facility location problem with penalties (FLPP)". Charikar et al. [3] gave a 3-factor approximation algorithm for FLPP. It was improved to 2-factor in [4]. Approximation factor achieved so far is 1.5148 for the case of linear penalties by Li et al. [5]. Further we assume d_j to be 1. Arbitrary demands can be easily handled, details of which can be found in [6].

A local search algorithm for the capacitated variant of FLPP is proposed in our work. In this variant, a facility $i \in F$ has a capacity c_i, a limited capacity, resulting in a constraint on the clients that can be served by a facility. This problem is named as "capacitated facility location problem with penalties (CFLPP)". In many natural settings, when similar types of facilities are required to be set up, they also have similar/same capacities. For example, consider the case of a soft drink company that wants to install bottle dispensers in different locations of a city. All the dispensers are of same type, size, and capacity. This variant of CFLPP is called Uniform capacitated FLPP (UCFLPP), and $c_i = c$ for every facility. This problem has not been explored much. Lv and Wu [7] have proposed an approximation algorithm with a 5.732-factor. Another approximation algorithm based on local search technique with a 5.83-factor has been proposed by Xu et al. in [8] for "universal facility location problem with penalties" which is a general form of CFLPP.

A solution for UCFLPP is proposed in this work which is same as given by Korupolu et al. in [9] for UCFLP. We show that the proposed local search algorithm is a 3-approximation for UCFLPP. We build on the analysis of Aggarwal et al. in [10] to prove the result. To maintain continuity and completeness of the paper we have rephrased some key arguments. The local search algorithm for UCFLPP is discussed in section 2. Section 3 describes the algorithm. It is shown that the algorithm is a 3-factor approximation algorithm.

2. Local Search Algorithm

In "uniform capacitated facility location problem with penalties (UCFLPP)", the input includes *F*, which is a "set of facilities". A uniform capacity "*c*" is defined for facility *i* belonging to *F*. The input also includes a set of clients denoted by *C*. Metric c_{ij}, a cost incurred to serve each client j by a facility *i*. A penalty cost p_j is associated with client *j* if it remains unserved in the solution. Similar to the approach put forth by Korupolu et al. [9], a local search algorithm has been presented for the problem. Their algorithm does not incorporate penalty cost. The algorithm is described ahead:

To start with, a solution having a subset of facilities S of set F are opened. A min-cost flow problem along with penalties is used to assign the clients belonging to *C* to the facilities belonging to *S*.

[1]Indraprastha College for Women, University of Delhi, Delhi, India
[2]Miranda House, University of Delhi, Delhi, India
[3*](corresponding Author)PGDAV College, University of Delhi, Delhi, India
[4]Department of Computer Science, University of Delhi, India

It is done by considering a dummy facility N having capacity $|C|$, and an edge from each client j to N having cost p_j. Since there are only incoming edges coming into N, N does not affect the distances between the facilities and clients, defined by the distance metric. Now solve a min-cost flow problem. We abuse the notation and call S a solution. The cost of this solution S, $(P(S))$ is given by sum total of its facility cost $(P_f(S))$, its service cost $(P_s(S))$ and its penalty cost $(P_p(S))$ and is denoted by $P(S) = P_f(S) + P_s(S) + P_p(S)$. If possible, to improve the solution, the following operations are performed:

- **add**(f) : $S \leftarrow S \cup \{f\}$; f not belongs to S
- **delete**(f) : $S \leftarrow S - \{f\}$; f belongs to S
- **swap**(f, t) : $S \leftarrow S \cup \{t\} - \{f\}$; t not belongs to S, f belongs to S

Repeatedly we perform the operations on the current solution till the cost of the solution S is decreasing. We stop when none of the operations are able to reduce the cost. The solution thus obtained is locally optimal.

3. Bounding cost of the solution

Consider an optimal solution O and a locally optimal solution S. In S, assume a facility $\rho(j)$ assigned to a client j which belongs to C. And in solution O assume j is assigned facility $\rho'(j)$. Let S_j be the servicing cost of client j in S. Let O_j be the servicing cost of client j in O. Consider two hypothetical facilities N and N^*. For the purpose of analysis, we will assume N to be a part of S and N^* to be a part of O. A bipartite graph G is constructed on $F \cup \{N, N^*\} \cup C$ in a similar manner as done by Chudak and Williamson in [11]. An edge is drawn from every client $j \in C$ to facility $\rho'(j)$ or N^* (if j is paying penalty in O). Also, an edge is drawn to every client $j \in C$ from facility $\rho(j)$ or N (if j is paying penalty in S). Due to this, there is an incoming edge for every client and an outgoing edge in G. Next, perform a path decomposition on G to break it up into maximal paths and cycles. Assume P is the maximal paths set. Assume Y is the maximal cycles set. For a path p belonging to P beginning at facility $s \in S$ and ending at facility $t \in O$, $p = s, j_1, s_2, j_2, ...s_{k-1}, j_k, t$. Recall that $N \in S$ and $N^* \in O$. Note that $s_2, ..., s_{k-1} \in S \cap O$. The following are defined with respect to p

- **front(p)** = j, such that $\rho(j)$ equal to s   Note that $j = j_1$
- **end(p)** = j', such that $\rho'(j')$ equal to t Note that $j' = j_k$
- **move(p)** = $\sum_{J\in C\cap p}(O_j - S_j)$
- **length(p)** = $\sum_{J\in C\cap p}(O_j + S_j)$

move(p) represents the cost of reassignment of clients on a path p if one client is to be shifted from facility s to facility t. So, j_i is reassigned to s_{i+1} for $i = 1, 2, ...k-1$ and j_k is reassigned to t. Set of paths P is further partitioned into three sets

- **swap paths (P_s)** : $s - t$ is swap path if $s \in S$ and $t \in O - S$
- **transfer paths (P_t)** : $s - t$ is transfer path if $s, t \in S \cap O$
- **penalty paths (P_p)** : $s - t$ is penalty path if either $s = N$ or $t = N^*$ or both. Note that a penalty path is a swap path as well.

move(p) is defined a little differently if p is a penalty path. Let j be front(p) and j' be end(p). If

- p begins at N and ends at $t \neq N^* \in O - S$: On path p after N let there be a facility s. Let p' be the path that begins at s and ends at t. Then move(p) = $-p_j + O_j$ + move(p').
- p begins at $s \neq N \in S$ and ends at N^*: Assuming t be the facility that is just before N^* on the path. Let p' be the path that begins at s and ends at t . Then move(p) = move(p') $- S_{j'} + p_{j'}$.
- p begins at N and ends at N^*: Assuming s be the facility that is after N on p path. Assuming t be the facility that is just before N^* on the path. Let p' be the path that begins at s and ends at t. Then move(p) = $-p_j + Oj$ + move(p') $- S_{j'} + p_{j'}$.

To bound solution S cost, we separately prove the bound on all the three costs, i.e. the "service cost", "penalty cost" and cost of facilities for S.

3.1. Bounding Cost of service and Cost of penalty

To prove a bound on the cost of service and cost of penalty, we use add operation. Let the set of paths ($N_O(o)$) terminate at a facility o belonging to O whereas set of paths ($N_S(s)$) begin at a facility s belonging to S and the set of paths (N_S^o) begin at facility s and end at facility o. Consider a facility $o \in O - S$. We can write the following inequality with respect to adding a facility $o \in O - S$:

$$f_o + \sum_{p \in N_O(o)} move(p) \geq 0$$

Since S is locally optimal, therefore adding a facility o and reassigning some clients to o cannot decrease the cost of S. We write such inequalities for all $o \in O-S$.

Let o be a facility belonging to $O \cap S$. Then we can write the following inequality with respect to o:

$$\sum_{p \in N_O(o)} move(p) \geq 0$$

And, for such a facility, moving $|N_O(o)|$ more clients to o will not violate the capacity constraint at o. This is because, if that many paths are terminating at o then this implies that o is serving that many more clients in O. For $p \in N_O(N^*)$, we can write the following inequalities:

$$\sum_{p \in N_O(o)} move(p) \geq 0$$

Due to this move, few clients will be reassigned to other facilities on the path and tail client pays the penalty. The solutions cost will not be decreased because of this reassignment.
A move along a cycle can also be defined. Consider a cycle $y \in Y$, then

$$move(y) = \sum_{j \in C \cap y} (O_j - S_j) = 0$$

Above equality is due to the fact that the assignment of clients in O and in S is done optimally.

Assume $C_1 \subseteq C$ is the subset of clients that pay a penalty in both solutions solution S and O.
Let $C_2 \subseteq C$ be a subset of clients that pay a penalty in S only. Let $C_3 \subseteq C$ be the subset of clients that pay penalty in O only. Then, adding all the inequalities with respect to:
(1) add operation for $o \in O - S$
(2) move along paths in $N_O(o)$ $\forall$ o $\in O \cap S$ and in $N_O(N^*)$
(3) move along cycles in Y

we get :

$$\sum_{o \in O-S} f_o + \sum_{j \in C-(C_1+C_2+C_3)} (O_j - S_j) + \sum_{j \in C_2} (O_j - P_j) \sum_{j \in C_3} (P_j - S_j) \geq 0$$

Thus,

$$\sum_{j \in C-(C_1+C_2)} S_j + \sum_{j \in C_2} P_j \leq \sum_{o \in O-S} f_o + \sum_{j \in C-(C_1+C_3)} O_j + \sum_{j \in C_3} P_j$$

Or

$$\begin{aligned}\sum_{j \in C-(C_1+C_2)} S_j + \sum_{j \in C_2} P_j + \sum_{j \in C_1} P_j &\leq \sum_{o \in O-S} f_o + \sum_{j \in C-(C_1+C_3)} O_j + \sum_{j \in C_3} P_j + \sum_{j \in C_1} P_j \\ &\leq \sum_{o \in O-S} f_o + \sum_{o \in O \cap S} f_o + \sum_{j \in C-(C_1+C_3)} O_j + \sum_{j \in C_3} P_j + \sum_{j \in C_1} P_j\end{aligned}$$

Which results into the following inequality

$$\begin{aligned} P_s(S) + P_p(S) &\leq P_f(O) + P_s(O) + P_p(O) \\ &= P(O) \end{aligned} \qquad (1)$$

3.2. Bounding Facility Cost

Facility cost of the solution S is bounded separately. For that, we require only those paths that begin at facilities in $S - O$. Therefore, all the paths that begin at facilities in $S \cap O$ and those that begin at N are not considered, and are removed from the various sets of paths defined in the previous sections. The clients that belong to these paths are also removed from C. The cycles and the clients belonging to cycles are also removed from C, due to the same reason. For an $s \in S-O$, $N_S(s)$ consists of swap paths, transfer paths and penalty paths. Let $S(s), T(s), P(s) \subseteq N_S(s)$ denote the set of swap paths, transfer paths and penalty paths, respectively, in $N_S(s)$.
To identify the inequalities for bounding facility cost, we will consider delete/swap operation for each facility $f \in S - O$. The clients of f will be reassigned to various facilities belonging to S other than f and/or to a facility $o \in O - S$. S being a locally optimal solution, none of these operations help in improving the solution , which gives us desired inequalities.
To reassign the clients served by facility f, that lie at the front of a transfer path or a penalty path, we will perform a move along that path. So, consider one such path, say p. If $p \in T(f)$, then a move(p) is performed along that path, which results in reassignment of clients on the path p. Facilities along the path would lose and receive a client. And the facility, say t, at the end of the transfer path would receive an additional client. The facility t may receive at most as many clients through such moves as is the number of transfer paths terminating at that facility.
Recall that $t \in S \cap O$. If r number of transfer paths terminate at t, then t would be serving r more clients in O than in S. Therefore, it can receive r more clients through such shifts without violating the capacity constraint at t. If p is a penalty path, then the facility at the end of p is the hypothetical facility N^*. All the clients on the path are reassigned to facilities on the path, as explained earlier, except tail(p) which will now be paying penalty cost.

We will also use a fractional mapping π, as discussed in Aggarwal et al. [10] to reassign the clients of *f* to other facilities in *S*. For that define *init_ms(p)*, *ms(p)* and *rem_ms(p)*, for a swap path *p*, as follows:

$$init\_ms(p) = min(1, \frac{c-|S(s)|}{|S(s)|})$$

Consider a facility $o \in O - S$. *ms(p)* is defined for paths in *S(o)* as follows. If $init\_ms(N_s^o) \leq init\_ms(S(o))/2$ then $ms(p) = init\_ms(p)$ for $p \in N_s^o$. Otherwise, *init_ms(p)* is reduced to obtain *ms(p)*. Note that there can be at most one facility $s \in S - O$ for which $init\_ms(N_s^o) > init\_ms(S(o))/2$. Thus $ms(p) \leq init\_ms(p)$ and is such that for every $s \in S - O$ and $o \in O - S$, $ms(N_s^o) \leq ms(S(o))/2$. Mass of all the paths in a set equals mass of set as a whole. *rem_ms(p)* denotes the remainder mass of a swap path p and is equal to $1-ms(p)$.

Fractional mapping, π, is defined for $o \in O - S$, π_O: $S(o) \times S(o) \rightarrow R^+$ having following properties:

- $\pi_O(p, q) > 0$, if the paths *p* and *q* begin at distinct facilities in $S - O$
- $\sum_{q \in S(o)} \pi_o(q,p) = \sum_{q \in S(o)} \pi_o(p,q) = ms(p)$ *for all* $p \in S(o)$

Consider a delete/swap operation to close a facility s ∈ S−O. It is necessary to reassign the clients that s serves. Recall that the clients that lie at the head of a "transfer path" or a "penalty path" in $N_S(s)$, are reassigned using a move on that path. For a path p ∈ S(s), if π(p, q) > 0, then we perform a move along the path p and assign end(p) to s′, both upto an extent of π(p, q). Note that s′ is the facility at which p′ begins. end(p) client is assigned to additional facilities belonging to S up to an overall extent of total of ms(p). Let μ(s) denote the cost of reassignment of clients of s due to fractional mapping. Then

$$\begin{aligned}\sum_s \mu(s) &\leq \sum_s \sum_{p \in S(s)} \sum_{q \in P_S} \pi(p,q)(move(p) + length(q)) \\ &= \sum_{p \in P_S} ms(p)(move(p) + length(q)) \qquad (2)\end{aligned}$$

We also perform a move on the transfer paths and penalty paths in $N_S(s')$. This is done to avoid any violation of capacity constraint at facility s′ when additional clients are received by s′ due to fractional mapping. For all the transfer paths and penalty paths *q* belonging to union of *T(s′)*, *P(s′)*, we perform a move along q, up to $\pi(p, p')/ms(S(s'))$ extent. Due to this, total move along *q* is done up to a maximum of *1*. So, if s′ receives x more clients due to fractional mapping, it also loses y clients due to moves on transfer paths and penalty paths in $N_S(s')$. Therefore s′ gets $x-y$ additional clients. Now

$$y = |T(s') + P(s')| \sum_p \sum_{q \in S(s')} \frac{\pi(p,q)}{ms(S(s'))} = |T(s') + P(s')| \ \frac{x}{ms(S(s'))}$$

Also, $x \leq ms(S(s'))$. Therefore,

$$\begin{aligned}x - y = x(1 + \frac{|T(s')+P(s')|}{ms(S(s'))} &\leq ms(S(s')) - |T(s') + P(s')| \\ &\leq c - |S(s')| - T(s') - P(s') \qquad (3)\end{aligned}$$

Before getting $x - y$ additional clients, s' was serving $|S(s')| + |T(s')| + |P(s')|$ clients. Therefore, capacity constraint will not get violated on s′ due to these additional clients.

Capacity is also not violated at any of the facility $s'' \in S \cap O$ that receives additional clients because of transfer paths shifts. The reason being that if t transfer paths terminating at s″are there, then s″would be serving *t* less clients in *S* then in *O* and can take in that many more clients in the process of reassignment within *S*.

Let *Γ*(*s*) denote the total cost of shifts on the transfer paths and penalty paths beginning at s. Shifts on $p \in T(s) \cup P(s)$ happen once when s is considered in a close/swap operation and once when transfer paths originating at *s* are used to make room for additional clients received by s due to fractional mapping. Therefore

$$\Gamma(s) \leq 2 \sum_{p \in T(s) \cup P(s)} move(p) \qquad (4)$$

When a facility s is closed, a swap path p originating at s is used up to an extent of *ms(p)* for fractional assignment and a move is performed up to that extent on this path. This way, *ms(S(s))* amount of clients get reassigned to other facilities in *S*. To accommodate remaining rem_ms(S(s)) clients of s, we open a facility $o \in O - S$ when *s* is closed. Let *j* be *front(p)*, $p \in S(s)$ then j is assigned up to an extent of rem_ms(*p)* to o. This reassignment leads to total increment in service cost $c_{s,o} rem\_ms(S(s))$. Note that $c_{s,o} \leq length(p), p \in N_s^o$

When $o\in O-S$ is opened in a swap $< s, o >$ operation, o receives rem_ms(*S(s)*) clients and is still left with *c−rem_ms(S(s))* vacant capacity. To utilize this capacity, we will also perform a move along the swap paths in *S(o)*, up to an extent of $\beta_{s,o}$, where $\beta_{s,o}$ is defined as follows:

$$\beta_{s,o} = min(1, \frac{c-rem\_ms(S(s))}{rem\_ms(S(o))})$$

Let $\mu'(s,o)$ be the increment in cost of service because of this reassignment, then

$$\mu'(s,o) = \beta(s,o) \sum_{p\in S(o)} rem\_ms(p)\ \ move(p)$$

Now that we have considered all the reassignments due to swap $< s, o >$, the inequality can be written w.r.t. $< s, o >$ as

$$f_o - f_s + c_{s,o} rem\_ms(S(s)) + \mu(s) + \Gamma(s) + \mu'(s,o) \geq 0 \tag{5}$$

A linear combination of the inequalities resulting from each swap $< s, o >$, for $s \in S - O$, $o \in O - S$ with a mass $\phi_{s,o}$ results into the following inequality:

$$\sum_{s,o} \phi_{s,o} f_o - \sum_{s,o} \phi_{s,o} f_s + \sum_{s,o} \phi_{s,o} c_{s,o} rem\_ms(S(s)) + \sum_{s,o} \phi_{s,o} \mu(s) + \sum_{s,o} \phi_{s,o} \Gamma(s) + \sum_{s,o} \phi_{s,o} \mu'(s,0) \geq 0 \tag{6}$$

where,

$$\phi_{s,o} = \frac{rem\_ms(N_s^o)}{rem\_ms(S(s))}$$

and is 0 if rem_ms(S(s)) = 0

We will now separately simplify the terms of inequality (6). Consider second term of (6),

$$\sum_{s,o} \phi_{s,o} f_s = \sum_s \sum_o \frac{rem\_ms(N_s^o)}{rem\_ms(S(s))} f_s = \sum_s \frac{rem\_ms(S(s))}{rem\_ms(S(s))} f_s = \sum_s f_s \tag{7}$$

We will simplify the third term of (6) now

$$\sum_{s,o} \phi_{s,o} c_{s,o} rem\_ms(S(s)) = \sum_{s,o} c_{s,o} rem\_ms(N_s^o) \leq \sum_{p\in P_S} rem\_ms(p) length(p) \tag{8}$$

Since, $c_{s,o} \leq length(p), \forall p \in N_s^o$

Fourth term of (6) can be written as

$$\sum_{s,o} \phi_{s,o} \mu(s) = \sum_s \mu(s) \leq \sum_{p\in P_S} ms(p)(move(p) + length(p)) \tag{9}$$

where the last inequality is obtained due to (2). Fifth term of (6) can be written as:

$$\sum_{s,o} \phi_{s,o} \Gamma(s) = \sum_s \Gamma(s) \leq 2 \sum_s \sum_{p\in T(s)\cup P(s)} move(p) = 2 \sum_{p\in P_T \cup P_P} move(p) \tag{10}$$

where the second inequality is due to (4). First term and last term of (6) will be handled together. Now, we can write (6) as follows, using (7), (8), (9) and (10).

$$\begin{aligned}\sum_{s\in S-O} f_s &\leq \sum_{s,o} \phi_{s,o} f_o + \sum_{s,o} \phi_{s,o} \mu'(s,o) + \sum_{p\in P_s} rem\_ms(p) length(p) + \sum_{p\in P_S} ms(p)(move(p) + length(p)) + 2 \sum_{p\in P_T \cup P_P} move(p) \\ &\leq \sum_{s,o} \phi_{s,o} f_o + \sum_{s,o} \phi_{s,o} \mu'(s,o) + \sum_{p\in P_s} length(p) + \sum_{p\in P_S} ms(p)(move(p)) + 2 \sum_{p\in P_T \cup P_P} move(p)\end{aligned} \tag{11}$$

The first and second term of the equation on the RHS (11) are simplified next.

Lemma 3.1 $\forall f \in O, \sum_s \phi_{s,f} \leq 2$.

Proof: Note that $\forall s, f, \phi_{s,f} \leq 1$.

(1) Let $A \subseteq S - O$ be the facilities s such that $init\_ms(N_s^f) \leq init\_ms(S(f))/2$ *and* $|S(s)| \leq c/2$. *Let* $s \in A$ *and* $p \in N_s^f$ then ms(p) = init_ms(p) = 1 and so rem_ms(p) = 0. This implies that $rem\_ms(N_s^f) = 0$ and so for all $s \in A, \phi_{s,f} = 0$

(2) Let s be a facility not in A and $\mathrm{init\_ms}(\mathrm{N}_{\mathrm{s}}^{\mathrm{f}}) \leq \mathrm{init\_ms}(\mathrm{S(f)})/2$. For $p \in N_s^f$

$$rem\_ms(p) = 1 - ms(p) = 1 - init\_ms(p) = 2 - \frac{c}{|S(s)|}$$

However, for $p \in S(s)$ we have that

$$rem\_ms(p) = 1 - ms(p) \geq 1 - init\_ms(p) = 2 - \frac{c}{|S(s)|}$$

Therefore

$$\phi_{s,f} \leq \frac{|N_s^f|}{|S(s)|}$$

Thus, we only need to show that $\sum_{s \notin A} \phi_{s,f} \leq 2$. We now consider two cases

(1) If $\mathrm{init\_ms}(\mathrm{N}_{\mathrm{s}}^{\mathrm{f}}) \leq \mathrm{init\_ms}(\mathrm{S(f)})/2$, then

$$\sum_s \phi_{s,f} = \sum_{s \notin A} \phi_{s,f} \leq \sum_{s \notin A} \frac{|N_s^f|}{|S(s)|} \leq \sum_{s \notin A} \frac{|N_s^f|}{c/2} \leq \frac{|S(f)|}{c/2} \leq 2$$

(2) If $\mathrm{init\_ms}(\mathrm{N}_{\mathrm{s'}}^{\mathrm{f}}) > \mathrm{init\_ms}(\mathrm{S(f)})/2 \quad \forall \mathrm{s'} \in \mathrm{S} - \mathrm{O}$. This implies

$$\begin{aligned} init\_ms(N_{s'}^f) &\geq \sum_{s \neq s'} init\_ms(N_s^f) \\ &\geq \sum_{s \notin A \cup \{s'\}} init\_ms(N_s^f) \\ &= \sum_{s \notin A \cup \{s'\}} |N_s^f| \frac{c - |S(s)|}{|S(s)|} \\ &= \sum_{s \notin A \cup \{s'\}} \left(c \frac{|N_s^f|}{|S(s)|} - |N_s^f| \right) \end{aligned}$$

Since $\mathrm{init\_wt}(\mathrm{N}_{\mathrm{s'}}^{\mathrm{f}}) \leq |\mathrm{N}_{\mathrm{s'}}^{\mathrm{f}}|$ rearranging we get,

$$\sum_{s \notin A \cup \{s'\}} \frac{|N_s^f|}{|S(s)|} \leq \sum_{s \notin A} \frac{|N_s^f|}{c} \leq 1.$$

Now

$$\sum_{s \notin A \cup \{s'\}} \phi_{s,f} \leq \sum_{s \notin A \cup \{s'\}} \frac{|N_s^f|}{|S(s)|} \leq 1$$

and since $\phi_{s',f} \leq 1$

$$\sum_s \phi_{s,f} = \sum_{s \notin A} \phi_{s,f} \leq 2.$$

The proof is now complete.
So next,

$$\begin{aligned} \sum_s \phi_{s,f} \mu'(s,f) &= \sum_s (\phi_{s,f} \beta_{s,f} \sum_{p \in S(f)} rem\_ms(p) move(p)) \\ &= (\sum_s \phi_{s,f} \beta_{s,f}) \sum_{p \in S(f)} rem\_ms(p) move(p) \end{aligned}$$

If $\sum_s \phi_{s,f} \beta_{s,f} > 1$ then the value of $\beta_{s,f}$ is reduced in such a manner so that the sum is exactly 1. However, if $\sum_s \phi_{s,f} \beta_{s,f} = 1 - \Upsilon_f$, $\Upsilon_f > 0$, then the inequalities associated with the facility addition operation f ∈ O are considered, as follows

$$f_f + \sum_{p \in S(f)} rem\_ms(p) move(p) \geq 0 \tag{12}$$

and these are added to the inequality (11) with a mass Υ_f .

Therefore, taking Υ_f = maximum$\{0, 1-\sum_s \phi_{s,f}\beta_{s,f}\}$, we can substitute the second term of the inequality on the RHS (11) with

$$\sum_{s,f}\phi_{s,f}\mu'(s,f) + \sum_f \Upsilon_f(f_f + \sum_{p\in S(f)} rem\_ms(p)move(p))$$
$$= \sum_f \sum_{p\in S(f)}(1-\Upsilon_f)\ rem\_ms(p)move(p) + \sum_f \Upsilon_f f_f + \sum_f \sum_{p\in S(f)}\Upsilon_f\ rem\_ms(p)move(p)$$
$$= \sum_f \sum_{p\in S(f)} rem\_ms(p)move(p) + \sum_f \Upsilon_f f_f$$
$$= \sum_{p\in P(s)} rem\_ms(p)move(p) + \sum_f \Upsilon_f f_f$$

Therefore, inequality (11) now becomes

$$\sum_{s\in S-O} f_s \leq \sum_f(\Upsilon_f + \sum_s \phi_{s,f})\ f_f + \sum_{p\in P(s)} rem\_ms(p)ms(p) + \sum_{p\in P(s)} length(p) + \sum_{p\in P(s)} ms(p)move(p) + 2\sum_{p\in P_T\cup P_P} move(p) \quad (13)$$

$$\leq \sum_f(\Upsilon_f + \sum_s \phi_{s,f})\ f_f + \sum_{p\in P(s)}(move(p) + length(p)) + 2\sum_{p\in P_T\cup P_P} move(p)$$

$$\leq \sum_f(\Upsilon_f + \sum_s \phi_{s,f})\ f_f + \sum_{j\in P_s\cap C}(O_j - S_j + O_j + S_j) + 2\sum_{p\in(P_T\cup P_P)\cap(C-C_3)}(O_j - S_j) + 2\sum_{p\in P_T\cap C_3}(p_j - S_j)$$

$$\leq \sum_f(\Upsilon_f + \sum_s \phi_{s,f})\ f_f + 2\sum_{j\in C-C_3} O_j + 2\sum_{j\in C_3} p_j$$

Recall that $C_3 \subseteq C$ is client set incurring penalty only in O. Only thing that remains to be proved now is that

Lemma 3.2. $\sum_s \phi_{s,f}(1-\beta_{s,f}) \leq 1$.

Proof. When $\sum_s \phi_{s,f}\beta_{s,f} > 1$, value of some $\beta_{s,f}$ was reduced to make the sum become exactly equal to 1. So,

$$\sum_s \phi_{s,f}(1-\beta_{s,f}) = \sum_s \phi_{s,f} - 1 \leq 1,$$

according to Lemma 3.1, $\phi_{s,f} \leq 2$.

We now assume that no $\beta_{s,f}$ was reduced. Since rem_ms(S(f)) $\leq$ |S(f)| $\leq$ c
we have

$$\beta_{s,f} = min(1, \frac{c - rem\_ms(S(s))}{rem\_ms(S(f))}) \geq min(1, 1 - \frac{rem\_ms(S(s))}{rem\_ms(S(f))}) = 1 - \frac{rem\_ms(S(s))}{rem\_ms(S(f))}$$

Hence

$$\sum_s \phi_{s,f}(1-\beta_{s,f}) \leq \sum_s \frac{rem\_ms(N_s^f)}{rem\_ms(N_O(f))} = 1.$$

Using Lemma 3.2, the inequality 13 can now be written as

$$\sum_{s\in S-O} f_s \leq 2\sum_{f\in O-S} f_f + 2\sum_{j\in C-C_3} O_j + 2\sum_{j\in C_3} p_j \quad (14)$$

Or

$$\sum_{s\in S} f_s \leq 2\sum_{f\in O-S} f_f + \sum_{f\in S\cap O} f_f + 2\sum_{j\in C-C_3} O_j + 2\sum_{j\in C_3} p_j \quad (15)$$

$$\leq 2\sum_{f\in O} f_f + 2\sum_{j\in C-C_3} O_j + 2\sum_{j\in C_3} p_j$$

$\leq 2\ (P_f(O) + P_s(O) + P_p(O))$

Together with inequality 1, we obtain the following result

$$P(S) \leq 3P(O) .$$

Therefore, the algorithm based on local search having add operation, delete operation, and swap operation is a “3-factor approximation algorithm”. This approach addresses “uniform capacitated facility location problem with penalties”.

References

[1] S. Guha and S. Khuller, “Greedy strikes back: Improved facility location algorithms,” Journal of Algorithms, vol. 31, no. 1, pp. 28-248, 1999.

[2] S. Li, “A 1.488 approximation algorithm for the uncapacitated facility location problem,” in 38th International Conference on Automata, Languages and Programming, 2011.

[3] M. Charikar, S. Khuller, D. M. Mount and G. Narasimhan, “Algorithms for facility location problems with outliers,” in 12th Symposium on Discrete Algorithms, Washington DC, 2001.

[4] K. Jain, M. Mahdian, E. Markakis, A. Saberi and V. Vazirani, “ Greedy facility location algorithms analyzed using dual fitting with factor- revealing LP,” J. ACM, vol. 50, no. 6, pp. 795-824, 2003 .

[5] Y. Li, D. Du, N. Xiu and D. Xu, “Improved approximation algo- rithms for the facility location problems with linear/submodular penalties,” Algorithmica, vol. 73, no. 2, pp. 460-482, 2015.

[6] M. Pal, E. Tardos and T. Wexler, “ Facility location with nonuniform hard capacities.,” in 42nd IEEE Symposium on foundations of Computer Science, Washington DC, 2001.

[7] W. Lv and C. Wu, “An lp-rounding based algorithm for a capacitated uniform facility location problem with penalties,” J. Comb. Optim., vol. 41, no. 4, pp. 888-904, 2021.

[8] Y. Xu, D. Xu, D. Du and C. Wu, “ Improved approximation algorithm for universal facility location problem with linear penalties,” Theor. Comput. Sci, vol. 774, pp. 143-151, 2019.

[9] M. Korupolu, C. Plaxton and R. Rajaraman, “Analysis of a local search heuristic for facility location problems,” J. Algorithms, vol. 37, no. 1, pp. 146-188, 2000.

[10]A. Aggarwal, A. Louis, M. Bansal and e. al, “A 3-approximation algorithm for the facility location problem with uniform capacities.,” Math. Program., vol. 141, p. 527–547, 2013.

[11] F. Chudak and D. Williamson, “Improved approximation algorithms for capacitated facility location problems,” Math. Program., vol. 102, no. 2, p. 207–222, 2005.