Journal of the Operations Research Society of China
https://doi.org/10.1007/s40305-023-00478-0

Approximation Algorithm for the k-Product Uncapacitated Facility Location Problem with Penalties

Pei-Jia Yang[1] **· Wen-Chang Luo**[1]

Received: 26 July 2022 / Revised: 21 February 2023 / Accepted: 23 February 2023

Abstract
In the k-product uncapacitated facility location problem with penalties, we are given a set of demand points where clients are located and a set of potential sites where facilities with unlimited capacities can be opened. There are k different kinds of products to be supplied by a set of open facilities. Each open facility can supply only a distinct product with a non-negative fixed cost determined by the product it wants to supply. Each client is either supplied with k kinds of products by a set of k different open facilities or completely rejected. There is a non-negative service cost between each pair of locations and also a penalty cost for each client if its service is rejected. These service costs are assumed to be symmetric and satisfy the triangle inequality. The goal is to select a set of clients to reject their service and then choose a set of facilities to be opened to service the remaining clients so that the total cost of opening facilities, servicing the clients, and the penalty is minimized. We address two different integer programs to describe the problem. Based on the linear programming rounding technique, we propose a $(2k+1)$-approximation algorithm for this problem.

Keywords k-product · Facility location · Penalty · LP-rounding · Approximation algorithm

Mathematics Subject Classification 90B80 · 90C10

This work was supported by the National Natural Science Foundation of China (No. 11971252).

✉ Wen-Chang Luo
luowenchang@163.com

Pei-Jia Yang
1126905056@qq.com

[1] School of Mathematics and Statistics, Ningbo University, Ningbo 315211, Zhejiang, China

Published online: 03 April 2023

1 Introduction

The facility location problem is concerned with where to select a set of facilities to be opened and how to assign a set of clients for each open facility to be serviced so that the total cost which includes the open cost of facilities and the service cost of clients is minimized. The facility location problem is a classical NP-hard problem in combinatorial optimization and has many applications in operations research, management science, and computer science.

In the recent twenty years, a number of approximation algorithms with constant performance ratio have been proposed when the service cost is assumed to be in the metric space. By using the filtering technique, Shmoys et al. [1] gave the first constant factor approximation algorithm with the performance ratio of 3.16. After that, a large number of improved approximation algorithms were proposed by Jain and Vazirani [2], Guha and Khuller [3], Charikar and Guha [4], Korupolu et al. [5], Jain et al. [6], Mahdian et al. [7], Chudak and Shmoys [8], and Byrka and Aardal [9]. The current best known approximation ratio is 1.488, proposed by Li [10], which is close to the lower bound 1.463 derived by Guha and Khuller [3].

In many real applications, a client usually requires more than one product to be serviced by several different facilities. This problem can be viewed as multi-product uncapacitated facility location problem or k-product uncapacitated facility location problem. In this problem, each client has k different products in demands to be supplied and for each open facility, only one product can be serviced. The k-product uncapacitated facility location problem is stated as a variant of the uncapacitated facility location problem, which was presented as a special case of the submodular set function in [11]. Obviously, the classical uncapacitated facility location problem is a special case of k-product uncapacitated facility location problem with $k = 1$. The another variant of the uncapacitated facility location problem is the k-level uncapacitated facility location problem, in which each client must be serviced by a sequence of k different facilities [12]. In the k-product uncapacitated facility location problem and the k-level uncapacitated facility location problem, each client is required to be serviced by k different facilities. In the former the k different facilities are at the same level one, but in the latter the k different facilities constitute a sequence in which each is at the different level.

Klincewicz et al. [13] and Klincewicz and Luss [14] have proposed various optimal and heuristic algorithms for the k-product uncapacitated facility location problem in general space. As for the k-product capacitated facility location problem, it has been investigated by Lee [15, 16] and Mazzola and Neebe [17]. Huang and Li [18] studied the k-product uncapacitated facility location problem in metric space. When the fixed open costs are zero, they gave a $(2k - 1)$-approximation algorithm; for the general case, they derived a $(2k + 1)$-approximation algorithm. Nezhad et al. [19] proposed a Lagrangian relaxation heuristic algorithm. Yang et al. [20] gave a heuristic algorithm based on linear programming relaxation solutions, both of which reduce the time for solving the algorithm significantly and improve the efficiency of solving the problem by optimizing the upper and lower bounds of feasible solutions.

In real-life applications, some clients may be far from the facilities, who may choose not to serve these clients by paying some penalty costs. Under this consideration, it leads naturally to the uncapacitated facility location problem with penalties (or outliers).

For this topic, Charikar et al. [21] gave a primal dual algorithm with performance guarantee of 3 for the facility location problem with outliers. Xu et al. [22] proposed a linear programming rounding algorithm to obtain a performance guarantee of 2.736.

In this paper, we consider the k-product uncapacitated facility location problem with penalties (k-PUFLPWP) in the metric space and propose an approximation algorithm for this problem. The organization of this paper is as follows. In Sect. 2, we formulate the problem and derive the standard integer programming model. In Sect. 3, we derive another formulation for this problem in a different way and present an approximation algorithm with a performance guarantee of $(2k+1)$. In the final section, we conclude this paper and suggest some further research directions.

2 The Formulation for the k-PUFLPWP

In this section, we formally describe the k-product uncapacitated facility location problem with penalties and then derive its standard integer programming model.

In the k-PUFLPWP, we are given a set of demands points J where clients are located and a set of potential sites I where facilities with unlimited capacities can be opened. There are k different kinds of products $p_l, l = 1, 2, \cdots, k$ to be supplied by a set of open facilities. Each open facility at site $i \in I$ can supply only a distinct product with a non-negative fixed cost determined by the product it wants to supply. The cost of opening the facility at site i to supply the product p_l is f_{li} $(\geqslant 0)$, $l = 1, 2, \cdots, k$, $i \in I$. The service cost for client j to be supplied any kind of product by the open facility at site i is c_{ij}, $i \in I$, $j \in J$. Each client $j \in J$ should be either supplied with k different kinds of products by a set of k different open facilities or completely rejected. It should be reminded that splitting source at different open facilities is not allowed for a given product. If the client $j \in J$ is completely rejected then one should pay a penalty cost e_j, $j \in J$. Throughout this paper, the following assumptions on costs are required:

(a) $f_{li} \geqslant 0$, for each $i \in I, l = 1, 2, \cdots, k$;
(b) $c_{ij} \geqslant 0$, for each $i, j \in I \cup J$;
(c) $c_{ij} = c_{ji}$, for each $i, j \in I \cup J$, i.e., the service costs are symmetric;
(d) $c_{ij} \leqslant c_{ir} + c_{rj}$, for each $i, r, j \in I \cup J$, i.e., the service costs satisfy the triangle inequality;
(e) $e_j \geqslant 0$, for each $j \in J$.

The goal is to minimize the total cost which includes the open cost for opening the facilities at sites, the service cost for supplying the products from open facilities to the clients, and the penalty cost for rejecting some clients.

Below, we use the 0 –1 variable x_{lij} to denote whether the open facility at site i supplies the product p_l to client j, where $x_{lij} = 1$ means that the open facility at site i supplies the product p_l to the client j, and $x_{lij} = 0$ otherwise for any $l = 1, 2, \cdots, k, i \in I, j \in J$; the 0 –1 variable y_{li} indicates whether the facility at site i is opened to supply product p_l, where $y_{li} = 1$ means that the facility at site

i is opened to supply the product p_l to the clients, and $y_{li} = 0$ otherwise for any $l = 1, 2, \cdots, k$, $i \in I$; the 0–1 variable z_j indicates whether the client j is completely rejected, where $z_j = 1$ means that the client j is completely rejected, and $z_j = 0$ otherwise for any $j \in J$. Then, the k-product uncapacitated facility location problem with penalties can be formulated as the following standard integer programming model (P0):

$$\text{(P0)} \quad \min \sum_{l=1}^{k} \sum_{i \in I} f_{li} y_{li} + \sum_{l=1}^{k} \sum_{i \in I} \sum_{j \in J} c_{ij} x_{lij} + \sum_{j \in J} e_j z_j \tag{1}$$

$$\text{s.t.} \quad \sum_{i \in I} x_{lij} + z_j = 1, \quad \forall j \in J, l = 1, 2, \cdots, k; \tag{2}$$

$$x_{lij} \leqslant y_{li}, \quad \forall i \in I,\ j \in J,\ l = 1, 2, \cdots, k; \tag{3}$$

$$\sum_{l=1}^{k} y_{li} \leqslant 1, \quad \forall i \in I; \tag{4}$$

$$x_{lij}, y_{li}, z_j \in \{0, 1\}, \quad \forall i \in I,\ j \in J,\ l = 1, 2, \cdots, k. \tag{5}$$

In the above formulation (P0), constraint (2) ensures that for each client its service is either to be satisfied by supplying k different kinds of products subject to each product being precisely supplied by one open facility or completely rejected. Constraint (3) ensures that the facility at site i is opened to supply product p_l if client j is serviced with product p_l from the facility at site i. Constraint (4) ensures that the facility at any site is opened to supply at most one kind of product. Constraint (5) ensures that all decision variables are 0 –1 variables, i.e., binary variables. Clearly, if there is a feasible solution to the studied problem, then the number of open facilities at sites is greater than or equal to the number of different products, i.e., $|I| \geqslant k$.

3 Approximation Algorithm for the *k*-PUFLPWP

In this section, to derive our algorithm for the k-PUFLPWP, we turn to address another formulation for the k-PUFLPWP.

Below we use $s = (i_1, i_2, \cdots, i_k)$ to represent the sequence of k different open facilities at sites $i_l \in I, l = 1, 2, \cdots, k$. We also refer to s as a feasible sequence of open facilities, and the set of all feasible sequences is written as S. Given a sequence s, we use s^l to represent the lth open facility at site i_l in s, which means that s use the open facility at site i_l to supply the product $p_l, l = 1, 2, \cdots, k$. In any feasible solution of the k-PUFLPWP, for each client j, it should be assigned to a feasible sequence $s = (i_1, i_2, \cdots, i_k) \in S$ and then we denote the total cost incurred by the client j in this assignment as c_{sj} which is equal to $c_{i_1 j} + c_{i_2 j} + \cdots + c_{i_k j}$, i.e., $c_{sj} = \sum_{l=1}^{k} c_{i_l j}$.

Let x_{sj} denote whether client j is assigned to sequence s, where $x_{sj} = 1$ means that client j is assigned to sequence s, and 0 otherwise. Again let y_{li} be 1 if the facility at site i is opened to supply product p_l, and 0 otherwise.

By using the notations defined as above, the k-PUFLPWP can be formulated by the another integer programming model:

$$\text{(P1)}\quad \min \quad \sum_{l=1}^{k}\sum_{i\in I} f_{li}y_{li} + \sum_{s\in S}\sum_{j\in J} c_{sj}x_{sj} + \sum_{j\in J} e_j z_j \tag{6}$$

$$\text{s.t.} \sum_{s\in S} x_{sj} + z_j = 1, \quad \forall j \in J; \tag{7}$$

$$\sum_{s:s^l=i} x_{sj} \leqslant y_{li}, \quad \forall i \in I,\ j \in J,\ l = 1, 2, \cdots, k; \tag{8}$$

$$\sum_{l=1}^{k} y_{li} \leqslant 1, \quad \forall i \in I; \tag{9}$$

$$x_{sj}, y_{li}, z_j \in \{0, 1\}, \quad \forall i \in I,\ j \in J,\ l = 1, 2, \cdots, k. \tag{10}$$

In the above formulation (P1), constraint (7) ensures that for each client its service is either to be satisfied by a sequence s in S with supplying k different kinds of products or completely rejected. Constraint (8) ensures that the facility at site i is opened to supply product p_l if client j is serviced with product p_l from the facility at site i_l in sequence s. Constraint (9) ensures that the facility at any site is opened to supply at most one kind of product. Constraint (10) ensures that all decision variables are 0–1 variables, i.e., binary variables.

Now, we relax (P1) into the following linear programming model by removing constraint (9) and the integer conditions in constraint (10). We denote this linear programming as (P2).

$$\text{(P2)}\quad Z_{\text{LP}} = \min \quad \sum_{l=1}^{k}\sum_{i\in I} f_{li}y_{li} + \sum_{s\in S}\sum_{j\in J} c_{sj}x_{sj} + \sum_{j\in J} e_j z_j \tag{11}$$

$$\text{s.t.} \sum_{s\in S} x_{sj} + z_j = 1, \quad \forall j \in J; \tag{12}$$

$$\sum_{s:s^l=i} x_{sj} \leqslant y_{li}, \quad \forall i \in I,\ j \in J,\ l = 1, 2, \cdots, k; \tag{13}$$

$$x_{sj}, y_{li}, z_j \geqslant 0, \quad \forall i \in I,\ j \in J,\ l = 1, 2, \cdots, k. \tag{14}$$

Denote the dual variables corresponding to the primal constraints (12) and (13) in (P2) as v_j, $j \in J$ and w_{lij}, $i \in I$, $j \in J$, $l = 1, 2, \cdots, k$, respectively. Then, the dual linear program problem corresponding to (P2) is derived in the following by (P3).

$$\text{(P3)}\quad \max \sum_{j\in J} v_j \tag{15}$$

$$\text{s.t.} \quad \sum_{j\in J} w_{lij} \leqslant f_{li}, \quad \forall i \in I,\ l = 1, 2, \cdots, k; \tag{16}$$

$$v_j - \sum_{l=1}^{k} w_{ls^l j} \leqslant c_{sj}, \quad \forall s \in S,\ j \in J; \tag{17}$$

$$v_j \leqslant e_j, \quad \forall j \in J; \tag{18}$$

$$w_{lij} \geqslant 0, \quad \forall i \in I,\ j \in J,\ l = 1, 2, \cdots, k. \tag{19}$$

It is known that for a fixed k, we can solve (P2) and its dual (P3) in polynomial time [23]. Also, these two linear programs have the same optimal value Z_{LP}, and this value provides a lower bound for the integer program (P1).

With the above preparation, we turn to describe **AlgorithmH**.

In **Algorithm H**, we first determine the rejected client set according to the fractional optimal solution of (P3) and then set the center client in the remaining clients for each iteration, so as to satisfy all the remaining clients' services. Finally, we obtain an integer feasible solution to our problem.

Algorithm H

Step 1 First solve (P2) and (P3) and then define (x^*, y^*, z^*) and (v^*, w^*) be the optimal solutions to (P2) and (P3), respectively. Let $C_j^* = \sum_{s \in S} c_{sj} x_{sj}^*,\ \forall j \in J$, $S^{(j)} = \left\{ s \in S | x_{sj}^* > 0 \right\}$, $I^{(j)} = \{ i \in I | i$ belongs to at least one feasible sequence in $S^{(j)} \}$.

Step 2 Let $R = \left\{ j | v_j^* = e_j \right\}$ and reject all the clients in R. Let $A = J \backslash R$ be the set of clients whose service would be satisfied. Set $z_j = 1$, $j \in R$ and $z_j = 0$, $j \in A$. Initially, we set $\bar{A} := A,\ x := x^*,\ y := y^*$. Set $t = 1$.

Step 3 In iteration t, from $\bar{A}$, we select the client j with the minimum value of $kv_j^* + C_j^*$ and denote this client as j_t, which is referred to as the center client in the iteration t. Set $A_t = \left\{ j \in \bar{A} | I^{(j)} \cap I^{(j_t)} \neq \varnothing \right\}$. With probability $x_{sj_t}^*$, we choose a feasible sequence $s \in S^{(j_t)}$ and denote this sequence as s_t, which is referred to the center sequence in the iteration t. Then, we assign all the clients j in A_t to be serviced by s_t, which means that we set

$$y_{li} = 1, \ \ i = s_t^l, \qquad y_{li} = 0, \ \ \forall\, i \in I^{(j_t)} \text{ and } i \neq s_t^l, \tag{20}$$

$$x_{s_t j} = 1, \ \ \forall j \in A_t, \qquad x_{sj} = 0, \ \ \forall\, j \in A_t \text{ and } s \in S \backslash \{s_t\}. \tag{21}$$

Step 4 Update $\bar{A} = \bar{A} \backslash A_t$. If $\bar{A} \neq \varnothing$, then update $t = t + 1$ and go to Step 3; otherwise, output the final solution (x, y, z).

Next, we devote to analyze the performance ratio of **Algorithm H**.

Lemma 1 *For each $j \in J,\ s \in S$, if $x_{sj}^* > 0$ holds, then we have $v_j^* \geqslant c_{sj}$.*

Proof Note that (x^*, y^*, z^*) and (v^*, w^*) are the optimal solutions for (P2) and its dual (P3). By using complementary slackness conditions, we have

$$x_{sj}^* \left(v_j^* - \sum_{l=1}^{k} w_{ls^l j}^* - c_{sj} \right) = 0. \tag{22}$$

Due to $x^*_{sj} > 0$, we obtain $v^*_j = \sum_{l=1}^{k} w^*_{ls^l j} + c_{sj}$. With $w^*_{ls^l j} \geqslant 0$, we achieve $v^*_j \geqslant c_{sj}$.

Lemma 2 *When* **Algorithm H** *terminates, a feasible integer solution for problem* (P1) *is obtained.*

Proof It is easy to verify that, upon termination of **Algorithm H**, the solution (x, y, z) obtained by the **Algorithm H** is integral and satisfies constraints (7) and (8) of the integer program (P1).

Next, we will verify that when the algorithm terminates, the integer solution (x, y, z) satisfies constraint (9) of the integer program (P1). In each iteration t, since s_t is a feasible sequence, no facility can be opened to provide more than one kind of product. In any two different iterations t and $\hat{t}$, since $I^{(j_t)}$ and $I^{(j_{\hat{t}})}$ are disjoint, there is no common facility in s_t and $s_{\hat{t}}$. Thus, when the algorithm terminates, no facility is opened to provide more than one kind of product, which implies the obtained solution satisfing constraint (9) in the integer program (P1). In the end, (x, y, z) is a feasible integer solution to (P1) when the algorithm terminates.

Theorem 1 ***Algorithm H*** *outputs a feasible integer solution to* (P1) *with an expected total cost not exceeding* $(2k + 1)$ *times the optimal value of* (P2).

Proof We divide the value of objective function into three parts, that is, the cost of opening facilities, the cost of serving the clients, and the penalty cost of rejecting clients, which are denoted as F, D, and E respectively. Furthermore, we denote the cost of opening facilities incurred at the iteration t as F_t, the cost of serving the clients incurred at the iteration t as D_t.

First, we derive the upper bound on the expected cost on F_t. Before the modification of the solution (x, y) in iteration t, we have

$$x_{sj} = x^*_{sj}, \forall s \in S^{(j_t)}, j \in A_t, \tag{23}$$

$$y_{li} = y^*_{li}, \forall i \in I^{(j_t)}, l = 1, \cdots, k. \tag{24}$$

For the selected feasible sequence $s_t \in S^{(j_t)}$, the cost of opening facilities is $\sum_{l=1}^{k} f_{ls^l_t}$. According to the rules of **Algorithm H**, each $s \in S^{(j_t)}$ is selected with probability $x^*_{sj_t}$. Thus, the expected cost of opening facilities F_t incurred in iteration t for solution (x, y) is equal to

$$F_t = \sum_{s \in S^{(j_t)}} \left(\sum_{l=1}^{k} f_{ls^l_t} \right) x^*_{sj_t} = \sum_{l=1}^{k} \sum_{i \in I^{(j_t)}} f_{li} \left(\sum_{s:s^l=i} x^*_{sj_t} \right) \leqslant \sum_{l=1}^{k} \sum_{i \in I^{(j_t)}} f_{li} y^*_{li}, \tag{25}$$

where the inequality follows from constraint (13). Summing up the above F_t on t, we have

$$F \leqslant \sum_{l=1}^{k} \sum_{i \in I} f_{li} y^*_{li}. \tag{26}$$

Next, we derive the upper bound on D_t. In iteration t, for the center client j_t, its expected service cost is

$$\sum_{s\in S^{(j_t)}} c_{sj_t}x^*_{sj_t} = C^*_{j_t}. \tag{27}$$

For each $j \in A_t$ with $j \neq j_t$, the feasible sequence s to which it is assigned in the optimal solution has at least one facility in $I^{(j_t)}$. Now, we assume that $i \in I^{(j_t)}$ is the facility in s that supplies the lth product to client j, i.e., $i = s^l$. By the definition of $I^{(j_t)}$, there exists at least one feasible sequence $\bar{s} \in S^{(j_t)}$, where facility i supplies product p_l to client j_t, i.e., $i = \bar{s}^l$. Thus, the service cost incurred by client j is

$$c_{s_t j} = \sum_{l=1}^{k} c_{s_t^l j} \leqslant \sum_{l=1}^{k}\left(c_{s_t^l j_t} + c_{j_t i} + c_{ij}\right) \leqslant \sum_{l=1}^{k}\left(c_{s_t^l j_t} + c_{\bar{s} j_t} + c_{sj}\right) \tag{28}$$

$$\leqslant \sum_{l=1}^{k}\left(c_{s_t^l j_t} + v^*_{j_t} + v^*_j\right) = \sum_{l=1}^{k} c_{s_t^l j_t} + kv^*_{j_t} + kv^*_j \tag{29}$$

$$= c_{s_t j_t} + kv^*_{j_t} + kv^*_j, \tag{30}$$

where the first inequality follows from the triangle inequality, the second inequality follows from the fact that i is a facility in s and $\bar{s}$, and the last inequality follows from Lemma 1. Note that the probability of selecting $s \in S^{(j_t)}$ is $x^*_{sj_t}$, the upper bound on the expected service cost of client j is

$$\sum_{s\in S^{(j_t)}}\left(c_{s_t j_t} + kv^*_{j_t} + kv^*_j\right)x^*_{sj_t} = C^*_{j_t} + kv^*_{j_t} + kv^*_j \leqslant C^*_j + 2kv^*_j, \tag{31}$$

where the inequality follows from the selection rule of the center client in Step 2 of the **Algorithm H**. Thus, we obtain

$$D_t \leqslant \sum_{j\in A_t}\left(C^*_j + 2kv^*_j\right) = 2k\sum_{j\in A_t} v^*_j + \sum_{s\in S}\sum_{j\in A_t} c_{sj}x^*_{sj}. \tag{32}$$

Summing up the above D_t on t, we have

$$D \leqslant 2k\sum_{j\in A} v^*_j + \sum_{s\in S}\sum_{j\in A} c_{sj}x^*_{sj}. \tag{33}$$

Finally, we consider the penalty cost of rejecting clients. We have

$$E = \sum_{j\in R} e_j = \sum_{j\in R} v^*_j. \tag{34}$$

Now, summing up the above F, D,and E, the total expected cost is upper bounded by

$$
\begin{aligned}
&F + D + E && (35)\\
&\leqslant 2k \sum_{j\in A} v_j^* + \sum_{l=1}^{k}\sum_{i\in I} f_{li} y_{li}^* + \sum_{s\in S}\sum_{j\in A} c_{sj} x_{sj}^* + \sum_{j\in R} v_j^* && (36)\\
&\leqslant 2k \sum_{j\in A} v_j^* + \sum_{l=1}^{k}\sum_{i\in I} f_{li} y_{li}^* + \sum_{s\in S}\sum_{j\in J} c_{sj} x_{sj}^* + \sum_{j\in R} v_j^* && (37)\\
&\leqslant (2k-1) \sum_{j\in A} v_j^* + \sum_{j\in J} v_j^* + \sum_{l=1}^{k}\sum_{i\in I} f_{li} y_{li}^* + \sum_{s\in S}\sum_{j\in J} c_{sj} x_{sj}^* + \sum_{j\in J} e_j z_j^* && (38)\\
&\leqslant (2k-1) \sum_{j\in A} v_j^* + \sum_{j\in J} v_j^* + \sum_{j\in J} v_j^* && (39)\\
&\leqslant (2k-1) \sum_{j\in A} v_j^* + 2\sum_{j\in J} v_j^* && (40)\\
&= (2k+1) \sum_{j\in J} v_j^* && (41)\\
&= (2k+1)\, Z_{\mathrm{LP}}. && (42)
\end{aligned}
$$

Thus, Theorem 3 is proven.

To derandomize **Algorithm H**, we first obtain an optimal basic feasible solution for a fixed k. Then, we choose the sequence that minimizes the total cost for the cluster, and we can achieve a deterministic $(2k+1)$-approximation algorithm for the k-PUFLPWP. A more detailed similar proof can be found in [12].

4 Concluding Remarks

In this paper, we investigate the k-PUFLPWP and present an approximation algorithm based on LP-rounding with the performance ratio of $(2k+1)$. When k is a fixed number, our proposed algorithm runs in polynomial time. One interesting question for future research direction is whether there exists a better approximation algorithm with constant performance ratio independent of k.

Acknowledgements The authors are grateful to the editor and the reviewers for their helpful comments and suggestions, which have improved the presentation of the paper.

Author Contributions All authors contributed equally to this paper.

Conflict of interest The authors confirm that this article content has no conflict of interest.

References

[1] Shmoys, D.B., Tardos, É., Aardal, K.: Approximation algorithms for facility location problems. In: the 29th Annual ACM Symposium on Theory of Computing, pp. 265–274. ACM Press, Berlin (1997)

[2] Jain, K., Vazirani, V.V.: Primal–dual approximation algorithms for metric facility location and k-median problems. In: The 40th Annual IEEE Symposium on Foundations of Computer Science, pp. 2–13. ACM Press, USA (1999)

[3] Guha, S., Khuller, S.: Greedy strikes back: improved facility location algorithms. J. Algorithms **31**, 228–248 (1999)

[4] Charikar, M., Guha, S.: Improved combinatorial algorithms for the facility location and k-median problems. In: 40th Annual IEEE Symposium on Foundations of Computer Science, pp. 378–388. ACM Press, USA (1999)

[5] Korupolu, M.R., Plaxton, C.G., Rajaraman, R.: Analysis of a local search heuristic for facility location problems. J. Algorithms **37**(1), 146–188 (2000)

[6] Jain, K., Mahdian, M., Saberi, A.: A new greedy approach for facility location problem. In: the 34th ACM Symposium on Theory of Computing (STOC), pp. 731–740. ACM Press, New York (2002)

[7] Mahdian, M., Ye, Y., Zhang, J.: Improved approximation algorithms for metric facility location problems. In: the 5th International Workshop on Approximation Algorithms for Combinatorial Optimization, pp. 229–242. ACM Press, Berlin (2002)

[8] Chudak, F.A., Shmoys, D.B.: Improved approximation algorithms for the uncapacitated facility location problem. SIAM J. Comput. **33**(1), 1–25 (2004)

[9] Byrka, J., Aardal, K.: An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. SIAM J. Comput. **39**(6), 2212–2231 (2010)

[10] Li, S.: A 1.488 approximation algorithm for the uncapacitated facility location problem. Inf. Comput. **222**, 45–58 (2013)

[11] Fisher, M.L., Nemhauser, G.L., Wolsey, L.A.: An analysis of approximation for maximizing submodular set functions II. Math. Program. Study **8**, 73–87 (1978)

[12] Ardal, K., Chudak, F.A., Shmoys, D.B.: A 3-approximation algorithm for the k-level uncapacitated facility location problem. Inf. Process. Lett. **72**(5/6), 161–167 (1999)

[13] Klincewicz, J.G., Luss, H., Rosenberg, E.: Optimal and heuristic algorithms for multiproduct uncapacitated facility location. Eur. J. Oper. Res. **26**(2), 251–258 (1986)

[14] Klincewicz, J.G., Luss, H.: A dual-based algorithm for multiproduct uncapacitated facility location. Transp. Sci. **21**(3), 198–206 (1987)

[15] Lee, C.Y.: An optimal algorithm for the multiproduct capacitated facility location problem with a choice of facility type. Comput. Oper. Res. **18**(2), 167–182 (1991)

[16] Lee, C.Y.: A cross decomposition algorithm for a multiproduct-multitype facility location problem. Comput. Oper. Res. **20**, 527–540 (1993)

[17] Mazzola, J.B., Neebe, A.W.: Lagrangian-relaxation-based solution procedures for a multiproduct capacitated facility location problem with choice of facility type. Eur. J. Oper. Res. **115**(2), 285–299 (1999)

[18] Huang, H.C., Li, R.: A k-product uncapacitated facility location problem. Eur. J. Oper. Res. **185**, 552–562 (2008)

[19] Nezhad, A.M., Manzour, H., Salhi, S.: Lagrangian relaxation heuristics for the uncapacitated single-source multi-product facility location problem. Int. J. Prod. Econ. **145**, 713–723 (2013)

[20] Yang, M.M., Huang, Y.K., Dai, Y.H.: Linear relaxation solution based heuristics for a class of multi-product facility location problems. Oper. Res. Trans. **23**(3), 15–26 (2019)

[21] Charikar, M., Khuller, S., Mount, D., Narasimhan, G.: Algorithms for facility location problems with outliers. In: 12th ACM-SIAM Symposium on Discrete Algorithms, pp. 642-651. ACM Press, Washington (2001)

[22] Xu, G., Xu, J.: An LP rounding algorithm for approximating uncapacitated facility location problem with penalties. Inf. Process. Lett. **94**(3), 119–123 (2005)

[23] Fang, S.C., Puthenpura, S.: Linear Optimization and Extensions: Theory and Algorithms. Prentice Hall, Hoboken (1993)