Journal of Combinatorial Optimization (2023) 46:5
https://doi.org/10.1007/s10878-023-01067-w

A combinatorial approximation algorithm for *k*-level facility location problem with submodular penalties

Li Zhang[1] · Jing Yuan[1] · Zhizhen Xu[1] · Qiaoliang Li[1]

Accepted: 12 July 2023 / Published online: 4 August 2023

Abstract

We present an improved approximation algorithm for k-level facility location problem with submodular penalties, the new approximation ratio is 2.9444 for any constant k, which improves the current best approximation ratio 3.314. The central ideas in our results are as follows: first, we restructure the problem as an uncapacitated facility location problem, then we use the primal-dual scheme with greedy augmentation. The key technique of our result is that we change the way of last opening facility set in primal-dual approximation algorithm to get much more tight result for k-level facility location problem with submodular penalties.

Keywords Submodular penalties · Greedy augmentation · Primal-dual · k-level facility location problem

Mathematics Subject Classification 90C27 · 90C10

1 Introduction

The uncapacitated facility location problem (UFLP) has been extensively studied in the field of the facility location problem. Shmoys et al. (1997) provided the first constant factor approximation ratio 3.16 for UFLP. Chudak and Shmoys (2003) presented an improved approximation algorithm for UFLP with performance guarantee (1+1/e). Sviridenko (2002) used pipage rounding to get an approximation algorithm with approximation ratio 1.582. Jain and Vazirani (2001) proposed a primal-dual approximation algorithm for UFLP with approximation ratio 3. Mahdian et al. (2006) studied UFLP by dual-fitting and greedy augmentation which give approximation ratio 1.52.

The research is supported by National Nature Science Foundation of China (No.12071126).

✉ Qiaoliang Li
liqiaoliang@hunnu.edu.cn

[1] School of Mathematics and Statistics, Hunan Normal University, Changsha 410000, China

The currently best approximation ratio for the UFLP is 1.488 provided by Li (2013). By relating UFLP to set cover, Charikar and Guha (2005) proved the lower bound of UFLP is 1.463 by assuming NP $\subseteq$ DTIME($n^{\log\log n}$). The greedy augmentation introduced by Charikar and Guha (2005). Charikar et al. (2001) considered the uncapacitated facility location problem with penalties FLPWP and obtained a 3-performance guarantee based on primal-dual. Jain et al. (2003) presented a combinatorial 2-approximation ratio by dual fitting. Xu and Xu (2009) presented a 1.8526-approximation algorithm by primal-dual and local search heuristic. The currently best known approximation ratio for FLPWP is 1.488 which was given by Qiu and Kern (2016) based on dual-fitting and LP-rounding. Hayrapetyan et al. (2005) investigated the single level facility location problem with submodular penalties (FLPSP) and provided an 2.488-approximation algorithm. The penalties cost is a monotone increasing submodular function $h(\cdot)$ defined on client set $\mathcal{D}$, i.e., for any $A, B \subseteq \mathcal{D}$ with $A \subseteq B \subseteq \mathcal{D}$, $h(A) \leq h(B)$ and $h(A+j)-h(A) \geq h(B+j)-h(B)$. Li et al. (2013) gave an 2.375-approximation algorithm for the FLPSP by using the primal-dual and the greedy augmentation scheme. Then, Li et al. (2015b) provided an LP-rounding 2-approximation algorithm for the single level FLPSP. The k-facility location problem (k-UFLP) is a generalization of UFLP. The aim of k-UFLP is to open a subset of facilities and the size of the facility subset is at most k facilities, where k is a given constant positive integer, connect all clients to the closest opened facilities such that the total cost is minimized. Jain and Vazirani (2001) firstly considered the metric k-UFLP and gave a 6 primal-dual approximation algorithm. Jain et al. (2003) proved a 4-approximation algorithm by combining greedy scheme and dual fitting with factor-revealing LP. Zhang (2007) used local search approach and gave a $2+\sqrt{3}+\epsilon$-approximation ratio, which is the best approximation algorithm for k-UFLP. There are also many variant such as the squared metric k-facility location problem, Zhang et al. (2023) gave a $36.342+\epsilon$-approximation algorithm based on local search scheme, which is the currently best known approximation ratio. The k-facility location problem with linear penalties is also an extension of k-UFLP, such as Wang et al. (2018) proved a $2 + 1/p + \sqrt{3 + 2/p + 1/p^2} + \epsilon$ based on local search scheme, where p is positive integer, $\epsilon > 0$. k-level uncapacitated facility location problem (k-LFLP) is a generalization of UFLP, when $k = 1$, the k-LFLP is UFLP. The aim of k-LFLP is to connect all clients to opened facilities from level 1 to level k such that the sum of the opening and connection cost is minimized. By observing the structure of solution of k-LFLP, Byrka and Rybicki (2012) proposed a 3-approximation algorithm by rounding a fractional solution to an extended LP formulation, which is also the currently best approximation algorithm for k-LFLP. For k-LFLP, Krishnaswamy and Sviridenko (2012) proved that there is no polynomial time approximation algorithm with performance guarantee better than 1.61 unless NP $\subseteq$ DTIME($n^{O(\log\log n)}$). The k-level facility location problem with submodular penalties (k-FLPSP) is a variant of k-LFLP. Li et al. (2012) presented a primal-dual algorithms with a performance guarantee of 6 for k-FLPSP. Later on, Li et al. (2015a) proposed an LP-rounding $1 + \frac{2}{1-e^{-2}}$ (≈ 3.314)-approximation algorithm, which is the best known approximation ratio. In this paper, we present an improved combinatorial algorithm for the k-FLPSP. The main idea to get the result are local search and greedy augmentation. The main steps is as follows: firstly, we restructure the k-FLPSP, and then we present our primal-dual greedy augmentation approximation algorithm for the k-FLPSP. Finally, scaling the cost of the facility and penalty function, we prove that the presented algorithm has an approximation ratio 2.9444.

2 Problem statement and notation

In this paper, we present an improved combinatorial algorithm for the k-FLPSP. In the k-FLPSP, let $\mathcal{D}$ be the client set, $\mathcal{F}$ be the facility set, $\mathcal{F} = \bigcup_t \mathcal{F}^t$, $t \in \{1, 2, \cdots, k\}$. And $\mathcal{F}^t$ is the facilities set on the tth level, where $t \in \{1, 2, \cdots, k\}$. P is the paths set of k-level facilities, $P = \{p : p = (i_1 \in \mathcal{F}^1, i_2 \in \mathcal{F}^2, \cdots, i_k \in \mathcal{F}^k)\}$. Each client is assigned to a sequence of k different facilities, each of the k facilities belongs to a distinct level from level 1 to level k. Each client should pay a connection cost c_{jp} for being connected, where $c_{jp} = c_{ji_1} + \sum_{t=2}^{k} c_{i_{t-1}i_t}$ is the connection cost between client j and path p, c_{ji_1} is the connection cost between client j and facility $i_1 \in \mathcal{F}^1$, $c_{i_{t-1}i_t}$ is the connection cost between $i_{t-1} \in \mathcal{F}^{t-1}$ and $i_t \in \mathcal{F}^t$, where $t \in \{2, \cdots, k\}$. We consider the uncapacitated k-FLPSP, there is no capacitated restrictions for each facility. For the sake of simplification, f_{i_t} is the opening cost of facility $i_t \in \mathcal{F}^t$, $t \in \{1, 2, \cdots, k\}$. Given a nondecreasing submodular function $h(\cdot)$, $h(S)$ is the penalty cost of client set $S \subseteq \mathcal{D}$ and $h(\phi) = 0$. We consider the metric k-FLPSP, this means that the connection cost satisfies symmetry triangle inequality, such as $c_{ij} \leq c_{ij'} + c_{i'j'} + c_{i'j}$, for $i, i' \in \mathcal{F}$, $j, j' \in \mathcal{D}$. Our aim is to select the facility subset of each level to open and connect all clients to the open facilities such that the total cost of k-FLPSP is minimized. The k-FLPSP can be formulated as the following integer programming:

$$
\begin{aligned}
\min \quad & \sum_{l=1}^{k} \sum_{i_l \in \mathcal{F}^l} f_{i_l} y_{i_l} + \sum_{j \in \mathcal{D}} \sum_{p \in P} c_{jp} x_{jp} + \sum_{S \subseteq \mathcal{D}} h(S) z_S \qquad (1) \\
s.t. \quad & \sum_{p \in P} x_{jp} + \sum_{S \subseteq \mathcal{D}, j \in S} z_S \geq 1, \qquad \forall j \in \mathcal{D}, \\
& \sum_{p: i_l \in p} x_{jp} \leq y_{i_l}, \qquad \forall j \in \mathcal{D},\ i_l \in \mathcal{F}^l,\ l = 1, \cdots, k, \\
& x_{jp} \geq 0, \qquad \forall p \in P,\ j \in \mathcal{D}, \\
& y_{i_l} \geq 0, \qquad \forall i_l \in \mathcal{F}^l,\ l = 1, \cdots, k, \\
& z_S \geq 0, \qquad \forall S \subseteq \mathcal{D}.
\end{aligned}
$$

where $x_{jp} = 1$ if client j is connected to path p, otherwise variable $x_{jp} = 0$. When the facility on level l is open $y_{i_l} = 1$, and 0 otherwise. Variable z_S equals 1 if the client set S is penaltied, and 0 otherwise. The first constraint implies that a client can be connected to a path p or be punished at some set $S \subseteq \mathcal{D}$. The second constraint indicates that if a client is connected to a path p, all the facilities of the path p must be opened.

The relaxation of the integer programming (1) is given as follows:

$$
\begin{aligned}
\min \quad & \sum_{l=1}^{k} \sum_{i_l \in \mathcal{F}^l} f_{i_l} y_{i_l} + \sum_{j \in \mathcal{D}} \sum_{p \in P} c_{jp} x_{jp} + \sum_{S \subseteq \mathcal{D}} h(S) z_S \qquad (2)\\
s.t. \quad & \sum_{p \in P} x_{jp} + \sum_{S \subseteq \mathcal{D}, j \in S} z_S \geq 1, \qquad \forall j \in \mathcal{D},\\
& \sum_{p: i_l \in p} x_{jp} \leq y_{i_l}, \qquad \forall j \in \mathcal{D},\ i_l \in \mathcal{F}^l,\ l = 1, \cdots, k,\\
& x_{jp} \geq 0, \qquad \forall p \in P,\ j \in \mathcal{D},\\
& y_{i_l} \geq 0, \qquad \forall i_l \in \mathcal{F}^l,\ l = 1, \cdots, k,\\
& z_S \geq 0, \qquad \forall S \subseteq \mathcal{D}.
\end{aligned}
$$

The dual program corresponding to the linear programming relaxation (2) is the following:

$$
\begin{aligned}
\max \quad & \sum_{j \in \mathcal{D}} \alpha_j\\
s.t. \quad & \alpha_j \leq c_{jp} + \sum_{i_l \in P} \beta_{i_l j}, \qquad \forall p \in P, j \in \mathcal{D},\\
& \sum_{j \in S} \alpha_j \leq h(S), \qquad \forall S \subseteq \mathcal{D},\\
& \sum_{j \in \mathcal{D}} \beta_{i_l j} \leq f_{i_l}, \qquad \forall i_l \in \mathcal{F}^l,\ l = 1, \cdots, k,\\
& \alpha_j \geq 0, \qquad \forall j \in \mathcal{D},\\
& \beta_{i_l j} \geq 0, \qquad \forall j \in \mathcal{D}, i_l \in \mathcal{F}^l,\ l = 1, \cdots, k.
\end{aligned}
$$

where α_j is the total spending of client j in the process, and $\beta_{i_l j}$ indicates the opening cost of facility i_l given by client j.

3 Primal-dual approximation algorithm

Different from the primal-dual approximation algorithm of Li et al. (2012), we change the way of last opening facility set to get much more tight result for k-FLPSP.

In order to understand the situation of each client when we run primal-dual algorithm, we give some definitions. Initially, all clients in $\mathcal{D}$ are unfrozen, if client j is connected to path p of which each facility of path p is open, client j is frozen. When $\sum_{j \in \mathcal{D}} \beta_{i_l j} = f_{i_l}$, the facilities i_l is open. For some path $p = (i_1 \in \mathcal{F}^1, \cdots, i_l \in \mathcal{F}^l)$, if facilities $i_1, i_2, \cdots, i_{l-1}$ are all open and $\alpha_j = c_{jp} + \sum_{l'=1}^{l} \beta_{i_{l'} j}$, client j reaches facility $i_l \in \mathcal{F}^l$. If facilities $i_l \in \mathcal{F}^l$ is open, client j leaves i_l and makes contribution to connection cost for the facility of the $l+1$th level, $1 \leq l < k$ or $l = k$, client j is connected. The dual variable α_j of all unfrozen clients $j \in \mathcal{D}$ increase uniformly at unit rate of time t. $\beta_{i_l j}$ increases at same rate of α_j when client $j \in \mathcal{D}$ reaches unopened facility $i_l \in \mathcal{F}^l$. All dual variables $\beta_{i_l j}(j \in \mathcal{D})$ stop increasing when facility i_l is open. Time will stop until there is no unfrozen client. Facility $i_l \in \mathcal{F}^l$ is temporarily open when $\sum_{j \in \mathcal{D}} \beta_{i_l j} = f_{i_l}$, t_{i_l} is the moment of facility i_l temporarily open. The predecessor of i_l will be the facility in the $l-1$th level via which i_l was for the first time reached by a client, i.e. $pred(i_l) := \arg\min_{i \in \mathcal{F}^{l-1}} \{t_i + c_{ii_l}\}$, t_i is time of facility i temporarily open, the predecessor of $i_1 \in \mathcal{F}^1$ is the client which is closest to i_1, $t_{pred(i_1)} := 0$.

Algorithm 1

Step 1. Initialization: we introduce the notion of time t, Initially, $t = 0$, set $\alpha_j = 0 (\forall j \in \mathcal{D})$, $\beta_{i_l j} = 0 (i_l \in \mathcal{F}^l, 1 \le l \le k, j \in \mathcal{D})$, all facilities are unopened and all clients are unfrozen. $\tilde{S}$ is the set of penalized clients, initially $\tilde{S} = \emptyset$. With increase of time, there will occur three events:

Event 1. Facility i_k is temporarily open, we freeze those clients j with $\beta_{i_k j} > 0$ and let those clients be connected to i_k, facility i_k is the connecting witness for client j. The associated path of i_k is $p(i_k) = (i_1, i_2, \cdots, i_k), i_l = pred(i_{l+1}), \forall 1 \le l \le k-1$, the predecessor of i_1 is the client j_{i_1}. The neighborhood of facility i_k are the clients which pay for connected path $p(i_k)$, i.e. $N(i_k) := \{j \in \mathcal{D} | \beta_{i_l j} > 0, i_l \in p(i_k)\}$.

Event 2. When unfrozen client j reaches temporarily open facility i_k, freeze client j, we call facility i_k the connecting witness of client j.

Event 3. For some subset $S \subseteq \mathcal{D}$, if $\sum_{j \in S} \alpha_j = h(S)$, freeze all unfrozen clients in S, set $\tilde{S} := \tilde{S} \cup S$, we call clients in $\tilde{S}$ the penalized clients.

When all clients are frozen, Step 1 terminates. If several events occur simultaneously, the algorithm executes these events in an arbitrary order.

Step 2. We choose set $\tilde{S}$ in Step 1 as the penalized client set, the temporarily open facility set on level k is $\tilde{\mathcal{F}}^k$ and sort these facilities temporarily according to the open time t with nondecreasing order. Let $\bar{\mathcal{F}}^k$ be finial open facility set, we add facility i_k to $\bar{\mathcal{F}}^k$ with order if and only if there is no facility $i'_k \in \bar{\mathcal{F}}^k$, which is satisfied with $c_{i_k i'_k} \le 3t_{i_k}$, upset $\bar{\mathcal{F}}^k := \bar{\mathcal{F}}^k \cup \{i_k\}$, open facility $i_k \in \bar{\mathcal{F}}^k$ and the associated path $p(i_k)$. For each client $j \in N(i_k)$, if $i_k \in \bar{\mathcal{F}}^k$, connect client j to the associated path $p(i_k)$ of facility i_k. Otherwise, connect client j to the associated path $p(i'_k)$ of the closest facility i'_k.

Lemma 1 Li et al. (2012) *Algorithm 1 can be solved in polynomial time.*

Lemma 2 *If facility $i_k, i'_k \in \bar{\mathcal{F}}^k$, then $N(i_k) \cap N(i'_k) = \emptyset$.*

Proof We assume that $N(i_k) \cap N(i'_k) \neq \emptyset$, there exists a client $j \in N(i_k) \cap N(i'_k)$. Suppose that facility i_k is open after the opening of facility i'_k, then

$$c_{i_k i'_k} > 3t_{i_k} > 2t_{i_k} > t_{i_k} + t_{i'_k}.$$

Due to $j \in N(i_k)$, there exists a facility $i_l \in p(i_k)$ with $\beta_{i_l j} > 0$, and there exists a path p_{i_l} from $\mathcal{F}^1$ to facility i_l, where $c_{jp_{i_l}} \le t_{i_l}$. We consider the restructure of the path

Algorithm 1

Input: connection cost c_{ij}, opening cost f_i, submodular penalty function $h(\cdot)$, $i \in \mathcal{F}$, $j \in \mathcal{D}$.
Output: feasible integer solution.

```
Initialization: α_j = 0, β_ij = 0, the penalty set S = ∅, t = 0.
for j ∈ D do
    if i_k is temporarily open then
        freeze those clients j with β_{i_k j} > 0 and let those clients be connected to i_k;
    end if
    if unfrozen client j reaches temporarily open facility i_k then
        freeze client j and connect client j to facility i_k.
    end if
    if For subset S ⊆ D, Σ_{j ∈ S} α_j = h(S) then
        freeze all unfrozen clients in S, update S̃ := S̃ ∪ S.
    end if
end for
for temporarily opened facility set F̃^k on level k do
    if there is no facility i_{k'} ∈ F̄^k, which is satisfied with c_{i_k i_{k'}} ≤ 3t_{i_k} then
        open facility i_k ∈ F̄_k and the associated path p(i_k);
    end if
end for
```

p_{i_k} as following: firstly, along the path p_{i_l} from $\mathcal{F}^1$ to i_l, then along the associated path $p(i_k)$ of facility i_k from i_l to i_k. According to the definition of predecessor, $c_{jp_{i_k}} \leq t_{i_k}$. Analogously, there exists a path $p_{i'_k}$ of i'_k from $\mathcal{F}^1$ to i'_k with $c_{jp_{i'_k}} \leq t_{i'_k}$. In conclusion, $c_{ji_k} + c_{ji'_k} \leq c_{jp_{i_k}} + c_{jp_{i'_k}} \leq t_{i_k} + t_{i'_k} < c_{i_k i'_k}$, this conflict with triangle inequality. □

Lemma 3 *If $j \in N(i_k) \backslash \tilde{S}$, $i_k \in \bar{\mathcal{F}}^k$, then $t_{i_k} \leq 2\alpha_j$.*

Proof We assume that $t_{i_k} > 2\alpha_j$, since client $j \in N(i_k) \backslash \tilde{S}$, from the proof of Lemma 2, we know that there exists a path p_{i_k} of facility i_k from $\mathcal{F}^1$ to facility i_k with $c_{jp_{i_k}} \leq t_{i_k}$. Client j firstly reaches the open facility $i'_k \in \mathcal{F}^k$. Then we have $t_{i'_k} \leq \alpha_j$, there also exists a path $p_{i'_k}$, such that $c_{jp_{i'_k}} \leq \alpha_j$.

Case 1. If facility i'_k is open, then $c_{i_k i'_k} \leq c_{jp_{i'_k}} + c_{jp_{i_k}} \leq t_{i_k} + \alpha_j < \frac{3}{2} t_{i_k} < 3t_{i_k}$.

Case 2. If facility i'_k is not open, there exists an open facility $i''_k \in \bar{\mathcal{F}}^k$, $c_{i'_k i''_k} \leq 3t_{i'_k}$ and $t_{i''_k} \leq t_{i'_k} \leq \alpha_j < t_{i_k}$. So we can get

$$\begin{aligned} c_{i_k i''_k} &\leq c_{jp_{i_k}} + c_{jp_{i'_k}} + c_{i'_k i''_k} \\ &\leq t_{i_k} + \alpha_j + 3t_{i'_k} \\ &\leq t_{i_k} + 4\alpha_j \\ &< 3t_{i_k}. \end{aligned}$$

This conflict with the condition of facility opening in Algorithm 1. □

Lemma 4 Li et al. (2012) *At any moment t in Step 1 of Algorithm 1, the cost of penalized clients set $\tilde{S}$ is:*

$$\sum_{j \in \tilde{S}} \alpha_j(t) = h(\tilde{S}).$$

$\alpha_j(t)$ is equal to α_j which α_j is at moment t, and α_j increases uniformly with time t until client j is frozen.

In order to get tight bound of total cost of k-FLPSP, we analyze the opening cost of facilities in detail.

Lemma 5 $f(p(i_k)) \leq \sum_{i \in p(i_k)} \sum_{j \in N(i_k) \backslash \tilde{S}} \beta_{ij} + \sum_{j \in N(i_k) \cap \tilde{S}} \alpha_j,\ i_k \in \bar{\mathcal{F}}^k.$

Proof

$$\begin{aligned} f(p(i_k)) &= \sum_{i \in p(i_k)} \sum_{j \in N(i_k)} \beta_{ij} \\ &= \sum_{i \in p(i_k)} \sum_{j \in N(i_k) \backslash \tilde{S}} \beta_{ij} + \sum_{j \in N(i_k) \cap \tilde{S}} \beta_{ij} \\ &\leq \sum_{i \in p(i_k)} \sum_{j \in N(i_k) \backslash \tilde{S}} \beta_{ij} + \sum_{j \in N(i_k) \cap \tilde{S}} \alpha_j. \end{aligned}$$

□

Lemma 6 *On Step 2 of the Algorithm 1, client j is connected to facility i_k and i_k is not a connecting witness of client j, while j makes no contribution to facilities of path $p(i_k)$, i.e. $M(i_k) := \{j \in \mathcal{D} \backslash N(i_k) \cup \tilde{S} | i(j) \neq i_k, \bar{i}(j) = i_k\}$. If facility $i(j)$ is a connecting witness of client j at Step 1, client j at Step 2 is connected to facility $\bar{i}(j)$, then*

$$c_{jp(i_k)} \leq 6\alpha_j,\ j \in M(i_k).$$

Proof We assumed that client j is the connecting witness of i'_k at Step 1, then there exists a path $p_{i'_k}$, which $c_{jp_{i'_k}} \leq \alpha_j$ and $t_{i'_k} \leq \alpha_j$. According to triangle inequality, we have:

$$\begin{aligned} c_{jp(i'_k)} &= c_{ji_1} + \sum_{l'=2}^{k} c_{i_{l'-1} i_{l'}} \\ &\leq c_{jp_{i'_k}} + 2 \sum_{l'=2}^{k} c_{i_{l'-1} i_{l'}} \\ &\leq \alpha_j + 2t_{i'_k} \\ &\leq 3\alpha_j. \end{aligned}$$

If $i'_k \in \bar{\mathcal{F}}^k$, then $c_{jp(i_k)} \leq c_{jp(i'_k)} \leq 3\alpha_j$.

If $i'_k \notin \bar{\mathcal{F}}^k$, there exist facility $i''_k \in \bar{\mathcal{F}}^k$ with $c_{i'_k i''_k} \leq 3t_{i'_k}$, and $t_{i''_k} \leq t_{i'_k}$, $N(i'_k) \cap N(i''_k) \neq \emptyset$. Let client $j \in N(i'_k) \cap N(i''_k)$, from the proof above, we know that there exist a associated path $p(i''_k) := (i''_1, \cdots, i''_{k-1}, i''_k = i'')$ of facility i''_k. The connection cost of client j satisfies the following inequality:

$$\begin{aligned} c_{jp(i_k)} &\leq c_{jp(i''_k)} \\ &= c_{jp(i''_1)} + \textstyle\sum_{l=2}^{k} c_{i''_{l-1}i''_l} \\ &\leq c_{jp(i'_k)} + c_{i''_k i''_k} + 2\textstyle\sum_{l=2}^{k} c_{i''_{l-1}i''_l} \\ &\leq c_{jp(i'_k)} + c_{i''_k i''_k} + 2t_{i''_k} \\ &\leq \alpha_j + 5t_{i'_k} \\ &\leq 6\alpha_j. \end{aligned}$$

□

Lemma 7 *Client j is connected to facility i_k at Step 2 of Algorithm 1 and i_k is the connecting witness of client j at Step 1, while client j makes no contribution to the opening of facilities of path $p(i_k)$, i.e. $j \in M'(i_k)$, $M'(i_k) := \{j \in \mathcal{D}\backslash N(i_k) \cup \tilde{S} | i(j) = i_k, \bar{i}(j) = i_k\}$, $i(j)$ and $\bar{i}(j)$ are the same with Lemma 6, so we have*

$$c_{jp(i_k)} \leq 3\alpha_j,\ j \in M'(i_k).$$

Proof Since client j is the connecting witness of facility i_k at Step 1, from Lemma 6 we know that: $c_{jp(i_k)} \leq 3\alpha_j$, $j \in M'(i_k)$. □

Lemma 8 *For $i_k \in \bar{\mathcal{F}}^k$,*

$$3f(p(i_k)) + \sum_{j \in N(i_k)\backslash\tilde{S}} c(jp(i_k)) \leq 6 \sum_{j \in N(i_k)\backslash\tilde{S}} \alpha_j + 3 \sum_{j \in N(i_k)\cap\tilde{S}} \alpha_j.$$

Proof For client $j \in N(i_k)\backslash\tilde{S}$, the contribution of client j to the facilities of path $p(i_k)$, facility i_l is the first facility which client j makes contribution to, i.e. $i_l(j) := \{i_m \in p(i_k) | \beta_{i_m j} > 0, \beta_{i_m j} = 0, \forall 1 \leq n < m\}$. There exists a path $p_{i_l(j)}$ from $\bar{\mathcal{F}}^1$ to facility i_l with $c_{jp_{i_l(j)}} \leq t_{i_l}$. Let $A = \sum_{l'=2}^{l} c_{i_{l'-1}i_{l'}} + \sum_{l'=2}^{k} c_{i_{l'-1}i_{l'}}$

$$\begin{aligned} &3f(p(i_k)) + \sum_{j \in N(i_k)\backslash\tilde{S}} c_{jp(i_k)} \\ &\leq 3 \sum_{j \in N(i_k)\backslash\tilde{S}} \sum_{i \in p(i_k)} \beta_{ij} + 3 \sum_{j \in N(i_k)\cap\tilde{S}} \alpha_j + \sum_{j \in N(i_k)\backslash\tilde{S}} c_{jp(i_k)} \\ &\leq 3 \sum_{j \in N(i_k)\backslash\tilde{S}} \sum_{l'=1}^{k} \beta_{i_{l'} j} + \sum_{j \in N(i_k)\backslash\tilde{S}} (c_{jp_{i_l(j)}} + A) + 3 \sum_{j \in N(i_k)\cap\tilde{S}} \alpha_j \end{aligned}$$

$$
\begin{aligned}
&\le \sum_{j\in N(i_k)\setminus\tilde{S}} (3\sum_{l'=1}^{k}\beta_{i_{l'}j} + c_{jp_{i_l(j)}} + A) + 3\sum_{j\in N(i_k)\cap\tilde{S}} \alpha_j \\
&\le \sum_{j\in N(i_k)\setminus\tilde{S}} [(\sum_{l'=1}^{k}\beta_{i_{l'}j} + c_{jp_{i_l(j)}}) + 2A] + 3\sum_{j\in N(i_k)\cap\tilde{S}} \alpha_j \\
&\le \sum_{j\in N(i_k)\setminus\tilde{S}} [(\sum_{l'=1}^{k}\beta_{i_{l'}j} + c_{jp_{i_l(j)}}) + 2t_{i_k}] + 3\sum_{j\in N(i_k)\cap\tilde{S}} \alpha_j \\
&\le \sum_{j\in N(i_k)\setminus\tilde{S}} (2\alpha_j + 2t_{i_k}) + 3\sum_{j\in N(i_k)\cap\tilde{S}} \alpha_j \\
&\le 6\sum_{j\in N(i_k)\setminus\tilde{S}} \alpha_j + 3\sum_{j\in N(i_k)\cap\tilde{S}} \alpha_j .
\end{aligned}
$$

Lemma 8 is proved. □

Theorem 1 *For k-FLPSP, the approximation ratio of Algorithm 1 is 6.*

Proof F, C, h are the opening, connection and penalty cost, respectively.

$$
\begin{aligned}
&3(F+h)+C \\
&= 3(\sum_{i_k\in\bar{\mathcal{F}}^k} f(p(i_k)) + \sum_{j\in\tilde{S}}\alpha_j) + \sum_{i_k\in\bar{\mathcal{F}}^k}(\sum_{j\in M(i_k)} c_{jp(i_k)} + \sum_{j\in M'(i_k)} c_{jp(i_k)} \\
&\quad + \sum_{j\in N(i_k)\setminus\tilde{S}} c_{jp(i_k)}) \\
&= \sum_{i_k\in\bar{\mathcal{F}}^k}(3f(p(i_k)) + \sum_{j\in N(i_k)\setminus\tilde{S}} c_{jp(i_k)}) + \sum_{i_k\in\bar{\mathcal{F}}^k}(\sum_{j\in M(i_k)} c_{jp(i_k)} \\
&\quad + \sum_{j\in M'(i_k)} c_{jp(i_k)}) + 3\sum_{j\in\tilde{S}}\alpha_j \\
&\le \sum_{i_k\in\bar{\mathcal{F}}^k}(6\sum_{j\in N(i_k)\setminus\tilde{S}}\alpha_j + 3\sum_{j\in N(i_k)\cap\tilde{S}}\alpha_j) + \sum_{i_k\in\bar{\mathcal{F}}^k}(6\sum_{j\in M(i_k)}\alpha_j \\
&\quad + 3\sum_{j\in M'(i_k)}\alpha_j) + 3\sum_{j\in\tilde{S}}\alpha_j \\
&\quad \le 6\sum_{i_k\in\bar{\mathcal{F}}^k}(\sum_{j\in N(i_k)\setminus\tilde{S}}\alpha_j + \sum_{j\in M(i_k)}\alpha_j + \sum_{j\in M'(i_k)}\alpha_j) \\
&\quad + 3\sum_{i_k\in\bar{\mathcal{F}}^k}\sum_{j\in N(i_k)\cap\tilde{S}}\alpha_j + 3\sum_{j\in\tilde{S}}\alpha_j \\
&\quad \le 6\sum_{j\in\mathcal{D}\setminus\tilde{S}}\alpha_j + 6\sum_{j\in\tilde{S}}\alpha_j .
\end{aligned}
$$

Which proves the Theorem. □

We restructure the k-FLPSP and firstly present the method of restructuring in the following, then we give some notations. Lastly, we use greedy augmentation to improve the initial solution.

Restructures: Consider the k-level facilities of the k-FLPSP. For every $i_1 \in \mathcal{F}^1$, let $P_{i_1} = \{i_1, i_2 \in \mathcal{F}^2, \cdots, i_k \in \mathcal{F}^k\}$. Let $\mathcal{D}$ be the set of clients, and the set of the "facility" be $P := \{p : p \in P_{i_1}, \forall i_1 \in \mathcal{F}^1\}$, such as $p := (i_1 \in \mathcal{F}^1, i_2 \in \mathcal{F}^2, \cdots, i_k \in \mathcal{F}^k)$. the opening cost of each facility of each path p is calculated as follows:

$$f_i' := \begin{cases} f_i, \ when\ i\ firstly\ be\ computed\ in\ a\ path, \\ 0, \ \ otherwise. \end{cases}$$

This means that the opening cost of each facility can just be computed once. The opening cost of each path $p \in P$ is $f_p' = \sum_{i \in p} f_i'$. In order to improve the approximation ratio of the current solution, we use greedy augmentation technique. By executing some local search operations from an arbitrary integer feasible solution, we can improve the current approximation ratio. Let $\mathcal{F}_0$ be the set of open facilities, we randomly choose a facility of each level from $\mathcal{F}_0$, enumerate all paths, let P_0 be the path set and S_0 be the set of rejected clients in the current solution, let F_0, C_0, h_0 be the current opening, connection and penalty costs. $c_j^{\mathcal{F}_0}$ is the connection cost of the client to its closest open path in current solution, $C(\mathcal{F}_0 \cup p)$ is the connection cost of all clients after adding path p to facility set $\mathcal{F}_0$. $gain(S)$ and $gain(p)$ can be calculated as follows:

$$gain(S) = \sum_{j \in S \setminus S_0} c_j^{\mathcal{F}_0} + h_0 - h_S, \qquad S_0 \subseteq S;$$
$$gain(p) = C_0 - C(\mathcal{F}_0 \cup p) - f_p', \qquad p \in P \setminus P_0.$$

We will try to improve the current solution by one of the two local search operations: either *replacing* S_0 with a larger set S ($S_0 \subseteq S \subseteq \mathcal{D}$) or *incorporating* a path p from the path set P minus the current solution.

First, we extend the definition of $gain(p)$ for $p \in P_0$, and set $gain(p) = 0$ for $p \in P_0$. We will improve the approximation ratio by iteration upset the set penaltied client or facility set until there is no set satisfies the condition. The following is the new greedy augmentation algorithm.

Algorithm 2

Step 1. The arbitrary initial feasible solution SOL_0 with open facilities set $\mathcal{F}_0$, the path set is P_0 and rejected clients set S_0. Initialization $s := 1$.

Step 2. Find a path $p^* \in P \setminus P_{s-1}$ to maximize $\left\{ \frac{gain(p)}{f_p'} \right\}$, where P_{s-1} is the path set of current solution, when $s = 1$, $P_{s-1} = P_0$, let

$$r_s := \frac{gain(p^*)}{f_{p^*}'} = \max_{p \in P \setminus P_{s-1}} \left\{ \frac{gain(p)}{f_p'} \right\}.$$

We can calculate

$$m_k = \max_{S \subseteq \mathcal{D} \backslash S_{s-1}} gain(S_{s-1} \cup S).$$

Where S_{s-1} is the penaltied clients set in the $(s-1)$th iteration, S is client subset which $S \subseteq \mathcal{D} \backslash S_{s-1}$. If $m_s > 0$, find a set of unrejected clients S^* which is the optimal solution of the following optimization problem

$$r_s' := \max_{S \subseteq \mathcal{D} \backslash S_{s-1}} \left\{ \frac{gain(S_{s-1} \cup S)}{h(S_{s-1} \cup S) - h(S_{s-1})} \right\},$$

otherwise, set $r_s' := 0$.

Step 3. If $\max\{r_s, r_s'\} \leq 0$, the algorithm terminates, and outputs a feasible solution SOL_{s-1} (SOL_{s-1} is the solution which means we can't found any client subset or facility subset to be added, so the Algorithm terminates in the sth iteration. The Algorithm outputs the $(s-1)th$ solution.) with open facilities set P_{s-1} and the rejected clients set S_{s-1}.

Step 4. If $r_s \geq r_s'$, open the path p and maintain the rejected clients set, meaning that we get a feasible solution SOL_{s-1} with $P_s := P_{s-1} \cup \{p\}$ and $S_s := S_{s-1}$; otherwise, extending the rejected clients set to $S_{s-1} \cup S^*$ and maintaining the opening facilities set, meaning that we get a feasible solution SOL_{s-1} with $P_s := P_{s-1}$ and $S_s := S_{s-1} \cup S^*$. Update $s := s + 1$, and return to Step 1.

In the following, we present the whole algorithm.

Algorithm 3

Step 0. Given an instance of k-FLPSP, scale the facility cost and penalty function by a factor $\delta = 0.8571$.

Step 1. Through running the primal-dual algorithm ($Algorithm$ 1) on the scaled instance to obtain a feasible solution SOL_0 to the original instance.

Step 2. Let SOL_0 be the initial feasible solution, apply the greedy augmentation algorithm ($Algorithm$ 2) to get the solution $S\check{O}L$, Where $S\check{O}L$ is the solution which we obtain.

4 Analysis

We give the proof in Lemma 9–12 to prove that our algorithm can obtain a penaltied client subset in polynomial time or we can enumerate all the paths, and we also give the approximation ratio of Algorithm 3 in the following.

Lemma 9 Fujishige (2005) *Suppose that* f*:* $2^{\mathcal{D}} \rightarrow R$ *is a non-negative function with* $f(\emptyset) = 0$*, and* g*:* $2^{\mathcal{D}} \rightarrow R$ *is a nonnegative function satisfying* $g(\emptyset) = 0$ *and* $g(S) > 0$ *for some* $S \subseteq \mathcal{D}$*. Define the following minimum-ratio problem*

Algorithm 2

Input: Initial feasible solution SOL_0, open facilities set $\mathcal{F}_0$, rejected clients set S_0.
Output: integer feasible solution.

for $p^* \in P \backslash P'$ **do**

$$r_s := \frac{gain(p^*)}{f'_{p^*}} = \max_{p \in P \backslash P_{s-1}} \left\{ \frac{gain(p)}{f'_p} \right\}.$$

$$m_s = \max_{S \subseteq \mathcal{D} \backslash S_{s-1}} gain(S_{s-1} \cup S),$$

  if $m_s > 0$ **then**
    find a set of unrejected clients S^* which is the optimal solution of the following optimization problem

$$r'_s := \max_{S \subseteq \mathcal{D} \backslash S_{s-1}} \left\{ \frac{gain(S_{s-1} \cup S)}{h(S_{s-1} \cup S) - h(S_{s-1})} \right\},$$

  else
    $r'_s = 0$.
    if $\max\{r_s, r'_s\} \leq 0$ **then**
      outputs a feasible solution SOL_{s-1}.
    end if
    if $r_s \geq r'_s$ **then**
      open the path p and maintain the rejected clients set.
    else
      extending the rejected clients set to $S_{s-1} \cup S^*$ and maintaining the opening facilities set.
    end if
  end if
end for

Algorithm 3

Input: Given an instance of k-FLPSP, $\delta = 0.8571$.
Output: integer feasible solution of k-FLPSP.

Scaling the facility opening cost, penalty cost of given instance of k-FLPSP and call Algorithm 1 to get the SOL_0.

Let SOL_0 be the initial solution of Algorithm 2, run Algorithm 2 and obtain solution $S\check{O}L$.

$$\begin{aligned} \min \quad & \frac{f(S)}{g(S)} \\ s.t. \quad & g(S) > 0, \\ & S \subseteq \mathcal{D}. \end{aligned} \tag{4.1}$$

The Lagrangian function for f and $-g$ associated with (4.1) is given by

$$L(\lambda, S) = f(S) - \lambda g(S),$$

for $\lambda \geq 0$ *and* $S \subseteq \mathcal{D}$*. Then a nonnegative* $\hat{\lambda}$ *is the minimum value of* (4.1) *if and only if*

$$\min_{S\subseteq\mathcal{D}} L(\lambda, S) = 0, \quad 0 \leq \lambda \leq \hat{\lambda},$$
$$\min_{S\subseteq\mathcal{D}} L(\lambda, S) = 0, \quad \hat{\lambda} < \lambda.$$

Furthermore, the minimum-ratio problem (4.1) *is solvable in polynomial time if* $\min_{S\subseteq\mathcal{D}} L(\lambda, S)$ *is solvable in polynomial time for any* $\lambda \geq 0$.

Lemma 10 Li et al. (2013) *For Step 2 of Algorithm 2,*

$$m_s = \max_{S\subseteq\mathcal{D}\setminus S_{s-1}} \{gain(S_{s-1} \cup S)\}$$

is solvable in polynomial time and the maximum-ratio problem

$$r'_s := \max_{S\subseteq\mathcal{D}\setminus S_{s-1}} \left\{ \frac{gain(S_{s-1} \cup S)}{P(S_{s-1}) - P_{S_{s-1}}} \right\}$$

is solvable in polynomial time when $m_k > 0$.

Lemma 11 *The path* p *can be found in polynomial time in Algorithm 2.*

Proof There are total $|\mathcal{F}^1||\mathcal{F}^2|\cdots|\mathcal{F}^k|$ paths after we restructure the k-FLPSP, for any constant k. The number of paths of the final solution is at most $|\mathcal{D}||\mathcal{F}^1||\mathcal{F}^2|\cdots|\mathcal{F}^k|$ minus the number of paths which in the initial solution SOL_0. So Step 2 of the Algorithm 2 can be finished in polynomial time. □

Lemma 12 *The Algorithm 2 can be completed in polynomial time.*

Proof In the Step 1 of Algorithm 2, we know that the Algorithm of Li et al. (2013) can be complete in the polynomial time. For Lemmas 9–11, it is possible in polynomial time to find a path or a subset of clients to be punished. So the Algorithm 2 can be executed in polynomial time. □

Lemma 13 $\sum_{p\in P_{SOL}} gain(p) + gain(S_{SOL} \cup S_s) \geq C_s - (F_{SOL} + h_{SOL} + C_{SOL})$, *where* F_{SOL}, C_{SOL}, h_{SOL} *are the opening, connection and penalty costs of arbitrary integer solution* SOL *of k-FLPSP,* P_{SOL} *is the path set of solution* SOL*.* S_s *is the penaltied clients set in current solution* SOL_s *and* S_{SOL} *is the penaltied clients set in solution* SOL*.*

Proof For arbitrary path $p \in \mathcal{F}_{SOL}$, $\mathcal{D}_{SOL}(p)$ is the set of clients which are assigned to path p in SOL. For arbitrary client $j \in \mathcal{D}_{SOL}(p)$, let $\sigma(j)$ and $\sigma_{SOL}(j)$ be the paths servicing j in the current solution SOL_s(the output feasible solution of the sth iteration in Algorithm 3) and solution SOL, respectively. Let $c_{jp} = c_{ji_1} + \sum_{t=2}^{k} c_{i_{t-1}i_t}$.

By including path p and reassigning the clients in $\mathcal{D}_{SOL}(p)\backslash S_s$ to p, we can update the current solution. The $gain'(p)$ is the resulted saving in cost, i.e.,

$$gain'(p) = \sum_{j\in\mathcal{D}_{SOL}(p)\backslash S_s} (c_{j\sigma(j)} - c_{j\sigma_{SOL}(j)}) - f'_p.$$

Note that it may occur that $gain'(p) < 0$. From the definition of $gain(p)$, we know that $gain(p) > gain'(p)$. For $j \in \mathcal{D}_{SOL}(p)$, let $p = \sigma_{SOL}(j)$. Let P_{SOL} be the path set of solution SOL, $S = S_{SOL} \cup S_s$. We have

$$\begin{aligned}
&\sum_{p\in P_{SOL}} gain'(p) + gain(S)\\
&= \sum_{p\in P_{SOL}} (-f'_p + \sum_{j\in\mathcal{D}_{SOL}(p)\backslash S_s} (c_{j\sigma(j)} - c_{j\sigma_{SOL}(j)})) + \sum_{j\in S\backslash S_s} c_{j\sigma(j)}\\
&\quad - (h(S) - h(S_s))\\
&= -\sum_{p\in P_{SOL}} f'_p + (\sum_{p\in P_{SOL}} \sum_{j\in\mathcal{D}_{SOL}(p)\backslash S_s} c_{j\sigma(j)} + \sum_{j\in S\backslash S_s} c_{j\sigma(j)})\\
&\quad - \sum_{p\in P_{SOL}} \sum_{j\in\mathcal{D}_{SOL}(p)\backslash S_s} c_{j\sigma_{SOL}(j)} - (h(S) - h(S_s))\\
&\geq C_s - F_{SOL} - h_{SOL} - C_{SOL}.
\end{aligned}$$

So we obtain the inequality □

From the definition of r_{s+1} and r'_{s+1} in Algorithm 3, and Lemma 13, we have the following lemma.

Lemma 14 $\max\{r_{s+1}, r'_{s+1}\} \geq \frac{C_s - F_{SOL} - h_{SOL} - C_{SOL}}{F_{SOL} + h_{SOL}}$.

Proof Firstly, we assume $h(S_{SOL} \cup S_s) - h(S_s) > 0$. Otherwise, with slight simplification the following proof can be adapted to the special cases with $f'_p = 0$ for paths $p \in \mathcal{F}_{SOL}$ or $h(S_{SOL} \cup S_s) - h(S_s) = 0$. We consider the special case that there exists some special paths with $f'_p = 0$, when $f'_p = 0$, this means that the cost of facilities in p have been computed before, let P'' be the facility set of all the special paths, we can calculate as follows:

$$\begin{aligned}
&\max_{p\in P_{SOL}\backslash P''} \left\{\frac{gain(p)}{f'_p}\right\} F_{SOL} + \sum_{p\in P''} gain(p)\\
&= \max_{p\in P_{SOL}\backslash P''} \left\{\frac{gain(p)}{f'_p}\right\} \sum_{p\in P_{SOL}\backslash P''} \sum_{i\in p} f'_i + \sum_{p\in P''} gain(p)\\
&= \sum_{p\in P_{SOL}\backslash P''} \max_{p\in P_{SOL}} \left\{\frac{gain(p)}{\sum_{i\in p} f'_i}\right\} \sum_{i\in p} f'_i + \sum_{p\in P''} gain(p)
\end{aligned}$$

$$\geq \sum_{p \in P_{SOL} \setminus P''} \left(\frac{gain(p)}{\sum_{i \in p} f_i'} \sum_{i \in p} f_i' \right) + \sum_{p \in P''} gain(p)$$
$$= \sum_{p \in P_{SOL}} gain(p).$$

We can obtain the result by corresponding adjustment.

Now we consider the general case. From the above definition, we have

$$\begin{aligned} \max_{p \in P_{SOL}} \left\{ \frac{gain(p)}{f_p'} \right\} F_{SOL} &= \max_{p \in P_{SOL}} \left\{ \frac{gain(p)}{f_p'} \right\} \sum_{p \in P_{SOL}} \sum_{i \in p} f_i' \\ &= \sum_{p \in P_{SOL}} \max_{p \in P_{SOL}} \left\{ \frac{gain(p)}{\sum_{i \in p} f_i'} \right\} \sum_{i \in p} f_i' \\ &\geq \sum_{p \in P_{SOL}} \left(\frac{gain(p)}{\sum_{i \in p} f_i'} \sum_{i \in p} f_i' \right) \\ &= \sum_{p \in P_{SOL}} gain(p). \end{aligned} \tag{4.2}$$

The reason for the second equality holds is that the opening cost of each facility of path p is just computed once, then the connection cost of the opened facilities is 0 after it is first opened in a path. So the total opening cost of facility set of solution SOL is $F_{SOL} = \sum_{p \in P_{SOL}} \sum_{i \in p} f_i'$.

Due to the submodularity of $h(\cdot)$, we have

$$\begin{aligned} \frac{gain(S_{SOL} \cup S_s)}{h(S_{SOL} \cup S_s) - h(S_s)} h(S_{SOL}) &\geq \frac{gain(S_{SOL} \cup S_s)}{h(S_{SOL} \cup S_s) - h(S_s)} (h(S_{SOL} \cup S_s) - h(S_s)) \\ &= gain(S_{SOL} \cup S_s). \end{aligned} \tag{4.3}$$

Combining the result of Lemma 11 and (4.2), (4.3), we have

$$\begin{aligned} &\max \left\{ \max_{p \in P_{SOL}} \left\{ \frac{gain(p)}{f_p'} \right\}, \frac{gain(S_{SOL} \cup S_s)}{h(S_{SOL} \cup S_s) - h(S_s)} \right\} (F_{SOL} + h(S_{SOL})) \\ &\geq \max_{p \in P_{SOL}} \left\{ \frac{gain(p)}{f_p'} \right\} F_{SOL} + \frac{gain(S_{SOL} \cup S_s)}{h(S_{SOL} \cup S_s) - h(S_s)} h(S_{SOL}) \\ &\geq \sum_{p \in P_{SOL}} gain(p) + gain(S_{SOL} \cup S_s) \\ &\geq C_s - (F_{SOL} + h_{SOL} + C_{SOL}). \end{aligned}$$

According to the definitions of r_{s+1} and r'_{s+1}, we have the following:

$$\begin{aligned}\max\{r_{s+1}, r'_{s+1}\} &\geq \max\left\{\max_{p\in P_{SOL}\setminus P_s}\left\{\frac{gain(p)}{f'_p}\right\}, \frac{gain((S_{SOL}\setminus S_s)\cup S_s)}{h((S_{SOL}\setminus S_s)\cup S_s)-h(S_s)}\right\}\\ &= \max\left\{\max_{p\in P_{SOL}}\left\{\frac{gain(p)}{f'_p}\right\}, \frac{gain(S_{SOL}\cup S_s)}{h(S_{SOL}\cup S_s)-h(S_s)}\right\}\\ &\geq \frac{C_s-(F_{SOL}+h_{SOL}+C_{SOL})}{F_{SOL}+h_{SOL}}.\end{aligned}$$

□

Lemma 15 *Assume that SOL is an arbitrary feasible solution, F_0, C_0, h_0 are the opening, connection and penalty cost of the initial feasible solution SOL_0, respectively. After greedy augmentation, the total cost of the resulted solution is not exceeding* $F_0+h_0+(F_{SOL}+h_{SOL})\max\{0, \ln(\frac{C_0-C_{SOL}}{F_{SOL}+h_{SOL}})\}+F_{SOL}+h_{SOL}+C_{SOL}$.

Proof For iteration s ($s \geq 0$), the current solution SOL_s has opening cost F_s, connection cost C_s and penalty cost h_s. When $C_0 \leq F_{SOL}+h_{SOL}+C_{SOL}$, the lemma is true. Lemma 14 indicates that there exists an integer $m \geq 1$ such that $C_m \leq F_{SOL}+h_{SOL}+C_{SOL}$ and $C_k > F_{SOL}+h_{SOL}+C_{SOL}$ for all $0 \leq s < m$. It suffices to bound the cost at iteration m and the same bound will still hold for the final solution.

Consider an arbitrary iteration s ($0 \leq s < m$). It follows from Lemma 14 and Algorithm 3 that

$$\frac{C_s+F_s+h_s-(C_{s+1}+F_{s+1}+h_{s+1})}{F_{s+1}+h_{s+1}-(F_s+h_s)} \geq \frac{C_s-F_{SOL}-h_{SOL}-C_{SOL}}{F_{SOL}+h_{SOL}}.$$

or equivalently,

$$F_{s+1}+h_{s+1}-F_s-h_s \leq (F_{SOL}+h_{SOL})\frac{C_s-C_{s+1}}{C_s-C_{SOL}}.$$

From the definition that in every iteration only one of F_s and h_s changes. We have

$$\begin{aligned}F_m+h_m+C_m &= F_0+h_0+\sum_{s=1}^{m}(F_s+h_s-F_{s-1}-h_{s-1})+C_m\\ &\leq F_0+h_0+(F_{SOL}+h_{SOL})\sum_{s=1}^{m}\frac{C_{s-1}-C_s}{C_{s-1}-C_{SOL}}+C_m.\end{aligned} \tag{4.4}$$

The derivation of the last expression of (4.4) for C_m is $1-\frac{F_{SOL}+C_{SOL}}{C_{m-1}-C_{SOL}}$, for $C_{m-1} > F_{SOL}+C_{SOL} \geq C_{SOL}$, so the right hand of inequality (4.4) increases monotonically about C_m. When $C_m = F_{SOL}+h_{SOL}+C_{SOL}$, (4.4) can arrive the maximal value. In the following discussion, we assume that $C_m = F_{SOL}+h_{SOL}+C_{SOL}$. Finally, we have

$$
\begin{aligned}
&F_m + h_m + C_m \\
&\leq F_0 + h_0 + (F_{SOL} + h_{SOL}) \sum_{s=1}^{m} \tfrac{C_{s-1} - C_s}{C_{s-1} - C_{SOL}} + C_m \\
&= F_0 + h_0 + (F_{SOL} + h_{SOL}) \sum_{s=1}^{m} (1 - \tfrac{C_s - C_{SOL}}{C_{s-1} - C_{SOL}}) + C_m \\
&\leq F_0 + h_0 + (F_{SOL} + h_{SOL}) \sum_{s=1}^{m} \ln(\tfrac{C_{s-1} - C_{SOL}}{C_s - C_{SOL}}) + C_m \\
&= F_0 + h_0 + (F_{SOL} + h_{SOL}) \ln(\tfrac{C_0 - C_{SOL}}{C_m - C_{SOL}}) + C_m \\
&= F_0 + h_0 + (F_{SOL} + h_{SOL}) \ln(\tfrac{C_0 - C_{SOL}}{F_{SOL} + P_{SOL}}) + F_{SOL} + h_{SOL} + C_{SOL}.
\end{aligned}
$$

□

Theorem 2 *The approximation ratio for Algorithm 3 is no more than 2.9444.*

Proof Let $F_{OPT}, C_{OPT}, h_{OPT}$ denote the opening, connection and penalty costs of the optimal solution to the original instance. Let the opening, connection and penalty costs of the solution output of Algorithm 2 be F, C and h. Applying the primal-dual algorithm to the modified instance, we get a solution SOL_0 with facility opening cost F', penalty cost h' and connection cost C', which corresponding to facility cost $F_0 = F'/\delta$, penalty cost $h_0 = h'/\delta$ and connection cost $C_0 = C'$, respectively. If the initial solution SOL_0, which is the output solution of Algorithm 1, is viewed as a feasible solution of the original instance.

From Theorem 1 and scale the opening and penalty costs by a factor δ,

$$
\begin{aligned}
3\delta(F_0 + h_0) + C_0 &= 3(F' + h') + C' \\
&\leq 6[\delta(F_{OPT} + h_{OPT}) + C_{OPT}].
\end{aligned}
$$

There are two possibilities.

Case 1. $C_0 \leq F_{OPT} + h_{OPT} + C_{OPT}$.

$$
\begin{aligned}
\check{F} + \check{h} + \check{C} &\leq F_0 + h_0 + C_0 \\
&= \frac{3\delta(F_0 + h_0) + C_0}{3\delta} + (1 - \frac{1}{3\delta})C_0 \\
&\leq (3 - \frac{1}{3\delta})(F_{OPT} + h_{OPT}) + (1 + \frac{5}{3\delta})C_{OPT}.
\end{aligned}
$$

Case 2. $C_0 > F_{OPT} + h_{OPT} + C_{OPT}$.

$$
C_0 \leq 6[\delta(F_{OPT} + h_{OPT}) + C_{OPT}] - 3\delta(F_0 + h_0),
$$

Lemma 15 implies that the cost after greedy augmentation is at most

$$
F_0 + h_0 + (F_{OPT} + h_{OPT}) \ln\left(\frac{6\delta(F_{OPT} + h_{OPT}) + 5C_{OPT} - 3\delta(F_0 + h_0)}{F_{OPT} + h_{OPT}}\right) + F_{OPT} + h_{OPT} + C_{OPT}.
$$

The derivation of $F_0 + h_0$ in the above indicates that when

$$F_0 + h_0 = F_{OPT} + h_{OPT} + \frac{5}{3\delta} C_{OPT},$$

The polynomial achieves its maximal value.

$$\check{F} + \check{h} + \check{C} \leq (2 + \ln(3\delta))(F_{OPT} + h_{OPT}) + (1 + \frac{5}{3\delta}) C_{OPT}.$$

From Case 1 and Case 2, we have

$$\begin{aligned} \check{F} + \check{h} + \check{C} &\leq \max\{3 - \frac{1}{3\delta}, 2 + \ln(3\delta)\}(F_{OPT} + h_{OPT}) + (1 + \frac{5}{3\delta}) C_{OPT} \\ &\leq (2 + \ln(3\delta))(F_{OPT} + h_{OPT}) + (1 + \frac{5}{3\delta}) C_{OPT} \\ &\leq \max\{2 + \ln(3\delta), 1 + \frac{5}{3\delta}\} OPT. \end{aligned}$$

When $\delta = 0.8571$, Algorithm 3 can achieve the best approximation ratio 2.9444. So the approximation factor for the Algorithm 3 is no more than 2.9444. □

5 Conclusion

We consider k-FLPSP for any constant k in this paper and give an improved approximation algorithm. In Algorithm 2, when we use greedy augmentation for path, the facility in a path may appear in many paths, we define a new opening cost for each facility to avoid repeating the computation of opening costs, every facility cost is just computed once. How to analyse the opening cost properly, it is a question also exist in k-FLPSP and all extensions of k-FLPSP. k-FLPSP is an extension of k-LFLP, we don't know whether our algorithm can be used in k-LFLP. The lower bound of k-LFLP is 1.61, there is a sharp gap between the current upper bound and lower bound.

Funding The fund was provided by National Nature Science Foundation of China.

Data Availability All data generated or analysed during this study are included in this published article [and its supplementary information files]

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

Byrka J, Rybicki B (2012) Improved LP-rounding approximation algorithm for k-level uncapacitated facility location. In: Automata, languages, and programming. Part I, Lecture Notes in Comput. Sci., vol 7391, Springer, Heidelberg, pp 157–169

Charikar M, Guha S (2005) Improved combinatorial algorithms for facility location problems. SIAM J Comput 34(4):803–824

Charikar M, Khuller S, Mount DM, Narasimhan G (2001) Algorithms for facility location problems with outliers (extended abstract). In: Proceedings of the twelfth annual ACM-SIAM symposium on discrete algorithms (Washington, DC, 2001), SIAM, Philadelphia, PA, pp 642–651

Chudak FA, Shmoys DB (2003) Improved approximation algorithms for the uncapacitated facility location problem. SIAM J Comput 33(1):1–25

Fujishige S (2005) Submodular functions and optimization. Ann Discrete Math, vol 58, 2nd edn. Elsevier B. V, Amsterdam

Hayrapetyan A, Swamy C, Tardos E (2005) Network design for information networks. In: Proceedings of the sixteenth annual ACM-SIAM symposium on discrete algorithms, ACM, New York, pp 933–942

Jain K, Vazirani VV (2001) Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. J ACM 48(2):274–296

Jain K, Mahdian M, Markakis E, Saberi A, Vazirani VV (2003) Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. J ACM 50(6):795–824

Krishnaswamy R, Sviridenko M (2012) Inapproximability of the multi-level uncapacitated facility location problem. In: Proceedings of the twenty-third annual ACM-SIAM symposium on discrete algorithms, ACM, New York, pp 718–734

Li G, Wang Z, Xu D (2012) An approximation algorithm for the k-level facility location problem with submodular penalties. J Ind Manag Optim 8(3):521–529

Li G, Xu D, Du D, Wu C (2015) Approximation algorithms for the multilevel facility location problem with linear/submodular penalties. Front Algorithm. Springer International Publishing, Cham, pp 162–169

Li S (2013) A 1.488 approximation algorithm for the uncapacitated facility location problem. Inform Comput 222:45–58

Li Y, Du D, Xiu N, Xu D (2013) A combinatorial 2.375-approximation algorithm for the facility location problem with submodular penalties. Theoret Comput Sci 476:109–117

Li Y, Du D, Xiu N, Xu D (2015) Improved approximation algorithms for the facility location problems with linear/submodular penalties. Algorithmica 73(2):460–482

Mahdian M, Ye Y, Zhang J (2006) Approximation algorithms for metric facility location problems. SIAM J Comput 36(2):411–432

Qiu X, Kern W (2016) On the factor revealing lp approach for facility location with penalties. https://doi.org/10.48550/arXiv.1602.00192

Shmoys DB, Tardos E, Aardal K (1997) Approximation algorithms for facility location problems (extended abstract). In: Proceedings of the twenty-ninth annual ACM symposium on theory of computing, association for computing machinery, New York, NY, USA, STOC '97, pp 265–274. https://doi.org/10.1145/258533.258600

Sviridenko M (2002) An improved approximation algorithm for the metric uncapacitated facility location problem. In: Integer programming and combinatorial optimization, Lecture Notes in Comput. Sci., vol 2337, Springer, Berlin, pp 240–257

Wang Y, Xu D, Du D, Wu C (2018) An approximation algorithm for k-facility location problem with linear penalties using local search scheme. J Comb Optim 36(1):264–279

Xu G, Xu J (2009) An improved approximation algorithm for uncapacitated facility location problem with penalties. J Comb Optim 17(4):424–436

Zhang P (2007) A new approximation algorithm for the k-facility location problem. Theoret Comput Sci 384(1):126–135

Zhang Z, Feng Q, Huang J, Wang J (2023) Improved approximation algorithms for solving the squared metric k-facility location problem. Theoret Comput Sci 942:107–122