TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 06/20 pp1694–1702
DOI: 10.26599/TST.2023.9010152
Volume 29, Number 6, December 2024

Approximation and Heuristic Algorithms for the Priority Facility Location Problem with Outliers

Hang Luo, Lu Han*, Tianping Shuai, and Fengmin Wang

Abstract: In this paper, we propose the Priority Facility Location Problem with Outliers (PFLPO), which is a generalization of both the Facility Location Problem with Outliers (FLPO) and Priority Facility Location Problem (PFLP). As our main contribution, we use the technique of primal-dual to provide a 3-approximation algorithm for the PFLPO. We also give two heuristic algorithms. One of them is a greedy-based algorithm and the other is a local search algorithm. Moreover, we compare the experimental results of all the proposed algorithms in order to illustrate their performance.

Key words: priority facility location; primal-dual; approximation algorithm

1 Introduction

The Uncapacitated Facility Location Problem (UFLP) had been extensively studied in the field of operations research[1]. In the UFLP, a facility location set and a client location set are given. Opening a facility at some facility location incurs a non-negative opening cost, and connecting a client from its location to some facility location incurs a connection cost. The aim is to open facilities at some facility locations, connect each client from its location to the location of some opened facilities, such that the total cost (i.e., the sum of opening and connection costs) is minimized. A great deal of approximation algorithms were proposed in order to solve the UFLP. A ρ-approximation algorithm of a minimization problem is an algorithm that, for any instance of the problem, could always output a feasible solution within a factor of ρ of the optimum in polynomial time. For a ρ-approximation algorithm, we call ρ its approximation ratio.

In general, we assume that the given connection costs are metric, which means they are non-negative, symmetric, and satisfy the triangle inequality. Based on the technique of Location Problem (LP) rounding, the first constant-factor approximation algorithm for the UFLP was designed by Shmoys et al.[2], which has an approximation ratio of 3.16. By combining the technique of LP-rounding and dual-fitting, Li[3] gave the currently best 1.488-approximation algorithm. Sviridenko[4], showed that unapproximable lower bound of the UFLP is 1.463 unless NP = P.

In addition to these three important results presented above for the UFLP, many other approximation algorithms were also given using the techniques of LP-rounding[4–6], primal-dual[7], dual-fitting[8–10], and local search[11, 12]. Among all the approximation algorithms used to solve the UFLP, Jain and Vazirani[7] proposed an elegant primal-dual 3-approximation algorithm. It is worth mentioning that their primal-dual algorithm is very versatile and can be adapted to solve many generalizations of the UFLP.

A limitation of the model of the UFLP is that some clients with relatively large connection cost could have a huge impact on the total cost. To overcome this limitation, Charikar et al.[13] proposed two

• Hang Luo is with International School, Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: l.hangg@icloud.com.
• Lu Han and Tianping Shuai are with School of Science, Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: hl@bupt.edu.cn; tpshuai@bupt.edu.cn.
• Fengmin Wang is with Beijing Jinghang Research Institute of Computing and Communication, Beijing 100074, China. E-mail: casic_wfm@163.com.
* To whom correspondence should be addressed.
Manuscript received: 2023-07-06; revised: 2023-10-04; accepted: 2023-12-04

generalizations of the UFLP, which are the Facility Location Problem with Penalties (FLPP) and Robust Facility Location Problem (RFLP). The RFLP is sometimes called the Facility Location Problem with Outliers (FLPO). Compared with the UFLP, in the FLPP, each client has a penalty cost, and the aim is to open facilities at some facility locations, connect some clients from their locations to the locations of the opened facilities, and pay the penalty costs of all the remaining clients, such that the total cost (i.e., the sum of opening, connection and penalty costs) is minimized. Compared with the UFLP, in the FLPO, a non-negative integer q is given, the aim is to open facilities at some facility locations, select at most q clients as outliers (i.e., the clients that do not need to be connected), connect all the remaining clients from their locations to the locations of the opened facilities, such that the total cost (i.e., the sum of opening and connection costs) is minimized. Based on the technique of primal-dual, Charikar et al.[13] introduced their 3-approximation algorithms for both the FLPP and FLPO.

The Priority Facility Location Problem (PFLP) is also a generalization of the UFLP. Compared with the UFLP, in the PFLP, we are given a level set $\mathcal{L} = \{1, 2, \ldots, L\}$. Each client has a level-of-service requirement belonging to $\mathcal{L}$, and each facility can be opened at any facility location and at any level in $\mathcal{L}$. The opening cost of a facility is associated with both its location and opening level. Assume that the cost of opening a facility at any location is non-decreasing as its opening level increases. If the level-of-service requirement of a client is no more than the opening level of some opened facility, it can be connected to the opened facility and incurs a connection cost. We assume that the connection cost of each facility-client pair is only related to their locations and not related to their corresponding levels. The aim is also to minimize the total cost (i.e., the sum of opening and connection costs). Ravi and Sinha[14] proposed the PFLP along with an LP-rounding 7-approximation algorithm. Mahdian[15] provided a primal-dual 3-approximation algorithm. Combining the technique of primal-dual and greedy augmentation procedure, Li et al.[16] presented the currently best 1.8526-approximation algorithm.

The PFLP also has the limitation that some clients may have excessive influence on its objective value. The only result to overcome the limitation was provided by Wang et al.[17], who proposed the Priority Facility Location Problem with Penalties (PFLPP) and gave two constant-factor approximation algorithms. On the other hand, the generalization of the PFLP considering outliers has not been studied yet.

In this paper, we propose the Priority Facility Location Problem with Outliers (PFLPO), which is a generalization of both the FLPO and PFLP. Compared with the PFLP, in the PFLPO, a non-negative integer q is given, and we select at most q clients as outliers. The outliers are allowed not to be connected. As our main contribution, we combine the works of Charikar et al.[13] on the FLPO and Mahdian[15] on the PFLP to design a constant-factor approximation algorithm for the PFLPO. The approximation ratio of our algorithm is 3. We also provide two heuristic algorithms. One is a greedy-based algorithm and the other is a local search algorithm. The performance of all the proposed algorithms are compared on synthetic data sets.

The remainder structure of our paper is as follows. Section 2 gives the formal description of the PFLPO along with its integer program, linear program relaxation, and dual program. Sections 3 and 4 provide the approximation algorithm and the heuristic algorithms, respectively. Section 5 presents our experimental results and some conclusions.

2 Preliminary

In a PFLPO instance $\mathcal{I}_{\mathrm{PFO}}$, we are given a facility location set $\mathcal{F}$, a client location set C, and a level set $\mathcal{L} = \{1, 2, \ldots, L\}$. Assume that $|\mathcal{F}| = m$ and $|C| = n$. A non-negative integer $q < n$ is also given. Each client, in addition to its location, is also associated with a level-of-service requirement which belongs to $\mathcal{L}$. For simplicity, we use a binary (j, l) to denote a client, where $j \in C$ and $l \in \mathcal{L}$ are its location and level-of-service requirement, respectively. Let $\mathfrak{C}$ be the set of all client binaries. Each facility, in addition to its location, can be opened at any level in $\mathcal{L}$. For simplicity, (i, s) denotes a facility, where $i \in \mathcal{F}$ and $s \in \mathcal{L}$ are its location and opening level, respectively. Let $\mathfrak{F}$ be the set of all facility binaries. Opening a facility located at $i \in \mathcal{F}$ at some level of $s \in \mathcal{L}$ (i.e., opening facility (i, s)) incurs a non-negative opening cost of f_{is}. We assume that the cost of opening a facility at any location is non-decreasing as the opening level increases. If the level-of-service requirement l of a client (j, l) is no more than the opening level of some opened facility (i, s), it can be connected to the opened facility and incurs a connection cost of c_{ij}. Note that each connection cost is only related to the locations of the corresponding facility and client. Assume that the connection costs are non-negative and symmetric, and satisfy the triangle inequality. The objective is to open some facilities in $\mathfrak{F}$, select at most q clients as outliers (i.e., the clients that do not need to be connected), connect all the remaining clients, such that the total cost (i.e., the sum of opening and connection costs) is minimized.

In order to provide the integer program of the given PFLPO instance $\mathcal{I}_{\text{PFO}}$, we introduce three types of 0−1 primal variables $\{x_{is,jl}\}_{(i,s)\in\mathfrak{F},(j,l)\in\mathfrak{C}}$, $\{y_{is}\}_{(i,s)\in\mathfrak{F}}$, and $\{z_{jl}\}_{(j,l)\in\mathfrak{C}}$. The variable $x_{is,jl}$ indicates whether client (j,l) is connected to facility (i,s). The variable y_{is} indicates whether facility (i,s) is opened. The variable z_{jl} indicates whether client (j,l) is being selected as an outlier. The integer program of the given PFLPO instance $\mathcal{I}_{\text{PFO}}$ is as follows:

$$
\begin{aligned}
\min\ & \sum_{(i,s)\in\mathfrak{F}} f_{is}y_{is} + \sum_{(i,s)\in\mathfrak{F}:\, l\leqslant s}\ \sum_{(j,l)\in\mathfrak{C}} c_{ij}x_{is,jl},\\
\text{s. t,}\ & \sum_{(i,s)\in\mathfrak{F}:\, l\leqslant s} x_{is,jl} + z_{jl} \geqslant 1, \forall\,(j,l)\in\mathfrak{C},\\
& x_{is,jl} \leqslant y_{is},\ \ \forall\,(j,l)\in\mathfrak{C},\ (i,s)\in\mathfrak{F},\\
& \sum_{(j,l)\in\mathfrak{C}} z_{jl} \leqslant q,\\
& x_{is,jl} \in \{0,1\},\ \ \forall\,(j,l)\in\mathfrak{C}, (i,s)\in\mathfrak{F},\\
& y_{is} \in \{0,1\},\ \ \forall\,(i,s)\in\mathfrak{F},\\
& z_{jl} \in \{0,1\},\ \ \forall (j,l)\in\mathfrak{C}.
\end{aligned}
$$

The objective function represents the sum of opening and connection costs. The first constraint ensures that each client $(j,l)\in\mathfrak{C}$ is either be connected to some facility or selected as an outlier. The second constraint ensures that if there exists a client $(j,l)\in\mathfrak{C}$ connected to some facility $(i,s)\in\mathfrak{F}$, then the facility (i,s) must be opened. The third constraint ensures that the number of selected outliers is at most q.

If we replace the constraints of $x_{is,jl}\in\{0,1\}$, $y_{is}\in\{0,1\}$, and $z_{jl}\in\{0,1\}$ in the above integer program with the constraints of $x_{is,jl}\geqslant 0$, $y_{is}\geqslant 0$, and $z_{jl}\geqslant 0$, resprctively, the following linear program relaxation can be obtained:

$$
\begin{aligned}
\min\ & \sum_{(i,s)\in\mathfrak{F}} f_{is}y_{is} + \sum_{(i,s)\in\mathfrak{F}:\, l\leqslant s}\ \sum_{(j,l)\in\mathfrak{C}} c_{ij}x_{is,jl},\\
\text{s. t.,}\ & \sum_{(i,s)\in\mathfrak{F}:\, l\leqslant s} x_{is,jl} + z_{jl} \geqslant 1, \forall\,(j,l)\in\mathfrak{C},\\
& x_{is,jl} \leqslant y_{is},\quad \forall\,(j,l)\in\mathfrak{C}, (i,s)\in\mathfrak{F},\\
& \sum_{(j,l)\in\mathfrak{C}} z_{jl} \leqslant q,\\
& x_{is,jl} \geqslant 0,\quad \forall\,(j,l)\in\mathfrak{C},\ (i,s)\in\mathfrak{F},\\
& y_{is} \geqslant 0,\quad \forall\,(i,s)\in\mathfrak{F},\\
& z_{jl} \geqslant 0,\quad \forall\,(j,l)\in\mathfrak{C}.
\end{aligned}
$$

Note that the integrality gap of the above linear program relaxation is unbounded. That means, based on this relaxation, it is impossible to design an approximation algorithm with a bounded approximation ratio.

By introducing three types of dual variables $\{\alpha_{jl}\}_{(j,l)\in\mathfrak{C}}$, $\{\beta_{is,jl}\}_{(i,s)\in\mathfrak{F},(j,l)\in\mathfrak{C}}$, and γ, we obtain the following dual program of the given PFLPO instance $\mathcal{I}_{\text{PFO}}$:

$$
\begin{aligned}
\max\ & \sum_{(j,l)\in\mathfrak{C}} \alpha_{jl} - \gamma q,\\
\text{s. t,}\ & \alpha_{jl} - \beta_{is,jl} \leqslant c_{ij},\ \ \forall\,(i,s)\in\mathfrak{F},\ (j,l)\in\mathfrak{C}, l\leqslant s,\\
& \sum_{(j,l)\in\mathfrak{C}} \beta_{is,jl} \leqslant f_{is},\ \ \forall\,(i,s)\in\mathfrak{F},\\
& \alpha_{jl} \leqslant \gamma,\quad \forall\,(j,l)\in\mathfrak{C},\\
& \alpha_{jl} \geqslant 0,\quad \forall\,(j,l)\in\mathfrak{C},\\
& \beta_{is,jl} \geqslant 0,\quad \forall\,(i,s)\in\mathfrak{F},\ (j,l)\in\mathfrak{C},\\
& \gamma \geqslant 0.
\end{aligned}
$$

We can view each dual variable α_{jl} as the budget of client (j,l), and view each dual variable $\beta_{is,jl}$ as the contribution of client (j,l) to facility (i,s).

3 Approximation Algorithm

In this section, we propose a constant-factor approximation algorithm for the PFLPO.

3.1 Description of the algorithm

The proposed approximation algorithm is based on the technique of primal-dual, and it has three essential construction phases. The first pre-construction phase constructs a modified PFLPO instance $\mathcal{I}_{\text{PFO}}^{\text{exp}}$ based on the given PFLPO instance $\mathcal{I}_{\text{PFO}}$. The second dual construction phase constructs a dual feasible solution of the dual program of instance $\mathcal{I}_{\text{PFO}}^{\text{exp}}$. The third primal construction phase uses the obtained dual solution to construct a primal feasible solution of the integer program of instance $\mathcal{I}_{\text{PFO}}$.

Phase 1: Pre-construction phase

Phase 1 constructs a modified instance $\mathcal{I}_{\text{PFO}}^{\text{exp}}$ to overcome the obstacle that the integrality gap of the natural linear program relaxation of the given PFLPO instance $\mathcal{I}_{\text{PFO}}$ is unbounded. In Phase 1, we first guess the facility $(i_{\text{e}},s_{\text{e}})\in\mathfrak{F}$, which has the most expensive opening cost in an optimal solution of instance $\mathcal{I}_{\text{PFO}}$.

Then, we modify the opening cost of the facility (i_e, s_e) and all the facilities with more expensive opening costs to obtain the modified instance. Algorithm 1 is the formal description of Phase 1.

Phase 2: Dual construction phase

Phase 2 constructs a dual feasible solution by uniformly raise dual variables $\{\alpha_{jl}\}_{(j,l)\in\mathfrak{C}}$ from zero. Algorithm 2 is the formal description of Phase 2. In this algorithm, we use $\mathfrak{F}_{\text{tem}}$, $\mathfrak{C}_{\text{tem}}$, and $\mathfrak{O}_{\text{tem}}$ to denote the set of temporarily opened facilities, connected clients, and selected outliers, respectively. For each facility $(i,s)\in\mathfrak{F}$, denote by t_{is} the time it is temporarily opened. For each client $(j,l)\in\mathfrak{C}$, denote by $w(j,l)$ the facility that causes the dual variable α_{jl} to stop raising, and we call $w(j,l)$ the connecting witness of (j,l). The output of the dual variables $\{\alpha_{jl}\}_{(j,l)\in\mathfrak{C}}$, $\{\beta_{is,jl}\}_{(i,s)\in\mathfrak{F},(j,l)\in\mathfrak{C}}$ and γ form a dual feasible solution.

Note that the dual ascent process in Step 2 of Algorithm 2 does not violate any constraints of the dual program of $\mathcal{I}_{\text{PFO}}^{\text{exp}}$. Thus we have the following lemma.

Lemma 1 Algorithm 2 outputs a dual feasible solution of the dual program of the modified PFLPO instance $\mathcal{I}_{\text{PFO}}^{\text{exp}}$.

Phase 3: Primal construction phase

In order to construct a primal feasible solution with constant approximation ratio, Phase 3 requires that any two finally opened facilities cannot be contributed by the same client. Algorithm 3 is the formal description of Phase 3. In this algorithm, we use $\mathfrak{F}_{\text{fin}}$, $\mathfrak{C}_{\text{fin}}$, and $\mathfrak{O}_{\text{fin}}$ to denote the set of finally opened facilities, connected clients, and selected outliers, respectively. For each facility $(i,s)\in\mathfrak{F}_{\text{fin}}$, recall that t_{is} is the time (i,s) is temporarily opened. For each client $(j,l)\in\mathfrak{C}_{\text{fin}}$, denote by $\sigma(j,l)$ the facility to which it is connected. The output of the primal variables $\{x_{is,jl}\}_{(i,s)\in\mathfrak{F},(j,l)\in\mathfrak{C}}$, $\{y_{is}\}_{(i,s)\in\mathfrak{F}}$, and $\{z_{jl}\}_{(j,l)\in\mathfrak{C}}$ form a primal feasible solution.

In the following lemma, we show the feasibility of the obtained primal solution.

Lemma 2 Algorithm 3 outputs a primal feasible solution of the integer program of the given PFLPO instance $\mathcal{I}_{\text{PFO}}$.

Proof If Step 4 of Algorithm 3 can be successfully performed, the obtained solution must be feasible. Thus, we aim to prove that for each client $(j,l)\in\mathfrak{C}_{\text{fin}}$,

---

Algorithm 1 Pre-construction phase

Input: A given PFLPO instance $\mathcal{I}_{\text{PFO}}$.

Output: A modified PFLPO instance $\mathcal{I}_{\text{PFO}}^{\text{exp}}$.

Step 1: Guess the most expensive opened facility (i_e, s_e) in an optimal solution of the given PFLPO instance, $\mathcal{I}_{\text{PFO}} = (\mathfrak{F}, \mathfrak{C}, \{f_{is}\}_{(i,s)\in\mathfrak{F}}, \{c_{ij}\}_{i\in\mathcal{F}, j\in C})$, where $\mathfrak{F} := \{(i,s) : i\in\mathcal{F}, s\in\mathcal{L}\}$, and $\mathfrak{C} := \{(j,l) : j\in C, l\in\mathcal{L}\}$.

Step 2: For each facility $(i,s)\in\mathfrak{F}$, we define

$$f'_{is} := \begin{cases} 0, \text{ if facility } (i,s) = (i_e, s_e); \\ \infty, \text{ if facility } (i,s) \text{ satisfies } f_{is} > f_{i_e s_e}; \\ f_{is}, \text{ otherwise.} \end{cases}$$

Construct the modified PFLPO as $\mathcal{I}_{\text{PFO}}^{\text{exp}} = (\mathfrak{F}, \mathfrak{D}, \{f'_{is}\}_{(i,s)\in\mathfrak{F}}, \{c_{ij}\}_{i\in\mathcal{F}, j\in C})$.

Step 3: Output modified PFLPO instance $\mathcal{I}_{\text{PFO}}^{\text{exp}}$.

Algorithm 2 Dual construction phase

Input: Modified PFLPO instance $\mathcal{I}_{\text{PFO}}^{\text{exp}}$.

Output: A dual feasible solution of the dual program of the modified PFLPO instance $\mathcal{I}_{\text{PFO}}^{\text{exp}}$ and some useful sets.

Step 1: Initialization.

For any client $(j,l)\in\mathfrak{C}$, set $\alpha_{jl} := 0$. For any facility $(i,s)\in\mathfrak{F}$, client $(j,l)\in\mathfrak{C}$, set $\beta_{is,jl} := 0$, $\gamma := 0$. $\mathfrak{F}_{\text{tem}} := \varnothing$, $\mathfrak{C}_{\text{tem}} := \varnothing$, $\mathfrak{O}_{\text{tem}} := \mathfrak{C}$, and $t := 0$. For any facility $(i,s)\in\mathfrak{F}$, set $t_{is} := \infty$. Construct a dummy facility (i_d, s_d). For any client $(j,l)\in\mathfrak{C}$, set $w(j,l) := (i_d, s_d)$.

Step 2: Dual ascent process.

while $|\mathfrak{O}_{\text{tem}}| > q$ **do**

Raise t as well as each dual variable α_{jl} satisfying $(j,l)\in\mathfrak{O}_{\text{tem}}$ uniformly until some of the following events happens. If several events happen at the same time, we arbitrarily break ties.

Event 1. There exist some facility $(i,s)\in\mathfrak{F}\setminus\mathfrak{F}_{\text{tem}}$ and client $(j,l)\in\mathfrak{O}_{\text{tem}}$, such that

$l\leqslant s$ and $\alpha_{jl} = c_{ij}$.

Event 2. There exist some facility $(i,s)\in\mathfrak{F}_{\text{tem}}$ and client $(j,l)\in\mathfrak{O}_{\text{tem}}$, such that

$l\leqslant s$ and $\alpha_{jl} = c_{ij}$.

Event 3. There exists some facility $(i,s)\in\mathfrak{F}\setminus\mathfrak{F}_{\text{tem}}$, such that

$$\sum_{(j,l)\in\mathfrak{C}} \beta_{is,jl} = f'_{is}.$$

If Event 1 happens, update $\beta_{is,jl} := \alpha_{jl} - c_{ij}$. Raise $\beta_{is,jl}$ uniformly as α_{jl} increases. If Event 2 happens, stop raise α_{jl}. Update $\mathfrak{C}_{\text{tem}} := \mathfrak{C}_{\text{tem}}\cup\{(j,l)\}$, and $\mathfrak{O}_{\text{tem}} := \mathfrak{O}_{\text{tem}}\setminus\{(j,l)\}$. Update $w(j,l) := (i,s)$. If Event 3 happens, stop raise α_{jl} for each client $(j,l)\in\mathfrak{O}_{\text{tem}}$ satisfying $\beta_{is,jl} > 0$. Define $\mathfrak{C}_{(i,s)} := \{(j,l)\in\mathfrak{O}_{\text{tem}} : \beta_{is,jl} > 0\}$. Update $\mathfrak{F}_{\text{tem}} := \mathfrak{F}_{\text{tem}}\cup\{(i,s)\}$, $\mathfrak{C}_{\text{tem}} := \mathfrak{C}_{\text{tem}}\cup\mathfrak{C}_{(i,s)}$, and $\mathfrak{O}_{\text{tem}} := \mathfrak{O}_{\text{tem}}\setminus\mathfrak{C}_{(i,s)}$. Update $t_{is} := t$ and $w(j,l) := (i,s)$ for each client $(j,l)\in\mathfrak{C}_{(i,s)}$.

Stop raise t as well as each dual variable α_{jl} satisfying $(j,l)\in\mathfrak{O}_{\text{tem}}$, and update $\gamma := t$.

Step 3: Output dual variables $\{\alpha_{jl}\}_{(j,l)\in\mathfrak{C}}$, $\{\beta_{is,jl}\}_{(i,s)\in\mathfrak{F},(j,l)\in\mathfrak{C}}$ and γ. Set $\mathfrak{F}_{\text{tem}}$, $\mathfrak{C}_{\text{tem}}$, and $\mathfrak{O}_{\text{tem}}$.

Algorithm 3 Primal construction phase

Input: A dual feasible solution of the dual program of $\mathcal{I}_{\rm PFO}^{\rm exp}$ and sets $\mathfrak{F}_{\rm tem}$, $\mathfrak{C}_{\rm tem}$, and $\mathfrak{O}_{\rm tem}$ obtained from Algorithm 2.

Output: A primal feasible solution of the integer program of the given PFLPO instance $\mathcal{I}_{\rm PFO}$.

Step 1: Initialization.

For any facility $(i,s) \in \mathfrak{F}$, client $(j, l) \in \mathfrak{C}$, set $x_{is,\, jl} := 0$. For any facility $(i,s) \in \mathfrak{F}$, set $y_{is} := 0$. For any client $(j, l) \in \mathfrak{C}$, set $z_{jl} := 0$. Denote by $(i_{\rm la}, s_{\rm la})$ the last facility to be added to $\mathfrak{F}_{\rm tem}$ (i.e., the last temporarily opened facility).

Step 2: Determine outliers.

If $|\mathfrak{O}_{\rm tem}| = q$, set $\mathfrak{O}_{\rm fin} := \mathfrak{O}_{\rm tem}$. For each client $(j, l) \in \mathfrak{O}_{\rm fin}$, update $z_{jl} := 1$. If $|\mathfrak{O}_{\rm tem}| < q$, arbitrarily select $q - |\mathfrak{O}_{\rm tem}|$ clients in $\mathfrak{C}_{(i_{\rm la},\, s_{\rm la})}$, and denote by $\mathfrak{O}_{\rm la}$ these selected clients. Set $\mathfrak{C}_{\rm fin} := \mathfrak{C}_{\rm tem} \setminus \mathfrak{O}_{\rm la}$, and $\mathfrak{O}_{\rm fin} := \mathfrak{O}_{\rm tem} \cup \mathfrak{O}_{\rm la}$. For each client $(j, l) \in \mathfrak{O}_{\rm fin}$, update $z_{jl} := 1$.

Step 3: Open facilities.

Set $\mathfrak{F}_{\rm fin} := \varnothing$. For each facility $(i,s) \in \mathfrak{F}$, define

$\mathfrak{N}_{(i,\, s)} := \{(j, l) \in \mathfrak{C} : \beta_{is,\, jl} > 0\}$.

According to the opening levels of the facilities in $\mathfrak{F}_{\rm tem}$, order the facilities from the one with largest level to the one with smallest level. Set $k := 1$.

while $k \leqslant |\mathfrak{F}_{\rm tem}|$ **do**

For the k-th facility (i,s) in $\mathfrak{F}_{\rm tem}$, check whether there exists some facility (i',s') in $\mathfrak{F}_{\rm fin}$, such that

$\mathfrak{N}_{(i,\, s)} \cap \mathfrak{N}_{(i',\, s')} \neq \varnothing$.

If there exists such facility, update $k := k+1$. Otherwise, update $\mathfrak{F}_{\rm fin} := \mathfrak{F}_{\rm fin} \cup \{(i,s)\}$, and $y_{is} := 1$, and $k := k+1$.

Step 4: Connect clients.

For each client $(j, l) \in \mathfrak{C}_{\rm fin}$, find facility

$(i,s) := \arg \min\limits_{(i',\, s') \in \mathfrak{F}_{\rm fin}: l \leqslant s'} c_{i' j}$,

set $\sigma(j,l) := (i,s)$, and update $x_{is,jl} := 1$.

Step 5: Output primal variables $\{x_{is,\, jl}\}_{(i,\, s) \in \mathfrak{F},\, (j,\, l) \in \mathfrak{C}}$, $\{y_{is}\}_{(i,\, s) \in \mathfrak{F}}$, and $\{z_{jl}\}_{(j,\, l) \in \mathfrak{C}}$.

there must exist some facility $(i,s) \in \mathfrak{F}_{\rm fin}$, such that $l \leqslant s$. The proof can be split into two cases.

- **Case 1.** For client (j, l), its connecting witness $w(j, l) \in \mathfrak{F}_{\rm fin}$.

In this case, we use (i,s) to represent facility $w(j, l)$. From Step 2 of Algorithm 2, it can be seen that for client (j, l) and its connecting witness (i,s), the requirement of $l \leqslant s$ always holds.

- **Case 2.** For client (j, l), its connecting witness $w(j, l) \notin \mathfrak{F}_{\rm fin}$.

In this case, we use (i,s) to represent facility $w(j, l)$. Since $(i,s) \notin \mathfrak{F}_{\rm fin}$, from Step 3 of Algorithm 3, there must exists some facility $(i',s') \in \mathfrak{F}_{\rm fin}$ such that $\mathfrak{N}_{(i,\, s)} \cap \mathfrak{N}_{(i',\, s')} \neq \varnothing$ and $s \leqslant s'$. Since for client (j, l) and its connecting witness (i,s), the requirement of $l \leqslant s$ always holds, we have $l \leqslant s'$.

Combining Cases 1 and 2 completes the proof of this lemma. ■

3.2 Analysis of the algorithm

Denote by OPT and $\mathrm{OPT}^{\rm exp}$ the total cost of optimal solutions of instances $\mathcal{I}_{\rm PFO}$ and $\mathcal{I}_{\rm PFO}^{\rm exp}$, respectively. Recall that $(i_{\rm e}, s_{\rm e})$ is the facility which has the most expensive opening cost in an optimal solution of instance $\mathcal{I}_{\rm PFO}$. Define $\mathrm{OPT}' := \mathrm{OPT} - f_{i_{\rm e} s_{\rm e}}$.

We divide the set $\mathfrak{C}_{\rm fin}$ into two sets $\mathfrak{C}_{\rm fin}^1$ and $\mathfrak{C}_{\rm fin}^2$, where $\mathfrak{C}_{\rm fin}^1$ includes each client $(j, l) \in \mathfrak{C}_{\rm fin}$ that has a facility $(i,s) \in \mathfrak{F}_{\rm fin}$ satisfying $\beta_{is,\, jl} > 0$, and $\mathfrak{C}_{\rm fin}^2$ includes each client $(j, l) \in \mathfrak{C}_{\rm fin}$ that has no facility $(i,s) \in \mathfrak{F}_{\rm fin}$ satisfying $\beta_{is,\, jl} > 0$. For each client $(j, l) \in \mathfrak{C}_{\rm fin}^1$, define $\alpha_{jl}^{\rm F} := \beta_{is,\, jl}$ and $\alpha_{jl}^{\rm C} := c_{ij}$, where (i,s) is the only facility in $\mathfrak{F}_{\rm fin}$ which satisfies $\beta_{is,\, jl} > 0$. From Step 2 of Algorithm 2 for each client $(j, l) \in \mathfrak{C}_{\rm fin}^1$, we have

$$\alpha_{jl} = \beta_{is,\, jl} + c_{ij} = \alpha_{jl}^{\rm F} + \alpha_{jl}^{\rm C}.$$

For each client $(j, l) \in \mathfrak{C}_{\rm fin}^2$, define $\alpha_{jl}^{\rm F} := 0$ and $\alpha_{jl}^{\rm C} := \alpha_{jl}$, and we have

$$\alpha_{jl} = 0 + \alpha_{jl} = \alpha_{jl}^{\rm F} + \alpha_{jl}^{\rm C}.$$

Denote by FC the total opening cost of the obtained primal feasible solution, i.e.,

$$\mathrm{FC} = \sum_{(i,\, s) \in \mathfrak{F}} f_{is} y_{is}.$$

Lemma 3 Algorithm 3 outputs a primal feasible solution of total opening cost of FC, such that

$$\mathrm{FC} \leqslant \sum_{(j,\, l) \in \mathfrak{C}_{\rm fin}} \alpha_{jl}^{\rm F} + 2 f_{i_{\rm e} s_{\rm e}}.$$

Proof Since $y_{is} = 1$, if any only if $(i,s) \in \mathfrak{F}_{\rm fin}$, we have that

$$\mathrm{FC} = \sum_{(i,\, s) \in \mathfrak{F}} f_{is} y_{is} = \sum_{(i,\, s) \in \mathfrak{F}_{\rm fin}} f_{is} \tag{1}$$

From Step 2 of Algorithm 1 and the fact $f_{i_{\rm la} s_{\rm la}} \leqslant f_{i_{\rm e} s_{\rm e}}$, we have

$$\sum_{(i,\, s) \in \mathfrak{F}_{\rm fin}} f_{is} \leqslant \sum_{(i,\, s) \in \mathfrak{F}_{\rm fin} \setminus \{(f i_{\rm e},\, s_{\rm e}),\, (i_{\rm la},\, s_{\rm la})\}} f_{is} + f_{i_{\rm e} s_{\rm e}} + f_{i_{\rm la} s_{\rm la}} \leqslant \sum_{(i,\, s) \in \mathfrak{F}_{\rm fin} \setminus \{(i_{\rm e},\, s_{\rm e}),\, (i_{\rm la},\, s_{\rm la})\}} f'_{is} + 2 f_{i_{\rm e} s_{\rm e}} \tag{2}$$

Step 3 of Algorithm 3 guarantees that for each client

$(j,l) \in \mathfrak{F}$, there exists at most one facility $(i,s) \in \mathfrak{F}_{\text{fin}}$, such that $\beta_{is,jl} > 0$. Therefore,

$$\sum_{(i,s)\in\mathfrak{F}_{\text{fin}}\setminus\{(i_{\text{e}},s_{\text{e}}),(i_{\text{la}},s_{\text{la}})\}} f'_{is} \leqslant \sum_{(i,s)\in\mathfrak{F}_{\text{fin}}\setminus\{(i_{\text{e}},s_{\text{e}}),(i_{\text{la}},s_{\text{la}})\}} \sum_{(j,l)\in\mathfrak{N}_{(i,s)}} \beta_{is,jl} = \sum_{(i,s)\in\mathfrak{F}_{\text{fin}}\setminus\{(i_{\text{e}},s_{\text{e}}),(i_{\text{la}},s_{\text{la}})\}} \sum_{(j,l)\in\mathfrak{N}_{(i,s)}} \alpha^{\text{F}}_{jl} \leqslant \sum_{(j,l)\in\mathfrak{C}^1_{\text{fin}}} \alpha^{\text{F}}_{jl} = \sum_{(j,l)\in\mathfrak{C}_{\text{fin}}} \alpha^{\text{F}}_{jl} \tag{3}$$

Combining Formulas (1)–(3) completes the proof of Lemma 3. ■

Denote by CC the total connection cost of the obtained primal feasible solution, i.e.,

$$\text{CC} = \sum_{(i,s)\in\mathfrak{F}:\, l\leqslant s} \sum_{(j,l)\in\mathfrak{C}} c_{ij} x_{is,jl}.$$

Lemma 4 Algorithm 3 outputs a primal feasible solution of total connection cost of CC, such that

$$\text{CC} \leqslant 3 \sum_{(j,l)\in\mathfrak{C}_{\text{fin}}} \alpha^{\text{C}}_{jl}.$$

Proof Note that $x_{is,jl} = 1$ only if $(j,l) \in \mathfrak{C}_{\text{fin}}$. We consider the connection cost of each client $(j,l) \in \mathfrak{C}_{\text{fin}}$ according to two cases.

- **Case 1.** Client $(j,l) \in \mathfrak{C}^1_{\text{fin}}$.

For each client $(j,l) \in \mathfrak{C}^1_{\text{fin}}$, since there exists a facility $(i,s) \in \mathfrak{F}_{\text{fin}}$ satisfying $\beta_{is,jl} > 0$, its connection cost is no more than $c_{ij} = \alpha^{\text{C}}_{jl}$.

- **Case 2.** Client $(j,l) \in \mathfrak{C}^2_{\text{fin}}$.

For each client $(j,l) \in \mathfrak{C}^2_{\text{fin}}$, we use (i,s) to represent its connecting witness $w(j,l)$. If (i,s) belongs to $\mathfrak{F}_{\text{fin}}$, the connection cost of connecting client (j,l) is no more than $\alpha_{jl} = \alpha^{\text{C}}_{jl}$. If (i,s) does not belong to $\mathfrak{F}_{\text{fin}}$, there must exists some facility $(i',s') \in \mathfrak{F}_{\text{fin}}$ such that $\mathfrak{N}_{(i,s)} \cap \mathfrak{N}_{(i',s')} \neq \varnothing$ and $s \leqslant s'$. Let (j',l') be some client in $\mathfrak{N}_{(i,s)} \cap \mathfrak{N}_{(i',s')}$. Therefore, the connection cost of connecting client (j,l) is no more than

$$c_{i'j} \leqslant c_{i'j'} + c_{ij'} + c_{ij} \leqslant \alpha_{j'l'} + \alpha_{j'l'} + \alpha_{jl} \leqslant 3\alpha_{jl} = 3\alpha^{\text{C}}_{jl}.$$

The third inequality is due to that

$$\alpha_{j'l'} \leqslant \min\{t_{i's'}, t_{is}\} \leqslant t_{is} \leqslant \alpha_{jl}.$$

Combining Cases 1 and 2 completes the proof of Lemma 4. ■

Now, we are ready to give our main result.

Theorem 1 There is a 3-approximation algorithm for the PFLPO.

Proof From Lemmas 3 and 4, we have

$$\text{FC} + \text{CC} \leqslant \sum_{(j,l)\in\mathfrak{C}_{\text{fin}}} \alpha^{\text{F}}_{jl} + 2f_{i_{\text{e}}s_{\text{e}}} + 3\sum_{(j,l)\in\mathfrak{C}_{\text{fin}}} \alpha^{\text{C}}_{jl} \leqslant 3\sum_{(j,l)\in\mathfrak{C}_{\text{fin}}} \left(\alpha^{\text{F}}_{jl} + \alpha^{\text{C}}_{jl}\right) + 2f_{i_{\text{e}}s_{\text{e}}} \leqslant 3\sum_{(j,l)\in\mathfrak{C}_{\text{fin}}} \alpha_{jl} + 2f_{i_{\text{e}}s_{\text{e}}} \tag{4}$$

Since $\mathfrak{C} = \mathfrak{C}_{\text{fin}} \cup \mathfrak{O}_{\text{fin}}$, $\alpha_{jl} = \gamma$ for each client $(j,l) \in \mathfrak{O}_{\text{fin}}$, and $|\mathfrak{O}_{\text{fin}}| = q$, we have

$$\sum_{(j,l)\in\mathfrak{C}_{\text{fin}}} \alpha_{jl} = \sum_{(j,l)\in\mathfrak{C}} \alpha_{jl} - \sum_{(j,l)\in\mathfrak{O}_{\text{fin}}} \alpha_{jl} = \sum_{(j,l)\in\mathfrak{C}} \alpha_{jl} - \gamma q \leqslant \text{OPT}^{\text{exp}} \leqslant \text{OPT}' \tag{5}$$

The last inequality is due to that any optimal solution of instance $\mathcal{I}_{\text{PFO}}$ is a feasible solution of instance $\mathcal{I}^{\text{exp}}_{\text{PFO}}$.

Combining Formulas (4) and (5), we obtain

$$\text{FC} + \text{CC} \leqslant 3\text{OPT}' + 2f_{i_{\text{e}}s_{\text{e}}} \leqslant 3\text{OPT}.$$

We complete the proof of Theorem 1. ■

4 Heuristic Algorithms

In this section, based on existing basic techniques for solving facility location problems, we propose two heuristic algorithms, called greedy-based algorithm and local search algorithm, for the PFLPO.

4.1 Greedy-based algorithm

The proposed greedy-based algorithm starts with a feasible facility set, which selects some facility with the highest level that minimize the total cost. Then, we constantly update the currently facility set by finding the most valuable facility. A facility is valuable if adding it to the current facility set would reduces the total cost. Algorithm 4 is the formal description of the greedy-based algorithm. In this algorithm, we use $\mathfrak{F}_{\text{gb}}$, $\mathfrak{C}_{\text{gb}}$, and $\mathfrak{O}_{\text{gb}}$ to denote the set of opened facilities, connected clients, and selected outliers, respectively. For each client $(j,l) \in \mathfrak{C}_{\text{gb}}$, denote by $\sigma_{\text{gb}}(j,l)$ the facility to which it is connected. The output of the primal variables $\{x^{\text{gb}}_{is,jl}\}_{(i,s)\in\mathfrak{F},(j,l)\in\mathfrak{C}}$, $\{y^{\text{gb}}_{is}\}_{(i,s)\in\mathfrak{F}}$, and $\{z^{\text{gb}}_{jl}\}_{(j,l)\in\mathfrak{C}}$ form a primal feasible solution.

4.2 Local search algorithm

Same as the greedy-based algorithm, the proposed local search algorithm also starts with a feasible facility set. Then, we constantly update the current facility set if some local change could reduce its total cost.

Algorithm 4 Greedy-based algorithm

Input: A given PFLPO instance $\mathcal{I}_{\text{PFO}}$.
Output: A primal feasible solution of the integer program of the given PFLPO instance $\mathcal{I}_{\text{PFO}}$.
Step 1: Initialization.

For any facility $(i,s) \in \mathfrak{F}$, client $(j,l) \in \mathfrak{C}$, set $x^{\text{gb}}_{is,jl} := 0$. For any facility $(i,s) \in \mathfrak{F}$, set $y^{\text{gb}}_{is} := 0$. For any client $(j,l) \in \mathfrak{C}$, set $z^{\text{gb}}_{jl} := 0$. Set $\mathfrak{F}_{\text{gb}} := \varnothing$, $\mathfrak{C}_{\text{gb}} := \varnothing$, and $\mathfrak{O}_{\text{gb}} := \mathfrak{C}$. Construct a dummy facility $(i_{\text{d}}, s_{\text{d}})$, where $s_{\text{d}} = L$. Set $c_{i_{\text{d}}j} := \infty$ for each client $(j,l) \in \mathfrak{C}$. For each facility set $\mathfrak{F}' \subseteq \mathfrak{F}$, and client $(j,l) \in \mathfrak{C}$, define
$$(i_{(\mathfrak{F}',j)}, s_{(\mathfrak{F}',j)}) := \arg \min_{(i,s) \in \mathfrak{F}' \cup \{(i_{\text{d}}, s_{\text{d}})\}: l \leqslant s} c_{ij},$$
and $\tau(\mathfrak{F}', j)$ is the location of the facility $(i_{(\mathfrak{F}',j)}, s_{(\mathfrak{F}',j)})$, i.e., $\tau(\mathfrak{F}', j) := i_{(\mathfrak{F}',j)}$. For each facility set $\mathfrak{F}' \subseteq \mathfrak{F}$, define
$$T(\mathfrak{F}') := \sum_{(i,s) \in \mathfrak{F}'} f_{is} + \sum_{(j,l) \in \mathfrak{C}^{\min}_{\mathfrak{F}'}} c_{\tau(\mathfrak{F}',j)j},$$
where $\mathfrak{C}^{\min}_{\mathfrak{F}'}$ are the first $n-q$ clients in $\mathfrak{C}$ with the smallest connection cost of $c_{\tau(\mathfrak{F}',j)j}$.

Step 2: Open facilities.

Step 2.1: Find facility
$$(i_{\text{gb}}, s_{\text{gb}}) := \arg \min_{(i,s) \in \mathfrak{F}: s = L} T(\{(i,s)\}).$$
Update $\mathfrak{F}_{\text{gb}} := \mathfrak{F}_{\text{gb}} \cup \{(i_{\text{gb}}, s_{\text{gb}})\}$.

Step 2.2: For each facility $(i,s) \in \mathfrak{F} \setminus \mathfrak{F}_{\text{gb}}$, define
$$\text{Gain}(i,s) := T(\mathfrak{F}_{\text{gb}}) - T(\mathfrak{F}_{\text{gb}} \cup \{(i,s)\}).$$
Find the facility
$$(i_{\text{h}}, s_{\text{h}}) := \arg \max_{(i,s) \in \mathfrak{F} \setminus \mathfrak{F}_{\text{gb}}} \frac{\text{Gain}(i,s)}{f_{is}}.$$
If $\text{Gain}(i_{\text{h}}, s_{\text{h}}) > 0$, update $\mathfrak{F}_{\text{gb}} := \mathfrak{F}_{\text{gb}} \cup \{(i_{\text{h}}, s_{\text{h}})\}$, and repeat Step 2.2. Otherwise, update $y^{\text{gb}}_{is} := 1$ for each facility $(i,s) \in \mathfrak{F}_{\text{gb}}$, and go to Step 3.

Step 3: Determine outliers.

Update $\mathfrak{O}_{\text{gb}} := \mathfrak{C} \setminus \mathfrak{C}^{\min}_{\mathfrak{F}_{\text{gb}}}$, where $\mathfrak{C}^{\min}_{\mathfrak{F}_{\text{gb}}}$ are the first $n-q$ clients in $\mathfrak{C}$ with the smallest connection cost of $c_{\tau(\mathfrak{F}_{\text{gb}},j)j}$. For each client $(j,l) \in \mathfrak{O}_{\text{gb}}$, update $z^{\text{gb}}_{jl} := 1$.

Step 4: Connect clients.

Update $\mathfrak{C}_{\text{gb}} := \mathfrak{C}^{\min}_{\mathfrak{F}_{\text{gb}}}$. For each client $(j,l) \in \mathfrak{C}_{\text{gb}}$, find facility
$$(i,s) := \arg \min_{(i',s') \in \mathfrak{F}_{\text{gb}}: l \leqslant s'} c_{i'j},$$
set $\sigma_{\text{gb}}(j,l) := (i,s)$, and update $x^{\text{gb}}_{is,jl} := 1$.

Step 5: Output primal variables $\{x^{\text{gb}}_{is,jl}\}_{(i,s)\in\mathfrak{F},(j,l)\in\mathfrak{C}}$, $\{y^{\text{gb}}_{is}\}_{(i,s)\in\mathfrak{F}}$, and $\{z^{\text{gb}}_{jl}\}_{(j,l)\in\mathfrak{C}}$.

Algorithm 5 is the formal description of the local search algorithm. In this algorithm, we use $\mathfrak{F}_{\text{ls}}$, $\mathfrak{C}_{\text{ls}}$, and $\mathfrak{O}_{\text{ls}}$ to denote the set of opened facilities, connected clients, and selected outliers, respectively. For each client $(j,l) \in \mathfrak{C}_{\text{ls}}$, denote by $\sigma_{\text{ls}}(j,l)$ the facility to which it is connected. The output of the primal variables $\{x^{\text{ls}}_{is,jl}\}_{(i,s)\in\mathfrak{F},(j,l)\in\mathfrak{C}}$, $\{y^{\text{ls}}_{is}\}_{(i,s)\in\mathfrak{F}}$, and $\{z^{\text{ls}}_{jl}\}_{(j,l)\in\mathfrak{C}}$ form a primal feasible solution.

Algorithm 5 Local search algorithm

Input: A given PFLPO instance $\mathcal{I}_{\text{PFO}}$.
Output: A primal feasible solution of the integer program of the given PFLPO instance $\mathcal{I}_{\text{PFO}}$.
Step 1 Initialization.

For any facility $(i,s) \in \mathfrak{F}$, client $(j,l) \in \mathfrak{C}$, set $x^{\text{ls}}_{is,jl} := 0$. For any facility $(i,s) \in \mathfrak{F}$, set $y^{\text{gb}}_{ls} := 0$. For any client $(j,l) \in \mathfrak{C}$, set $z^{\text{ls}}_{jl} := 0$. Set $\mathfrak{F}_{\text{ls}} := \varnothing$, $\mathfrak{C}_{\text{ls}} := \varnothing$, and $\mathfrak{O}_{\text{ls}} := \mathfrak{C}$. Construct a dummy facility $(i_{\text{d}}, s_{\text{d}})$, where $s_{\text{d}} = L$. Set $c_{i_{\text{d}}j} := \infty$ for each client $(j,l) \in \mathfrak{C}$. For each facility set $\mathfrak{F}' \subseteq \mathfrak{F}$, and client $(j,l) \in \mathfrak{C}$, define
$$(i_{(\mathfrak{F}',j)}, s_{(\mathfrak{F}',j)}) := \arg \min_{(i,s) \in \mathfrak{F}' \cup \{(i_{\text{d}}, s_{\text{d}})\}: l \leqslant s} c_{ij},$$
and $\tau(\mathfrak{F}', j)$ is the location of the facility $(i_{(\mathfrak{F}',j)}, s_{(\mathfrak{F}',j)})$, i.e., $\tau(\mathfrak{F}', j) := i_{(\mathfrak{F}',j)}$. For each facility set $\mathfrak{F}' \subseteq \mathfrak{F}$, define
$$T(\mathfrak{F}') := \sum_{(i,s) \in \mathfrak{F}'} f_{is} + \sum_{(j,l) \in \mathfrak{C}^{\min}_{\mathfrak{F}'}} c_{\tau(\mathfrak{F}',j)j},$$
where $\mathfrak{C}^{\min}_{\mathfrak{F}'}$ are the first $n-q$ clients in $\mathfrak{C}$ with the smallest connection cost of $c_{\tau(\mathfrak{F}',j)j}$.

Step 2: Open facilities.

Step 2.1: Find facility
$$(i_{\text{ls}}, s_{\text{ls}}) := \arg \min_{(i,s) \in \mathfrak{F}: s = L} T(\{(i,s)\}).$$
Update $\mathfrak{F}_{\text{ls}} := \mathfrak{F}_{\text{ls}} \cup \{(i_{\text{ls}}, s_{\text{ls}})\}$.

Step 2.2: For facility set $\mathfrak{F}_{\text{ls}}$, define
$$\begin{aligned}&\mathcal{N}(\mathfrak{F}_{\text{ls}}) := \\ &\{\mathfrak{F}_{\text{ls}} \cup \{(i,s)\} : (i,s) \in \mathfrak{F} \setminus \mathfrak{F}_{\text{ls}}\} \bigcup \\ &\{\mathfrak{F}_{\text{ls}} \setminus \{(i,s)\} : (i,s) \in \mathfrak{F}_{\text{ls}}, \mathfrak{F}_{\text{ls}} \setminus \{(i,s)\} \\ &\quad \text{can be connected by at least } n-q \text{ clients}\} \bigcup \\ &\quad \{\mathfrak{F}_{\text{ls}} \setminus \{(i,s)\} \cup \{(i',s')\} : (i,s) \in \mathfrak{F}_{\text{ls}}, \\ &\quad (i',s') \in \mathfrak{F} \setminus \mathfrak{F}_{\text{ls}}, \mathfrak{F}_{\text{ls}} \setminus \{(i,s)\} \cup \{(i',s')\} \\ &\quad \text{can be connected by at least } n-q \text{ clients}\}\end{aligned}$$
If there exists some facility set $\mathfrak{F}' \in \mathcal{N}(\mathfrak{F}_{\text{ls}})$ satisfying $T(\mathfrak{F}') < T(\mathfrak{F})$, update $\mathfrak{F}_{\text{ls}} := \mathfrak{F}'$ and repeat Step 2.2. Otherwise, update $y^{\text{ls}}_{is} := 1$ for each facility $(i,s) \in \mathfrak{F}_{\text{ls}}$, and go to Step 3.

Step 3: Determine outliers.

Update $\mathfrak{O}_{\text{ls}} := \mathfrak{C} \setminus \mathfrak{C}^{\min}_{\mathfrak{F}_{\text{ls}}}$, where $\mathfrak{C}^{\min}_{\mathfrak{F}_{\text{ls}}}$ are the first $n-q$ clients in $\mathfrak{C}$ with the smallest connection cost of $c_{\tau(\mathfrak{F}_{\text{ls}},j)j}$. For each client $(j,l) \in \mathfrak{O}_{\text{ls}}$, update $z^{\text{ls}}_{jl} := 1$.

Step 4: Connect clients.

Update $\mathfrak{C}_{\text{ls}} := \mathfrak{C}^{\min}_{\mathfrak{F}_{\text{ls}}}$. For each client $(j,l) \in \mathfrak{C}_{\text{ls}}$, find facility
$$(i,s) := \arg \min_{(i',s') \in \mathfrak{F}_{\text{ls}}: l \leqslant s'} c_{i'j},$$
set $\sigma_{\text{ls}}(j,l) := (i,s)$, and update $x^{\text{ls}}_{is,jl} := 1$.

Step 5: Output primal variables $\{x^{\text{ls}}_{is,jl}\}_{(i,s)\in\mathfrak{F},(j,l)\in\mathfrak{C}}$, $\{y^{\text{ls}}_{is}\}_{(i,s)\in\mathfrak{F}}$, and $\{z^{\text{ls}}_{jl}\}_{(j,l)\in\mathfrak{C}}$.

5 Experimental Simulation

In order to illustrate the performance of all the proposed algorithms, we compare the experimental results of them running on synthetic data sets. By randomly generating the PFLPO instances with different number of clients, outliers and facilities, the experiments aim to observe the effect of each number on the total cost. For all the generated instances, we set the maximum level to be 3.

5.1 Effect of the number of clients

In this experiment, we fix the number of facilities at 50 and vary the number of clients. Figure 1 shows the changing of the total cost of each algorithm. As depicted in the figure, our primal-dual algorithm consistently outperforms the other two heuristic algorithms, achieving the lowest total cost.

5.2 Effect of the number of outliers

In this experiment, we fix the number of facilities at 50 and the number of clients at 1000. We introduce outliers by varying the number of them. Figure 2 presents the changing of the total cost of each algorithm. Interestingly, the curves for the three algorithms intersect, indicating that the performance of the algorithms may be influenced by the presence of outliers. Our primal-dual algorithm incorporates a random selection mechanism when adding outliers, while the other two heuristic algorithms employ a greedy approach to select the optimal solution at each step. The difference in the selection strategy may lead to the intersections in the total cost curves.

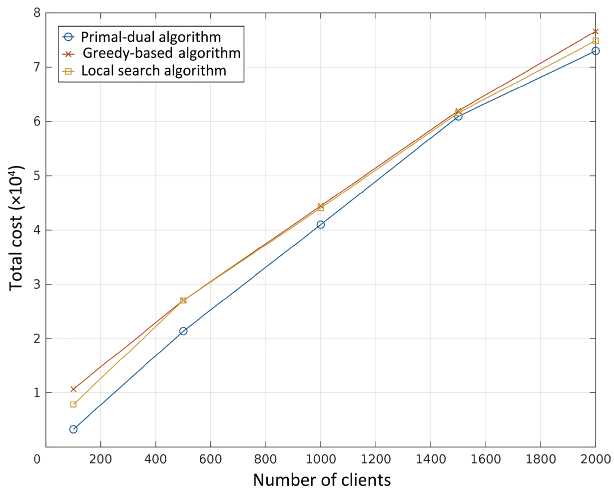


Fig. 1 Effect of number of clients.

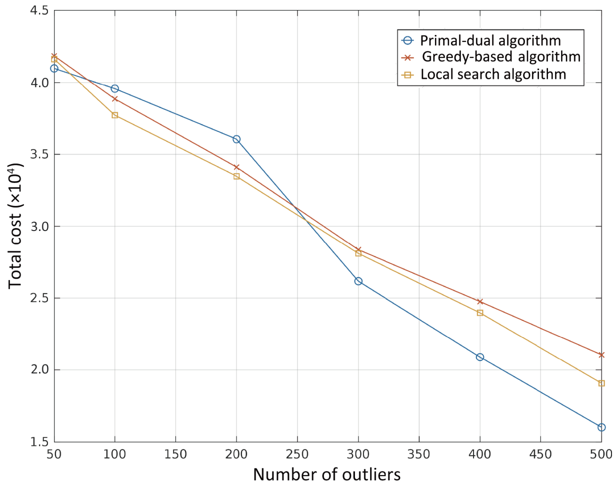


Fig. 2 Effect of number of outliers.

5.3 Effect of the number of facilities

In this experiment, we fix the number of clients at 500 and the number of outliers at 50. We vary the number of facilities. Figure 3 illustrates the changing of the total cost for each algorithm. The results indicate that our primal-dual algorithm consistently outperforms the other two heuristic algorithms, with a significant advantage. It is worth mentioning that the greedy-based algorithm may achieve better solutions compared with the primal-dual algorithm while the number of facilities is not very large. This phenomenon can be attributed to the complexity of the primal-dual algorithm of selecting the opened facilities. However, as the number of facilities increases, the greedy-based algorithm tends to get trapped in a local optimal solution, resulting in a larger cost gap. Therefore, the performance of the greedy-based algorithm tends to be deteriorated as the number of facilities increases.

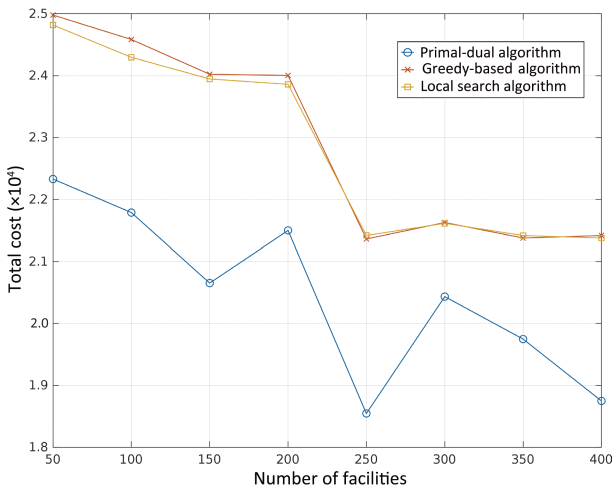


Fig. 3 Effect of number of facilities.

Acknowledgment

This work was supported by the Beijing Municipal Natural Science Foundation (No. Z220004), the National Natural Science Foundation of China (Nos. 12371321, 12171051, and 12171052), and the Research Innovation Fund for College Students of Beijing University of Posts and Telecommunications.

References

[1] P. B. Mirchandani and R. L. Francis, *Discrete location theory*. New York, NY, USA: John Wiley & Sons, 1990.

[2] D. B. Shmoys, E. Tardos, and K. Aardal, Approximation algorithms for facility location problems, in *Proc. of the 29th Annual ACM Symposium on Theory of Computing*, El Paso, TX, USA, 1997, pp. 265–274.

[3] S. Li, A 1.488 approximation algorithm for the uncapacitated facility location problem, *Inf. Comput*., vol. 222, pp. 45–58, 2013.

[4] M. Sviridenko, An improved approximation algorithm for the metric uncapacitated facility location problem, in *Proc. of the 9th Int. Conf. on Integer Programming and Combinatorial Optimization*, Copenhagen, Denmark, 2002. pp. 240–257,

[5] J. Byrka and K. Aardal, An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem, *SIAM J. Comput.*, vol. 39, no. 6, pp. 2212–2231, 2010.

[6] F. A. Chudak and D. B. Shmoys, Improved approximation algorithms for the uncapacitated facility location problem, *SIAM J. Comput.*, vol. 33, no. 1, pp. 1–25, 2003.

[7] K. Jain and V. V. Vazirani, Approximation algorithms for metric facility location and *k*-Median problems using the primal-dual schema and Lagrangian relaxation, *J. ACM*, vol. 48, no. 2, pp. 274–296, 2001.

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

[9] K. Jain, M. Mahdian, and A. Saberi, A new greedyapproach for facility location problems, in *Proc. of the 34th Annual ACM Symposium on Theory of Computing*, Montréal, Canada, 2002, pp. 731–740.

[10] M. Mahdian, Y. Ye, and J. Zhang, Improved approximation algorithms for metric facility location problems, in *Proc. of 5th International Workshop, APPROX 2002*, Rome, Italy, 2002, pp. 229–242.

[11] V. Arya, N. Garg, R. Khandekar, A. Meyerson, and K. Munagala, Local search heuristic for k-median andfacility location problems, in *Proc. of the 33th Annual ACM Symposium on Theory of Computing*, Heraklion, Greece, 2001, pp. 21–29.

[12] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman, Analysis of a local search heuristic for facility location problems, *J. Algoritms.*, vol. 37, no. 1, pp. 146–188, 2000.

[13] M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan, Algorithms for facility location problems with outliers, in *Proc. of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, Washington, DC, USA, 2001, pp. 642–651.

[14] R. Ravi and A. Sinha, Multicommodity facility location, in *Proc. of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, New Orleans, LA, USA, 2004, pp. 342–349.

[15] M. Mahdian, Facility location and the analysis of algorithms through factor-revealing programs, PhD dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 2004.

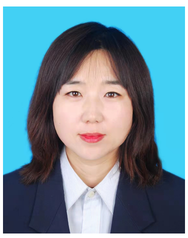
[16] G. Li, Z. Wang, and C. Wu, Approximation algorithms for the stochastic priority facility location problem, *Optimization*, vol. 62, no. 7, pp. 919–928, 2013.

[17] F. Wang, D. Xu, and C. Wu, Approximation algorithms for the priority facility location problem with penalties, *J. Syst. Sci. Complex.*, vol. 28, no. 5, pp. 1102–1114, 2015.

Hang Luo is an undergraduate student at Beijing University of Posts and Telecommunications, China. His research interests include approximation algorithm, machine learning, etc.

Tianping Shuai received the PhD degree from Academy of Mathematics and Systems Science, Chinese Academy of Sciences, China in 2005. Currently he is an associate professor at Beijing University of Posts and Telecommunications, China. His research interests include combinatorial optimization, graph theory, etc.

Lu Han received the PhD degree from Beijing University of Technology, China in 2019. Currently she is an associate professor at Beijing University of Posts and Telecommunications, China. Her research interests include combinatorial optimization, approximation algorithm, etc.



Fengmin Wang received the PhD degree from Beijing University of Technology, China in 2016. Currently she is a senior engineer at Beijing Jinghang Research Institute of Computing and Communication, China. Her research interests include approximation algorithm, machine learning, etc.