Toward Balancing the Efficiency and Effectiveness in k-Facility Relocation Problem

HU WANG and HUI LI, Xidian University, China
MENG WANG, Xi'an Polytechnic University, China
JIANGTAO CUI, Xidian University, China

Facility Relocation (FR), which is an effort to reallocate the placement of facilities to adapt to the changes of urban planning, has remarkable impact on many areas. Existing solutions fail to guarantee the result quality on relocating $k > 1$ facilities. As k-FR problem is NP-complete and is not submodular or non-decreasing, traditional greedy algorithm cannot be directly applied. We propose to transform k-FR into another facility placement problem, which is submodular and non-decreasing. We prove that the optimal solutions of both problems are equivalent. Accordingly, we present the first approximate solution toward the k-FR, FR2FP. Our extensive comparison over both FR2FP and the state-of-the-art solution shows that FR2FP, although it provides approximation guarantee, cannot necessarily given superior results. The comparison motivates us to present an advanced approximate solution, FR2FP-ex. Moreover, based on Lagrangian relaxation, we develop an algorithm that can adjust the approximation ratio. Extensive experiments verified that, FR2FP-ex demonstrates the best result quality, and it is very close to the optimal solution. In addition, we also unveil the scenarios when the state-of-the-art would fail. We further generalize the k-FR problem, considering the budget for relocation and the cost of each facility. We also present corresponding approximate solutions toward the new problem and prove the approximation ratio.

CCS Concepts: • **Theory of computation** → **Facility location and clustering**; *Approximation algorithms analysis*; • **Information systems** → Wrappers (data mining);

Additional Key Words and Phrases: Facility Relocation, submodular, approximate algorithm

ACM Reference format:
Hu Wang, Hui Li, Meng Wang, and Jiangtao Cui. 2023. Toward Balancing the Efficiency and Effectiveness in k-Facility Relocation Problem. *ACM Trans. Intell. Syst. Technol.* 14, 3, Article 52 (April 2023), 24 pages.
https://doi.org/10.1145/3587039

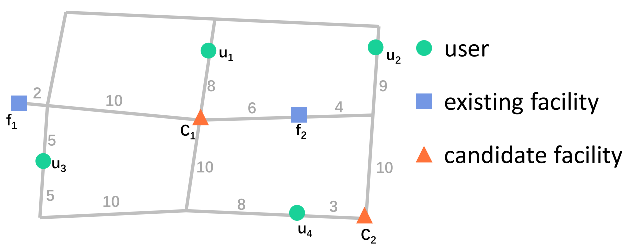
52

1 INTRODUCTION

Facility relocation (FR) problem aims to reallocate facilities in light of changes in users' locations. This can improve service quality and is useful in many applications. As an example, consider there

This work is supported by National Natural Science Foundation of China (Grants No. 61972309, No. 62272369, and No. 61976168), CCF-Huawei Database System Innovation Research Plan (Grant No. 2020010B), Key Scientific Research Program of Shaanxi Provincial Department of Education (Grant No. 20JY014), and Natural Science Basic Research Program of Shaanxi (Grant No. 2023-JC-YB-558).
Authors' addresses: H. Wang, H. Li, and J. Cui, Xidian University, Xi'an, China; emails: wh2mail@163.com,{hli, cuijt}@xidian.edu.cn; M. Wang, Xi'an Polytechnic University, Xi'an, China; email: wamengit@sina.com.

2157-6904/2023/04-ART52 $15.00
https://doi.org/10.1145/3587039



(a) Users, facilities & candidates

facility	cost
f_1	40
f_2	70
c_1	100
c_2	50

(b) Example of cost

Fig. 1. Example of facility relocation.

is a new subway line launched, many people may resettle to different locations accordingly. Due to that, various facilities (e.g., chain store, firehouse, etc.) have to be reallocated to provide better user experience (e.g., minimize the average distance to the nearest facilities).

FR problem has been extensively studied in literature [9, 13, 22, 23, 29]. Specifically, given a set of users U (e.g., $u_1, \ldots, u_4$ in Figure 1(a)), existing facility locations F (e.g., f_1, f_2 in Figure 1(a)), a set of new locations C (e.g., c_1, c_2 in Figure 1(a)) to move to, assume that each user is associated with a nearest facility in F (e.g., $u3$'s nearest facility is f_1), it is rational to expect that the distance between each user and her nearest facility is minimized. Driven by that, FR [22] aims to relocate k arbitrary facilities $f \in F$ with new locations $c \in C$ in order that the average distance between all users and their nearest facilities is minimized.

1.1 Motivation

Most of the existing solutions to the FR problem only consider the situation where we only relocate one facility [22, 27], i.e., $k = 1$. However, in practice, it is ubiquitous that multiple facilities are reallocated at the same time. For instance, consider population changes in 254 counties in Texas. During the period from 2010 to 2021,[1] there are 38 counties with a population change rate greater than 20% (including both increase and decrease). Among them, 19 counties have a change rate over 30%. Suppose there is a chain store (e.g., McDonald's) that deployed facilities according to the population distribution in 2010. By 2021, due to the change in population distribution, it will definitely be unable to provide the best service and thus fail to obtain the maximum benefits especially when their competitors have improved the service by relocating facilities. Therefore, there exist enough motivation for the store to relocate a series of the existing facilities to improve the service network (i.e., to minimize the serving distance).

1.2 Challenges and Intuition

Notably, given the existing solutions toward FR problem with $k = 1$, referred to as 1-FR, extensively addressing the problem with $k > 1$, referred to as k-FR, is not a trivial task due to the following challenges. First, the solution space changes from $|F| \times |C|$ to $\sum_{i=1}^{k} \binom{|F|}{i} \times \binom{|C|}{i}$, which is not polynomial. Note that the solution space does not necessarily become $\binom{|F|}{k} \times \binom{|C|}{k}$, but is eventually much larger. The reason is, although we hope to relocate k facilities, the average distance of users may be already minimized when $k'(k' < k)$ facilities are replaced. Actually, the k-FR problem is NP-complete, which is proved in Section 4 . Second, as we shall show in Section 4, k-FR is neither submodular nor non-decreasing (monotone), such that greedy approximate strategy cannot be directly applied. Although there exist heuristic solutions [27] toward the k-FR problem. They

[1] https://worldpopulationreview.com/us-counties/states/tx.

can not provide approximation ratio over the results, such that the effectiveness of the solutions cannot be guaranteed.

In addition to the hardness of the k-FR problem, we advocate that the users considered in FR problem should not be necessarily static but allowed to be dynamic. That is, when considering the serving distance from a facility toward a user, the user cannot be simply modeled as a static position, but a series of positions along her movement track. Similar problem statement can also be found in a series of related works [4, 27].

To address the limitations above, in this article, we propose to transform k-FR into a facility placement problem [4, 6, 25], which is submodular and non-decreasing, and prove that the transformed problem is equivalent to the k-FR. By converting the k-FR problem into the facility placement problem, we proposed the first approximate solution toward the k-FR problem, the results of which can provide an approximation guarantee to the optimal ones. By comparing with the state-of-the-art heuristic solution [27], we observe that the practical effectiveness of the basic approximate algorithm is not superior to the heuristic one. In fact, as k is small, the result quality of the basic solution is even poorer than that of Reference [27]. We exploited the reason for this phenomenon through plenty of theoretical and empirical studies, and accordingly proposed an advanced approximate solution, namely, FR2FP-ex, which can obtain the best performance over the competitors while ensuring the same approximation ratio.

In spite that FR2FP-ex can provide guaranteed sub-optimal results, its approximation ratio is fixed (but not adjustable) and may not meet the needs of all potential users. For this reason, inspired by Lagrangian relaxation and branch-and-bound strategy, we develop another algorithm that can answer k-FR problem with adjustable approximation ratio. On the one hand, as long as it takes enough time, this algorithm can give results at any required approximation ratio. On the other hand, the experimental results of this algorithm also justify that the results obtained by the FR2FP-ex is eventually almost the same to the optimal solution.

The classical definition of k-FR problem assume that the cost of relocating each facility is uniform. However, in practice, the cost for relocating a facility definitely varies depending on a series of factors, e.g., the distance between the new and old location, the rent at the new location, and so on. Driven by that, we extend the classical definition to a more general one as **general *k*-FR Problem** by allowing each reallocation be associate with a cost, and the overall reallocation cannot exceed a predefined budget. Accordingly, we extend both FR2FP and FR2FP-ex, and propose a pair of generalized version to solve this problem with guaranteed approximation ratio.

1.3 Contributions

Our technical contributions in this work can be summarized as follows:

- As the k-FR problem is not submodular or non-decreasing, it cannot be directly approximately addressed by hill-climb solutions. In light of that, we equivalently transform the k-FR into a facility placement problem and propose an approximate solution, FR2FP, accordingly. To the best of our knowledge, it is the first approximate solution toward the k-FR problem.
- We observe that, although the approximate solution provides result guarantee, it does not necessarily provide better results *practically*. Our thorough investigation, both theoretical and empirical, unveils the secret behind the unexpected performance of the existing heuristic solution in Reference [27].
- Given our insight in the unexpected phenomenon in the state-of-the-art heuristic solution [27], we propose an advanced approximate algorithm, FR2FP-ex, which empirically demonstrates superior performance compared to all existing solutions while ensuring the approximation ratio theoretically.

- We additionally develop an algorithm with adjustable approximation ratio. Experimental results show that FR2FP-ex can eventually generate a near-optimal result.
- We extend the k-FR problem to a general setting, where each reallocation is associated with a cost, and propose a pair of corresponding approximation algorithms toward it.

The rest of this article is organized as follows. We introduce the related work in Section 2. Section 3 formally gives the definition of the classical k-FR problem. In Section 4, three methods are proposed to solve k-FR problem with approximation ratio. In Section 5, we extend k-FR to a more general problem and give the corresponding solutions. Section 6 presents our experiments and results. In Section 7, we extensively discuss the scenarios where the state-of-the-art heuristic will fail in practice. In Section 8, we conclude this work.[2]

2 RELATED WORK

Min-dist in Euclidean. Reference [16] studied the k-medios problem based on the minimum distance. The goal is to select k center points from all points so that the summed distance between all other points and the nearest k centers is the smallest. Reference [36] studied another problem, by assuming that there are already some facilities and users, find a location in a given area to establish a new facility to minimize the average distance from the user to the nearest facility. The author proposed a progressive algorithm that can gradually obtain the optimal solution. Reference [21] studied a discrete form of the above problem. The possible location facilities are no longer the entire continuous space, but a set of discrete locations specified in advance. Their goal is to find a location from these locations to build a new facility that minimizes the average distance between users and the nearest facility. The above researches are all based on Euclidean space, which is different from the issue we consider on the road network.

Min-dist on the road. Reference [32] studied the optimal location problem on the road network. They proposed a framework based on the divide-and-conquer strategy to place multiple facilities at the same time. The problem studied by Reference [1] is the same as that of Reference [32]. Based on the idea of nearest local network [8], they proposed an efficient algorithm to solve the problem. As discussed in Reference [22], in real applications, we are always allowed to choose from some candidate locations. Hence the answers generated by these approaches may not eventually be valid in practice. Reference [20] finds a facility among existing ones that has the minimum average distance to all the users, w.r.t. Euclidean distance, while Reference [35] solved the problem on road network utilizing network connectivity information and spatial locality. In Reference [39], Voronoi diagram-based look-up tables were designed to avoid network traversal. Reference [33] presented a two-phase convex-hull-based pruning technique for both exact and approximate solutions. As these efforts only consider the facility placement but not reallocation, thus are orthogonal to our problem setting and cannot be applied to address k-FR.

Facility relocation. For the first time, Reference [29] considered the removal of old facilities in the facility location problem, which is the prototype of the FR problem, and proposed three approximate algorithms to solve the problem. But in this study, the distance between users and facilities is assumed to be known, which is not guaranteed in our problem. Reference [13] studied the re-planning of ambulances, which is essentially a special FR problem, which means that all facilities are relocated. They proposed a PAM-based method to solve this problem. Since it may not be necessary to relocate all facilities in k-FR, this method cannot solve the k-FR problem. Reference [9] studied the mobile facility location problem, in which users and facilities need to be relocated at the same time. However, only facilities can be relocated in k-FR, so this method is not applicable. Therefore, none of the above methods can be used to solve the k-FR problem.

[2] A shorter version of this work appeared in Reference [26].

Table 1. Notations

Notation	*Definition*
$G(V, E)$	Road network, V is the vertex set and E is the edge set
F	Set of existing locations
C	Set of candidate locations
F_{all}	$F \cup C$
$d(i, j)$	Distance between location i and j
$\Delta(\cdot)$	Reduction in total distance of a relocation pair
$\sigma(\cdot)$	Monotone and submodular function

Reference [22] considered an FR problem similar to our setting. Based on Replacement Influence Distance, which is used to restrict the search space, they solved this problem in Euclidean space. Reference [27] studied the FR problem on road network and they proposed two methods, depending on whether an index can be prepared beforehand, to solve the problem. In fact, although Reference [27] focuses on the 1-FR, the author extensively discussed how their solution to 1-FR can be extended to a heuristic solution to address the k-FR. However, they failed to provide any theoretical or empirical study for their effectiveness or efficiency in k-FR setting. In this work, we give a thorough study over their proposed strategy and unveil that (1) although failing to guarantee the result approximation ratio, the heuristic solution can be as good as, or even superior to, our basic approximation scheme; (2) the scenarios where the heuristic solution fails to provide acceptable results are also identified by our exhaustive study.

Submodular maximization. As the facility placement problem is essentially a variation of submodular maximization problem, we present the relevant research work. The fundamental submodular maximization problem, which aims to maximize a monotone submodular function under a general matroid constraint, has been well known since the late 1970s and the greedy algorithm achieves a 1/2 approximation [5]. Even for the special case of cardinality constraint, there is no algorithm that can obtain an approximation guarantee better than $1 - 1/e$ using polynomially many value queries [18]. Until the early 21st Century, when Reference [24] introduced a continuous greedy algorithm which approximately maximizes the multilinear extension of the submodular function and obtains the optimal $1 - 1/e$ approximation guarantee. Moreover, Reference [11] study the non-monotone submodular maximization problem and propose a constant-factor approximation algorithm. However, the k-FR problem is neither submodular nor monotone, thus none of these studies can be used directly and guarantee the approximation ratio.

3 PROBLEM STATEMENT

In this section, we first formally define the k-FR problem and then discuss the concept of locations of mobile users.

3.1 FR Problem Definition

To formally define the k-FR problem, we introduce some necessary terminologies. Table 1 lists a set of frequently used notations in this article. A location l in this article is a planar position on an edge in a given directed road network $G(V, E)$, with a geographical coordinate (i.e., latitude and longitude). Each directed edge in E between a pair of vertices in V is associated with a positive cost, i.e., travel distance or time, and so on. Given any two locations l_1 and l_2, the directed network distance from l_1 to l_2 is denoted by $d(l_1, l_2)$, which may not be equal to $d(l_2, l_1)$. In addition, we denote existing facilities as a set of locations $F = \{f_1, \ldots, f_{n1}\}$ and the candidates (locations) to deploy substitutes as another set $C = \{c_1, \ldots, c_{n2}\}$. For a user at a particular location ℓ, the nearest facilities with respect to F are denoted as $\phi_F(\ell)$, and the network distances from ℓ to $\phi_F(\ell)$ is denoted as $\psi_F(\ell) = dist(l, \phi_F(\ell))$. A FR pair that consists of k obsolete facilities $F_k \subseteq F$ and k candidates $C_k \subseteq C$ for substitution is denoted as $\langle F_k, C_k \rangle$. For a user located at ℓ, given the relocation pair as $\langle F_k, C_k \rangle$, the distance to his nearest facility will become $\psi_{F'}(\ell)$, where $F' = F \setminus F_k \cup C_k$.

Definition 1. Given a facility set F and an FR pair $\langle F_k, C_k \rangle$, **the change of total distance** between all users U and their respective nearest facilities is defined as

$$\Delta\left(\langle F_k, C_k \rangle\right) = \sum_{i=1}^{|U|} (\psi_F(loc(u_i)) - \psi_{F'}(loc(u_i))),$$

where $loc(u_i)$ denotes the location(s) of user u_i, and $F' = F \backslash F_k \cup C_k$.

Given the definition above, classical FR problem [15, 23] is defined as follows.

Definition 2. Given a directed road network G and a set of users U, the k **facility relocation (k-FR) problem** aims to find an FR pair $\langle F_{k'}, C_{k'} \rangle$ among a set of existing facilities F and a set of candidate locations C such that

$$\langle F_{k'}, C_{k'} \rangle_{OPT} = \arg\max \Delta\left(\langle F_{k'}, C_{k'} \rangle\right), \text{ s.t. } k' \leq k.$$

3.2 Locations of u_i

Notably, in Definition 1, we have to compute the distance between u_i and its nearest facility. Computation of the distance is trivial when each user is simplified as a single location. However, this setting is unrealistic with increasing mobility of end users and proliferation of mobile applications. In real life, moving objects are ubiquitous. To represent the movement of an object, there are two categories of approaches: discrete (e.g., check-in data [34]) and continuous (e.g., trajectory [38]). No matter which method is used, a moving object is modeled as a set of positions [28] (e.g., check-ins at POIs or sample points of a trajectory), and we have to get rids of the following issues, namely, noisy, passing-by and outlier. Noise [37] is caused by data or GPS errors. Passing-by points are the points that the user only passes through without performing any meaningful activities. The outlier points are those visited occasionally. These points can mislead us to find the correct result of FR.

In addition, moving users are intimately associated with two major behaviors, as reported by Reference [14], frequently appearing at some place and staying for a duration. Obviously, the points in these locations are more important. As a result, identifying these points will pave the way to effectively capture daily activity places for handling the k-FR problem. In this regard, inline with References [4, 27], we employ the kernel method [31] to identify these points, referred to as *reference locations*. The general procedure [4, 27] can be outlined as follows. First, we discretize the 2D space into grids and evaluate the density (of all locations for a user) for each grid. The top-5% grids with the highest density are selected. Adjacent grids are further aggregated into a group. In each group, the peak grid with the highest density is intuitively regarded as a reference location. Moreover, the density accumulation of each group is normalized and accounted as the probability that a user appears accordingly.

Following the above, each user is outlined as a series of reference locations, each of which is associated with a probability of occurrence. Given that each u_i is now referred to as a group of reference locations, it is a non-trivial task to compute the distance in Definition 1. In this regard, we shall propose a general version of Definition 1 such that each user is not necessarily simplified as a single position. For instance, in Figure 2, we extracted three reference locations and their

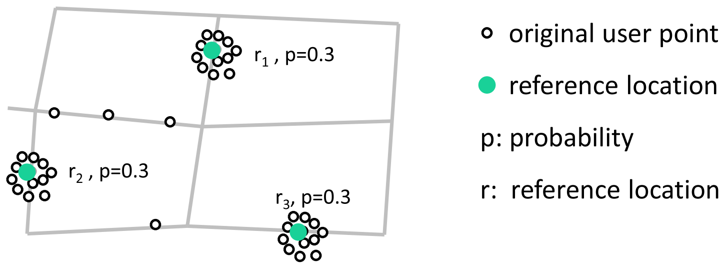


Fig. 2. Example of extracting reference location (virtual user).

probabilities from the user's historical movement logs. Considering each reference location as an independent *virtual user*, we can define the *change of total distance* for the mobile objects.

Definition 3. Given a facility set F, a FR pair $\langle F_k, C_k\rangle$ and a set of users U, each of whose movement can be modeled as a set of reference locations r, considered as a virtual user, and the corresponding probability p_r, **the change of total distance** between all virtual users R and their respective nearest facilities is defined as

$$\Delta\left(\langle F_k, C_k\rangle\right) = \sum_{i=1}^{|R|} \left(\psi_F(r_i) - \psi_{F'}(r_i)\right) \times p_{r_i}.$$

Next, we shall present our solutions toward the k-FR problem, where the $\Delta(\cdot)$ is defined upon Definition 3. In the following, all users represent virtual users, which are locations.

4 SOLUTIONS TO k-FR

Intuitively, it is straightforward to give a brute-force approach for k-FR as follows. Find all possible k'-FR pairs $\langle F_{k'}, C_{k'}\rangle$ $(1 \leq k' \leq k)$, and return the best one i.e., with the maximum $\Delta(\cdot)$. Unfortunately, the solution space is up to $\sum_{i=1}^{k} \binom{|F|}{i} \times \binom{|C|}{i}$. Moreover, as we will prove in Theorem 4, the k-FR problem is NP-complete. Therefore, one practical way to address it is to find a polynomial solution, either heuristic or approximated.

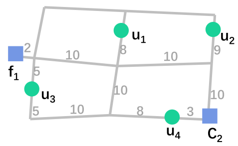
To this end, Reference [27] present a heuristic solution, referred to as kLNB, which greedily selects the best 1-FR pair[3] one-after-another.[4] As demonstrated by Lemma 1, the greedy strategy is an efficient algorithm for optimization problems that is submodular and non-decreasing.

Definition 4. Consider an arbitrary function $\sigma(\cdot)$ that maps subsets of a finite set P to non-negative real numbers. We say that σ is **submodular** if $\sigma(A \cup \{v\}) - \sigma(A) \geq \sigma(B \cup \{v\}) - \sigma(B)$, for all elements v and all pairs of sets $A \subseteq B \subseteq P$.

Lemma 1 ([19]). *For a non-decreasing, submodular function σ, a finite set P, an integer k and the problem $max_{P' \subseteq P}\{\sigma(P') : |P'| \leq k\}$, a greedy algorithm always produces a solution whose value is at least $1 - \frac{1}{e}$ times the optimal value.*

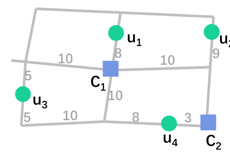
However, in k-FR problem, the function to be maximized (i.e., $\Delta(\cdot)$) is not submodular or non-decreasing, which is justified by Example 1. Therefore, the approximation ratio of the kLNB cannot be proved by Lemma 1. In fact, the kLNB fails to provide any approximation ratio, which we have theoretically studied and proved (shown in Section 7).

[3] A special case of FR pair $\langle F_k, C_k\rangle$, where $k = 1$.
[4] Reference [27] proposed a solution to the 1-FR problem.



Pair	distance reduction
$< f_1, c_1 >$	-2
$< f_1, c_2 >$	0
$< f_2, c_1 >$	-1
$< f_2, c_2 >$	2

(a) 1-st iteration



Pair	distance reduction
$< f_1, f_2 >$	-2
$< f_1, c_1 >$	4
$< c_2, c_1 >$	-3
$< c_2, f_2 >$	-2

(b) 2-nd iteration

Fig. 3. Facility distribution and marginal gain after different iterations in Example 1.

ALGORITHM 1: FR2FP Algorithm

Input: Road network $G(V, E)$; current facility locations F; candidate facility locations C; limit of relocation number k

Output: Facility relocation (FR) pair $\langle F_k, C_k \rangle$

Let $F_{all} = F \cup C$
The result set $S = \emptyset$
The number of locations in S that belong to C: $num = 0$
for i in 1 to $|F|$ **do**
  Calculate the total distance $D(l)$ for each location in F_{all} if it is selected
  select the location l_i whose $D(l)$ is minimum
  $S = S \cup l_i$
  if $l_i \in C$ **then**
    $num = num + 1$
    if $num == k$ **then**
      $F_{all} = F_{all} \setminus C$
    end if
  else
    $F_{all} = F_{all} \setminus \{l_i\}$
  end if
end for
return $\langle F \setminus S, C \cap S \rangle$

Example 1. In Figure 1(a) there are four virtual users $\{u_1, u_2, u_3, u_4\}$, two current facilities $\{f_1, f_2\}$ and two candidate locations $\{c_1, c_2\}$. If we want to relocate two facilities, then kLNB algorithm would find the 1-FR pair $\langle f, c \rangle$ that brings the greatest distance reduction. In the first iteration, all pairs and corresponding distance reductions are shown in Figure 3(a). Obviously, the kLNB will select $\langle f_2, c_2 \rangle$, whose distance reduction is 2. Then for the second iteration, pairs and reductions are shown in Figure 3(b). $\langle f_1, c_1 \rangle$ will be selected. Obviously, the marginal gain in the second iteration (i.e., 4) is larger than the first (i.e., 2). Moreover, the benefit of $\langle f_1, c_1 \rangle$ is better when the current solution is $\langle f_2, c_2 \rangle$ (i.e., 4) than when it is empty (i.e., 0).

To the best of our knowledge, there is no approximate algorithm toward k-FR, though kLNB suggests a heuristic solution. In the following, we shall present three algorithms that provide approximate results for the first time.

4.1 Basic Approximate Solution

Before presenting our first approximate solution, we introduce the facility placement problem.

Definition 5. Given a directed road network $G = (V, E)$, a set of existing facility locations F, a set of candidate facility locations C, and a set of virtual users U, suppose $D_U(F)$ denote the aggregate distance from each virtual user in U to the nearest facility in F, then the goal of **the facility placement problem** is to find $C' \subseteq C$ ($|C'| = k$), so that the total distance reduction, denoted as $\omega_{U,F}(F \cup C') = D_U(F) - D_U(F \cup C')$, is maximized.[5]

Theorem 1. *The facility placement problem in Definition 5 is NP-complete.*

Proof. First, we need to introduce the node cover problem. Given a graph G and an integer k, does there exist a subset of k nodes of G that cover all the edges of G? (Node v is said to cover edge e if v is an endpoint of e.) The node cover problem is NP-complete [3, 10].

We now show that the node cover problem is reduciable to the facility placement problem. Without loss of generality, consider a graph $G(V, E)$. We can construct an instance of the facility placement problem with the candidate facility set $C = V$ and set of users $U = E$. Assuming each user has a nearest facility, which is a dummy node, and the existing facility set $F = \{dummy\ nodes\}$, let the distance between each user and the corresponding dummy node be $|E| + 1$. In addition, let $d(v_i, e_j) = 1$ if $v_i \in V$ is an endpoint of $e_j \in E$, otherwise $d(v_i, e_j)$ is set to $+\infty$. This transformation is polynomial in the size of the graph. If there exists a k-covering, then the total distance would be less than or equal to $|E|$. So we just need to check if the improvement is more than or equal to $(|E|+1)|E| - |E| = |E|^2$. As long as there is one user not covered, the best choice is the dummy node, with cost |E| + 1. Therefore, the improvement is less than $(|E|+1)|E| - (|E|+1) = |E|^2 - 1$. According to the optimal solution of facility placement problem, we can immediately get the answer to the original node cover problem. Therefore, the facility placement problem is NP-complete. □

Theorem 2. *$\omega(\cdot)$ in Definition 5 is a non-decreasing and submodular function.*

Proof. First, $\forall A \subseteq C, \omega(A) \geq 0$. This is obvious, because when a new facility is added, the distance from the user to the nearest facility may only decrease or remain the same. Second, $\forall A \subseteq C, \forall l \in C \setminus A, \omega(A \cup \{l\}) \geq \omega(A)$. The reason is the same as the above. Thus, the function $\omega(\cdot)$ is non-decreasing. Last, suppose $A \subseteq B \subseteq C, \omega(A) \geq 0, \omega(B) \geq 0$. Then for $\forall l, l \in C \setminus B$, we can know that $\omega(A \cup \{l\}) - \omega(A) \geq \omega(B \cup \{l\}) - \omega(B)$. Thus the function is submodular. □

As discussed above, k-FR is not submodular or non-decreasing, thus presenting an approximate solution is extremely challenging. But the facility placement problem is submodular and non-decreasing. Therefore, is it possible for us to transform the k-FR into a facility placement problem such that both properties can be realized? Motivated by that, we propose to consider the k-FR problem in the perspective of *facility placement* as follows.

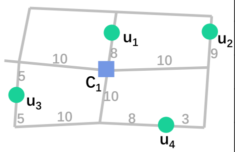
Definition 6. Given road network $G = (V, E)$, current facility set F, candidate facility set C, and set of virtual users U, $\emptyset$ is an empty facility set and $\omega(\cdot)$ is the total distance reduction in Definition 5, then the ***k*-FR problem from facility placement perspective** aims to find a set of locations H from $F \cup C$ so that

$$Maximize \quad \omega_{U,\emptyset}(H) \ \text{ s.t. } |H| = |F|, |H \cap C| \leq k. \tag{1}$$

Theorem 3. *Let H_{OPT} be the optimal solution to problem[6] 6, then it can be classified into two subsets, namely, $H^F_{OPT} = H_{OPT} \cap F$ and $H^C_{OPT} = H_{OPT} \cap C$, where $|H^C_{OPT}| \leq k$. Then $\langle F \setminus H^F_{OPT}, H^C_{OPT} \rangle$ is the optimal solution to problem 2. That is, the optimal solution of problem 2 can be directly acquired from the optimal solution of problem 6 [26].*

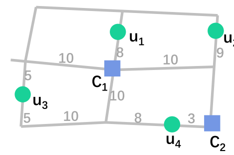
Corollary 1. *Suppose H_{app} is a solution of problem 6 with approximation ratio p, i.e., $\omega(H_{app}) \geq p \cdot \omega(H_{OPT})$, then $\langle F \setminus H^F_{app}, H^C_{app} \rangle$ is a p-approximate solution to problem 2 [26].*

[5]As U, F are fixed in the problem, we shall use $\omega(\cdot)$, $D(\cdot)$ for short in the sequel.
[6]We shall use problem x, for short, to refer to the problem defined in Definition x.



facility	total distance
f_1	88
f_2	65
c_1	**60**
c_2	76

(a) 1-st iteration



facility	total distance
f_1	52
f_2	53
c_2	**45**

(b) 2-nd iteration

Fig. 4. Facility distribution and total distances after each iteration of FR2FP.

THEOREM 4. *The k-FR problem is NP-complete.*

PROOF. According to Definition 6, the k-FR problem can be transformed into facility placement problem, which is proven to be NP-complete in Theorem 1. Therefore, the k-FR problem is NP-complete. □

As problem 6 is a facility placement problem. It is both submodular and non-decreasing, such that a greedy algorithm can guarantee the approximation ratio in the results. Accordingly, we are now ready to propose a basic approximation strategy toward both problems, namely, FR2FP (Facility Relocation to Facility Placement), showed in Algorithm 1. The algorithm begins by **assuming that there is no existing facility** and computes the $D(l)$ for each location $l \in F_{all}$ (Line 5), which represents the total distance between all virtual users and his nearest facility after joining the location l. Every time it will select the location l with the minimum $D(l)$ until $|F|$ locations (Line 6). Each time a location $c \in C$ is selected, it will increase the count of locations selected from C. Whenever the number is up to k, it removes all locations in C from F_{all} (Lines 8–12).

Example 2. In Figure 1(a), FR2FP assumes that there is no current facility, and then gradually selects the facility that can bring the minimum value in $D(\cdot)$. In the first iteration, the total distance after adding each facility is shown in Figure 4(a). Because the distance after adding c_1 is the smallest, it will be selected as the first facility. Since c_1 belongs to C, we need to judge whether the number of locations selected from C (1 at the current timestamp) has reached k (i.e., 2 in the example). If not, then we carry on to select the next facility. According to Figure 4(b), c_2 is the best choice with the minimum total distance 45 in the second iteration. The total number of locations selected now reaches $k = 2$. Thus, FR2FP will stop with the final facility set $\{c_1, c_2\}$. The k-FR pairs selected by FR2FP is $\langle F_2, C_2 \rangle$ ($F_2 = \{f_1, f_2\}, C_2 = \{c_1, c_2\}$).

Definition 7. Let P be a finite set and $\mathcal{I}$ be a nonempty collection of subsets of P. The system $\mathcal{M} = (P, \mathcal{I})$ is called a **matroid** if : (a) $\forall P_1 \in \mathcal{I}$ and $P_2 \subset P_1 \Rightarrow P_2 \in \mathcal{I}$; and (b) $\forall P' \subset P$ every maximal member of $\mathcal{I}(P') = \{I : I \in \mathcal{I}, I \subseteq P'\}$ has the same cardinality.

THEOREM 5. *According to Reference [5], for a non-decreasing, submodular function σ, a finite set P and the problem $\max_{P' \subseteq P}\{\sigma(P') : P' \in \mathcal{I}\}$ where $(P, \mathcal{I})$ is described by the intersection of N matroids. The greedy algorithm always produces a solution whose value is at least $\frac{1}{N+1}$ times the optimal value.*

THEOREM 6. *For the facility relocation problem in Definition 6, the FR2FP algorithm showed in Algorithm 1 can achieve a $\frac{1}{3}$ approximation ratio.*

In Definition 6, there are two matroid constraints in the problem, which are $|H| = |F|$ and $|H \cap C| \leq k$. According to Theorem 5, it is easy to prove that the approximation ratio of the Algorithm 1 is $\frac{1}{3}$.

ALGORITHM 2: FR2FP-ex Algorithm

Input: Road network $G(V, E)$; current facility locations F; candidate facility locations C; limit of relocation number k;
Output: Facility relocation (FR) pair $\langle F_k, C_k \rangle$
1: FR pair $\langle F_k, C_k \rangle$ = FR2FP(G, F, C, k)
2: Let $S = F \cup C_k \setminus F_k$ *//final facility locations selected by FR2FP*
3: Let $L = C \cup F_k \setminus C_k$ *//all locations not selected by FR2FP*
4: **while** true **do**
5: Select the interchange pair $< l, s >$ $(l \in L, s \in S)$ which is the same type and brings the maximum distance reduction
6: **if** reduction ≤ 0 **then**
7: break
8: **else**
9: $S = S \cup \{l\} \setminus \{s\}$
10: $L = L \cup \{s\} \setminus \{l\}$
11: **end if**
12: **end while**
13: return $\langle F \setminus S, S \cap C \rangle$

4.2 Advanced Approximate Solution

Although the approximation ratio of the FR2FP algorithm is $\frac{1}{3}$, it sometimes performs poorly in practice. To improve the performance of the algorithm, we propose a variation of FR2FP, namely, FR2FP-ex (FR2FP-with-exchange). The idea is to add an exchange after getting the results from FR2FP. The pseudo code is shown in Algorithm 2. Assuming that all facilities are classified into two groups according to whether they originally belonged to F or C, then the facilities in S also belong to both groups accordingly. Let L be all the facilities except S, which also belong to either F or C. The exchange is performed iteratively. Each time it selects the exchange pair $\langle l, s \rangle$ that brings the greatest distance reduction (Line 5). Both facilities in the pair must belong to the same group (both in F or C), which ensures the number of facilities in C will not exceed k. Whenever the current exchange pair does not cause the total distance to decrease, the algorithm stops (Lines 6 and 7).

The idea of the exchange is not complicated, but shows significant improvement compared with the FR2FP, especially when k is small. Moreover, as the exchange only happens when FR2FP is inferior, the results FR2FP-ex obtained is at least as good as that of the FR2FP. Therefore, FR2FP-ex still guarantees the approximation ratio.

4.3 Guarantee Tunable Solution

Notably, the approximation ratio of FR2FP-ex is fixed at $\frac{1}{3}$, which may be not good enough in a few scenarios. Some researchers have developed solution toward the k-median problem and the facility location problem based on Lagrangian Relaxation [2, 17, 29] (LR for abbreviation), which suggests a potential direction for addressing the k-FR. Driven by that, we propose another algorithm based on LR [7] that can adjust the approximation ratio if given enough running time.

4.3.1 Formulation. We first formulate the k-FR problem as an integer programming problem. It can be viewed as finding the locations from the existing facilities to close and the locations for new facilities to open so that the total travel distance between virtual users and their respective nearest facilities can be minimized. Given this view, we can separate the task into two parts: (1) selecting locations from $F \cup C$, each of which locates a facility (if a location in F is not selected, then the existing facility at the location should be closed; if a location in C is selected, then a new facility should be opened there), and (2) assigning virtual users in U to the closest *opening facility*. Define

$$y_j = \begin{cases} 1 \text{ if there is an opening facility at location } j \in F \cup C \\ 0 \text{ otherwise,} \end{cases}$$

and

$$x_{ij} = \begin{cases} 1 \text{ if virtual user } i \text{ is served by facility opened at } j \\ 0 \text{ otherwise.} \end{cases}$$

If $y_j = 0$ for $j \in F$, then it means that the existing facility at $j \in F$ is needed to be closed; if $y_j = 1$ for $j \in C$, then there will be a new facility to be opened at potential location j. Since the maximal number of relocation is k and $|F|$ facilities are desired, we have $\sum_{j \in C} y_j \leq k$ and $\sum y_j = |F|$. As a virtual user can only be assigned to exactly an opening facility, we have $x_{ij} \leq y_j$ ($\forall i \in U \wedge j \in F \cup C$) and $\sum_{j \in F \cup C} x_{ij} = 1$ ($\forall i \in U$). Therefore, the problem can be formulated as follows:

$$(P) \text{ Minimize} \quad \sum_{i \in U} \sum_{j \in F_{all}} d(i,j) x_{ij}, \;\; F_{all} = F \cup C,$$

subject to

$$\sum_{j \in C} y_j \leq k, \tag{2}$$

$$\sum_{j \in F_{all}} x_{ij} = 1 \quad \forall i \in U, \tag{3}$$

$$\sum_{j \in F_{all}} y_j = |F|, \tag{4}$$

$$x_{ij} \leq y_j \quad \forall i \in U, \; j \in F_{all}, \tag{5}$$

$$y_j \in \{0,1\} \quad \forall j \in F_{all}, \tag{6}$$

$$x_{ij} \in \{0,1\} \quad \forall i \in U, j \in F_{all}. \tag{7}$$

4.3.2 Algorithm Design. To solve this integer programming problem, we first construct the Lagrangian relaxation of (P) with respect to constraint Equations (2) and (3) for given Lagrangian multipliers (w, v), which is $L(w, v)$. It can be formed as

$$L(w,v) = \min_{x,y} \sum_{i \in U} \sum_{j \in F_{all}} d(i,j) x_{ij} + \sum_{i \in U} w_i \left(1 - \sum_{j \in F_{all}} x_{ij} \right) + v \left(\sum_{j \in C} y_j - k \right) \text{ subject to constraint Equations (4)–(7).}$$

The objective function in the relaxation $L(w, v)$ can be rewritten as

$$\sum_{i \in U} \sum_{j \in F_{all}} \left(d(i,j) - w_i \right) x_{ij} + v \sum_{j \in C} y_j + \sum_{i \in U} w_i - vk. \tag{8}$$

Define $(d(i,j) - w_i)^- = min\{0, \; d(i,j) - w_i\}$, and let

$$\rho_j(w,v) = \begin{cases} \sum_{i \in U} (d(i,j) - w_i)^- & \text{if } j \in F, \\ \sum_{i \in U} (d(i,j) - w_i)^- + v & \text{if } j \in C. \end{cases}$$

Suppose we rank all $\rho_j(w, v)$ according to their values in ascending order and denote the top-$|F|$ minimum values as $\rho_{j_1}, \rho_{j_2}, \ldots, \rho_{j_{|F|}}$ accordingly, then the corresponding location set is $\{j_1, j_2, \ldots, j_{|F|}\}$. Since the last two terms in Equation (8) are constant, independent of x and y, it follows by inspection that the optimal solution (Minimal) to the LR for a given w and v is

$$y_j = \begin{cases} 1 & \text{if } j \in \{j_1, j_2, \ldots, j_{|F|}\} \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

$$x_{ij} = \begin{cases} y_j & \text{if } d(i,j) - w_i < 0 \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

As stated in Reference [7] the result value after Lagrangian relaxation is less than or equal to that of the original problem. The Lagrangian dual of (P) with respect to constraint Equations (2) and (3) is maximizing $L(w, v)$, subject to $v \geq 0$. This dual can be solved via the following subgradient algorithm.

Subgradient algorithm

For the problem after Lagrangian relaxation, we can use the subgradient algorithm to update the Lagrangian multipliers (w and v) iteratively to gradually approach the maximal solution. We will briefly introduce it in this part.

Let Q be the incumbent solution set (i.e., the best set of locations for opening facilities found so far, such as the result of kLNB or FR2FP), γ be the objective value of Q (i.e., γ is the best upper bound obtained so far for the primal problem (P)), and β be the best objective value found for the Lagrangian dual, the main idea of subgradient algorithm is

Step 1. Initialize by setting $\beta = -\infty$, $iter = 1$, $v^1 = 0$ and $w_i^1 = min\{d(i,j) : j \in Q\}, i \in U$.

Step 2. Find an optimal solution $(y_j^{iter}, x_{ij}^{iter})$, $\forall i \in U, j \in F_{all}$ of the Lagrangian relaxation $L(w^{iter}, v^{iter})$ by using Equations (9) and (10). Calculate the partial derivatives and let

$$\theta_i = 1 - \sum_{j \in F_{all}} x_{ij} \quad \forall i \in U \qquad \text{and} \qquad \eta = \sum_{j \in C} y_j - k.$$

When $\theta_i = 0\ (\forall i \in U), \eta \leq 0$ and $v^{iter} = 0$, the procedure stops and $(y_j^{iter}, x_{ij}^{iter})$ solves (P).

Step 3. Update the value β, if $(y_j^{iter}, x_{ij}^{iter})$ provides a better objective function value for the dual.

Step 4. Define a step size $\lambda_{iter} = (\gamma - \beta)/(\|\theta\| t^{iter})$, where t^{iter} is a factor related to iteration $iter$. Use the step size to update the Lagrangian multipliers. For $\forall i \in U$, the $iter + 1$th value of w_i is $w_i + \theta_i \lambda_{iter}$. The updated value of v is $max\{0, v + \eta\lambda_{iter}\}$. Increase $iter$ by 1 and if $iter$ is greater than the number of iterations permitted, stop; otherwise, go to step 2.

The following LR approximation scheme uses this subgradient algorithm and branch and bound strategy to obtain an ϵ-optimal solution to problem (P).

Definition 8. Let z be the objective function value of a solution x and z^* be the theoretical optimal objective function value, then x is an ϵ-optimal solution if $(z - z^*)/z^* \leq \epsilon$.

LR approximation algorithm

Next, we introduce how to obtain approximate solutions using the branch and bound strategy. We branch by deciding whether to choose a certain location j (y_j is 1 or 0).

Step 1. Set the Q,[7] which is a solution to the primal problem (P), and γ is the objective value of Q, which is the upper bound.

Step 2. Solve the problem via a branch and bound technique. The lower bound on each branch is obtained by solving the Lagrangian dual using the *subgradient algorithm* presented above. Branch on the facility y_j with the most virtual users assigned to it. A branch with lower bound no less than $\gamma/(1+\epsilon)$ will be pruned from the branch and bound tree, where γ is the best objective function value of (P) found so far.

The reason for branching on the facility with the most virtual users is that the facility stands a good chance to be within the optimal solution (in which case its y value will be set to 1). As a result, we hope to reduce the size of branch and bound tree by fixing such variables at an early stage.

Theoretically, it is possible to obtain results with any approximation ratio using this algorithm, though this algorithm cannot guarantee a solution in *polynomial time*. Aside from the tunable approximation ratio, another advance for this solution is that it can generate *the optimal* results if given plenty running time. The empirical results justify that the effectiveness of the FR2FP-ex (respectively, FR2FP) algorithm is very close to *the optimal*, which will be described in detail in Section 6.

4.4 Cost Analysis

In this section, we study the time complexity of the proposed methods: FR2FP, FR2FP-ex, and LR.

The complexity of calculating the distance between a user and facilities using Dijkstra algorithm is $O(|E| + |F_{all}|log|F_{all}|)$. We need calculate the distance between each user and facilities before FR2FP selects plans. So the complexity is $O\ (|U| \cdot (|E| + |F_{all}|log|F_{all}|))$. FR2FP uses the fast greedy algorithm proposed in Reference [30] to select the plans, and its time complexity is $O(|F| \cdot |U'| \cdot |F_{all}|)$ where U' represents the users involved in each iteration. As $|U'| \ll |U|$, the process of choosing plans is much faster than calculating the distance.

Compared with FR2FP, FR2FP-ex requires an additional exchange process, and we use the LNB method proposed in Reference [27] to achieve it. Because we can initialize the required data structures when calculating the distance. We only need to consider the time it takes for exchanging, which is $O(E')$. E' represents the edge actually involved in the algorithm and it can be shown in Reference [27] that $E' \ll E$. Therefore, the cost of exchanging is very small compared to the previous steps.

LR is not a polynomial time algorithm. Its execution time is closely related to the initial solution and approximation ratio. The worst time complexity will reach $O(2^{|F_{all}|})$.

5 EXTENSIONS

In this part, we discuss an extension to the k-FR problem for more general scenario. Two algorithms are presented to address this general problem. The theoretical guarantees for the result quality are also provided for both approaches.

5.1 Budgeted k-FR Problem

In the k-FR problem in Definition 2, we are free to replace existing facilities with candidate ones. The marginal cost for each replacement (i.e., replace a facility at i with a new one at j) is assumed to be uniform. However, in practice, this assumption may not be true, e.g., the rent change for each facility replacement is not necessarily the same. In addition, the budget available for relocation is

[7]It can be the result of FR2FP or FR2FP-ex or even F. Different initial values will only affect the running time of the algorithm, not the approximation ratio.

ALGORITHM 3: FR2FP-B Algorithm

Input: Road network $G(V, E)$; current facility locations F; candidate facility locations C; limit of relocation number k; budget B
Output: Facility relocation (FR) pair $\langle F_k, C_k \rangle$
1: Let $F_{all} = F \cup C$
2: The result set $S = \emptyset$
3: The number of locations in S that belong to C: $num = 0$
4: Total cost for relocation $COST = 0$
5: **for** i in 1 to $|F|$ **do**
6:     Calculate the total distance $D(l)$
7:     Calculate relocation cost $cost_l$ for each location in F_{all} if it is selected
8:     select the location l_i whose $D(l)$ is minimum and $COST + cost_l \leq B$
9:     $S = S \cup \{l_i\}$
10:     $COST = COST + cost_{l_i}$
11:     **if** $l_i \in C$ **then**
12:         $num = num + 1$
13:         **if** $num == k$ **then**
14:             $F_{all} = F_{all} \setminus C$
15:         **end if**
16:     **else**
17:         $F_{all} = F_{all} \setminus \{l_i\}$
18:     **end if**
19: **end for**
20: return $\langle F \setminus S, C \cap S \rangle$

usually limited. Unfortunately, solutions presented above cannot work in this scenario. For example, in Figure 1(b), we still use the data from Figure 1(a), but adding the cost of opening or closing each facility. Every facility we change has a corresponding cost, and we only have a limited budget. We called this problem the budgeted k-FR problem and it can be formalized as follows.

Definition 9. Given a directed road network G, a set of virtual users U, a set F of existing facilities $\{f_1, f_2, \ldots\}$ and the corresponding closing costs $\{c_{f_1}, c_{f_2}, \ldots\}$, a set C of candidate locations $\{c_1, c_2, \ldots\}$ and the corresponding opening costs $\{c_{c_1}, c_{c_2}, \ldots\}$ and a budget limit B. The **Budgeted k-FR Problem** aims to find pair $\langle F_{k'}, C_{k'} \rangle$ among F and C subject to B such that

$$\langle F_{k'}, C_{k'} \rangle_{OPT} = \arg\max \ \Delta \left(\langle F_{k'}, C_{k'} \rangle \right) . (k' \leq k).$$

5.2 Algorithm Design

Although FR2FP algorithm does not consider the diverse costs, we can attach a series of modifications accordingly to deal with the new problem while guaranteeing the corresponding approximation ratio, which is shown in Algorithm 3.

Before proving the approximation ratio, we explain some details of the algorithm, especially the difference from FR2FP. In Line 7, we need to calculate the cost $cost_l$ of each relocation. This is different from the cost c_f, i.e., for facilities in F to close, or cost c_c, i.e., for facilities in C to open. If we select the location in F, then we did not actually relocate any facilities. That means the relocation cost $cost_l$ is 0. Conversely, if we select the location c_x in C, then we must open a new facility at c_x and close a facility in F at the same time. However, at this step, we are not sure which facility should be closed. Hence, we assume the cheapest facility (f_y) remaining in F is closed to minimize the total cost. Therefore, the relocation cost $cost_l$ is $c_{c_x} + c_{f_y}$.

Example 3. A company intends to relocate up to two facilities in Figure 1, and the budget is only 200. The total distance can refer to Figure 4. In the first iteration, c_1 is best and the cost to open a new facility in c_1 is 100. Every time a new facility is built in C, a facility in F must be closed. We assume the cheapest facility remaining in F to be closed, which is f_1. So the relocation cost is $100 + 40 = 140$. In the second iteration, c_2 is the best choice. But the remaining budget is only 60, which is not enough. Therefore, we can only choose the suboptimal solution f_1. In the previous step, f_1 has been removed by our assumption; if we choose f_1 now, then another facility must be removed. We still choose the cheapest facility remaining in F and the answer is f_2. The relocation cost in this iteration is $70 - 40 = 30$. Overall, we replace f_2 with c_1, and the total cost is $100 + 70 = 170$.

For the problem in Definition 9, if we transform it into a facility placement problem like Equation (1), then it has the form

$$Maximize \quad \omega_{U,\emptyset}(H) \text{ s.t. } |H| = |F|, |H \cap C| \leq k, Cost(H) \leq B,$$

where $Cost(H)$ represents the total cost used for relocation. Because $Cost(H) \leq B$ is not a matroid constraint, we cannot use Theorem 5 directly. Below, we must use other methods to prove the approximation ratio.

THEOREM 7. *For the budgeted k-FR problem defined in Definition 9, given the distance reductions of the optimal solution and the solution found by Algorithm 3 D^* and D_g, respectively, and we assume that the facilities selected by Algorithm 3 in the first h times are the same as those of Algorithm 1, then*

$$D_g \geq \frac{h}{3 \cdot |F|} D^*.$$

PROOF. Let the distance reduction of the optimal solution of problem 2 and the solution selected by Algorithm 1 be D and D_1, respectively. It is obvious that $D \geq D^*$. For a submodular maximization problem, greedy algorithm has diminishing marginal return at each iteration [26]. Because we assume that the facilities selected by Algorithm 3 in the first h times are the same as those of Algorithm 1, then $D_g \geq \frac{h}{|F|} D_1$. According to Theorem 6, we know that $D_1 \geq \frac{1}{3} D$. Thus, we have $D_g \geq \frac{h}{3 \cdot |F|} D^*$. □

We can get Algorithm 4 by adjusting FR2FP-ex. In Line 6, if the locations of interchange pair $< l, s >$ are both selected from C, then the interchange cost can be calculated as $cost = c_l - c_s$. If the locations of interchange pair $< l, s >$ are both selected from F, then the interchange cost will be evaluated as $cost = c_s - c_l$. Because Algorithm 4 is based on Algorithm 3, it can obtain a solution in which the quality is at least as good as Algorithm 3.

6 EXPERIMENTS

In this section, we empirically evaluate the performance of the proposed approximate solutions. We firstly introduce the experiments of the k-FR problem, and then the experiments for the budgeted k-FR problem.

6.1 k-FR Problem

6.1.1 Experiment Setup.

Datasets. Two real-world datasets, **California (CA)** [12] and **Beijing (BJ)**, are adopted in our experiment. Both of them are get from the author of Reference [27]. There are 21,693 bidirectional edges and 21,047 vertices in CA. BJ consists of 433,391 unidirectional edges and 171,504 vertices. The type and number of facilities in each dataset are shown in Table 2. The number and distribution

Table 2. Facility Set of CA and BJ

CA	$\|F\|$	BJ	$\|F\|$
Hospital	1,000	Bank	1,000
Park	1,000	Cafe	1,000
Post Office	1,000	Logistic	1,000
School	1,000	Gas Station	1,000

ALGORITHM 4: FR2FP-ex-B Algorithm

Input: Road network $G(V, E)$; current facility locations F; candidate facility locations C; limit of relocation number k; budget B

Output: Facility relocation (FR) pair $\langle F_k, C_k \rangle$

FR pair $\langle F_k, C_k \rangle$ = FR2FP-B(G, F, C, k, B)
Let $S = F \cup C_k \setminus F_k$ *//final facility locations selected by FR2FP-B*
Let $L = C \cup F_k \setminus C_k$ *//all locations not selected by FR2FP-B*
Let $COST = COST_{FR2FP-B}$ *//cost used by FR2FP-B*
while true **do**
  Select the interchange pair $<l, s>$ $(l \in L, s \in S)$ of the same type, which brings the maximum distance reduction and interchange cost $cost$ no more than $B - COST$
  if reduction ≤ 0 **then**
    break
  else
    $S = S \cup \{l\} \setminus \{s\}$
    $L = L \cup \{s\} \setminus \{l\}$
    $COST = COST + cost$
  end if
end while
return $\langle F \setminus S, S \cap C \rangle$

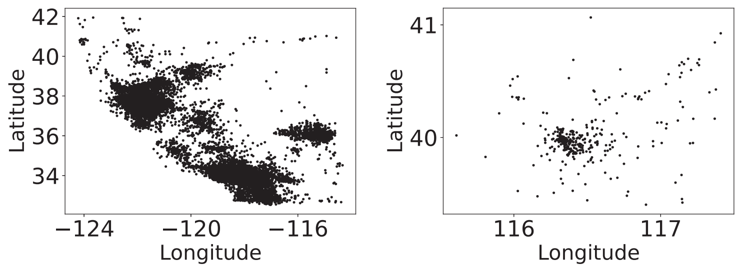
of each type of facility are different. To verify the influence of facility distribution, we uniformly sample 1,000 points of facilities for the experiment. The users in BJ dataset is available from Reference [38]. Each user has 136,686 sample points on average. For CA, we adopt a real-world check-in data.[8] We utilize the kernel method to extract the virtual users from the row data, and then conduct the experiments. The virtual users are all clustered and distributed in a certain area as shown in Figure 5. Candidate facilities are constructed in two ways[9]: (1) uniformly randomly generated geographically; (2) constructed according to the distribution of users. The former is used by default.

In addition, we also conduct experiments on two other synthetic graphs. Both of them have 50,000 vertices, 100,000 undirected edges, 1,000 virtual users, 100 existing facilities, and 100 candidate facilities. On one of the graphs, users and facilities are clustered and distributed in only a few areas. On the other graph, users and facilities are uniformly distributed.

Compared solutions. We report the performance of the following algorithms, all of which are implemented in C++ and tested on a 3.2 GHz quad-core machine with 16G RAM.

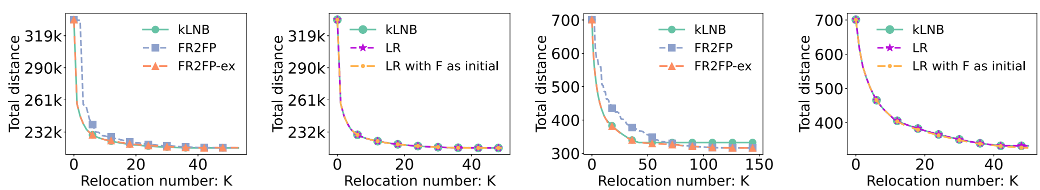
- kLNB: The heuristic solution of Reference [27], which iteratively select the optimal 1-FR pair for k times.
- FR2FP: The approximate solution proposed in Algorithm 1.
- FR2FP-ex: The advanced solution proposed in Algorithm 2.

[8]Obtained from http://snap.stanford.edu/data/.
[9]For the objects that are not exactly located on roads, we shift them to the closest point of on the road network.



(a) CA (b) BJ

Fig. 5. User distribution in the two datasets.



(a) CA:100 park (b) CA:100 park (c) BJ:1000 bank (d) BJ:1000 bank

Fig. 6. Effectiveness comparison.

- LR: The Lagrangian approximated algorithm, with $\epsilon = 0.1$. We use the solution of kLNB or the existing facility set F as its initial upper bound. The former is the default.

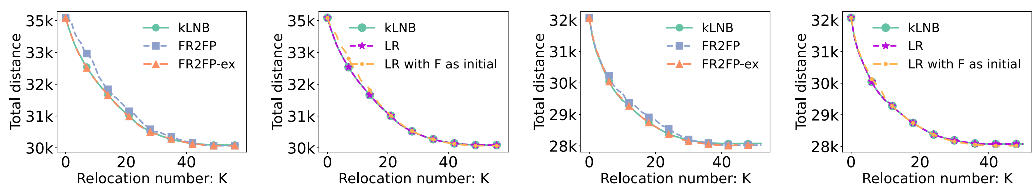
6.1.2 Experimental Results.

Comparison of effectiveness. First, we vary the size of F from 100 to 1,000 to test the performance of the four approaches. The results are similar, so for each dataset we only select one type of facility. For clarity, we separate the LR results and the resuls are shown in Figure 6. The x-axis and y-axis, respectively, represent k and the aggregated distance from the virtual user to the nearest facility. There are two points that need careful analysis.

First, from Figures 6(a) and 6(c), we can see that when k is small, the performance of FR2FP is the worst, even inferior to kLNB. The reason is that although the kLNB does not provide any guarantee, if only one facility is relocated, then this algorithm can generate the optimal solution. However, the approximation ratio of FR2FP is fixed at $\frac{1}{3}$. Therefore, when k is small, it is not as good as kLNB.

Second, from Figures 6(b) and 6(d), we can find that the results of LR are completely identical with the that of kLNB, even with F as its initial upper bound. This result shows that the solution obtained by kLNB has satisfied the approximation rate of $\epsilon = 0.1$ (according to Definition 8). Combining Figures 6(a) and 6(c), we can know that all of kLNB, FR2FP, and FR2FP-ex achieve near-optimal results eventually.

In addition to the above experiments, we have tested the effects of other factors on the effectiveness, including user distribution, facility distribution, and candidate distribution. In all these tests, all the compared solutions behave similar to what we can see from Figure 6, so we will not show them. For details, please refer to Reference [26]. Moreover, we also tested the performance of the solutions on synthetic datasets by varying the distributions of users and facilities, shown in Figure 7. Obviously, the phenomenon in synthetic datasets is inline with what we have seen in Figure 6.



(a) Aggregated distribution (b) Aggregated distribution (c) Even distribution (d) Even distribution

Fig. 7. Effectiveness comparison on synthetic graph.

Table 3. Execution Time of Each Algorithm (in Seconds)

	CA							BJ						
	1	5	10	15	20	25	30	1	5	10	15	20	25	30
*k*LNB	27	29	31	35	38	40	42	6	6	7	8	8	8	8
FR2FP	2,183	2,194	2,196	2,197	2,198	2,198	2,198	420	420	420	420	420	420	420
FR2FP-ex	2,213	2,217	2,214	2,217	2,216	2,217	2,217	427	428	426	427	426	426	426
LR	7,298	2,741	2,209	2,209	2,209	2,209	2,209	440	438	433	433	433	433	433

Comparison of efficiency. In this part, we compare the execution time of different algorithms and the results are shown in Table 3. Since *k*LNB does not need to calculate the distance matrix between virtual users and facilities, it runs the fastest. It is followed by FR2FP and FR2FP-ex, as these two algorithms need to compute distance matrices. The LR algorithm runs the slowest, especially for the CA dataset. This is because the LR algorithm not only needs to calculate the distance matrix but also iterates the subgradient algorithm many times. In addition, LR is not a polynomial time algorithm and its execution time will be affected by many factors (e.g., initial solution).

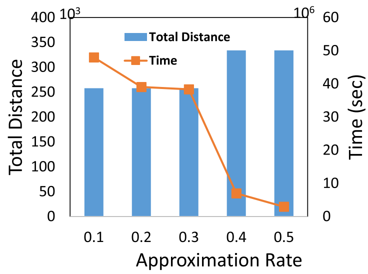
Effect of approximation ratio. The advantage of the LR algorithm is that the approximation ratio can be adjusted. In the previous experiments, we fixed the ratio (i.e., ϵ) to 0.1. In this experiment, we examine the time required for the LR algorithm and the final results under different ϵ. To better show the changes of the LR algorithm, we use the existing facility set F as the initial solution. The results are shown in Figure 8. As the approximation ratio gets better (decreased), the time required for the algorithm increases, and the final distance decreases, which is in line with our expectations. However, we can also find that the results of the algorithm may be the same under different approximation ratio. This is because if a solution satisfies the approximation ratio of 0.4, then it must also satisfy the approximation ratio of 0.5 and the two solutions are the same.

6.2 Results for the Budgeted k-FR Problem

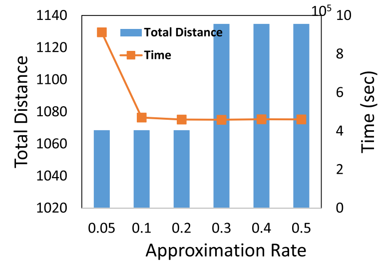
6.2.1 *Budgets in Experiment Setup.*

Datasets. We use the real data set described in Section 6.1.1. As there is no cost information in it, we select to generate it manually. The opening cost of candidate facilities was randomly generated from a uniform distribution over (200, 300). The closing cost of existing facilities was randomly generated from a uniform distribution over (50, 100). The closing cost is smaller than the opening cost, because opening a new facility is typically more costly than closing an existing facility. Notably, according to Reference [29], the scale of these costs is not relevant to the results (order) of replacement pairs.

The budget was also randomly generated from a uniform distribution. The sum of the cheapest k opening costs and the cheapest k closing costs is the lower bound of the distribution. The upper limit is the sum of the most expensive k opening costs and closing costs.



(a) CA



(b) BJ

Fig. 8. Effect of approximation rate.

Compared solutions. Since there is no off-the-shelf algorithm that can be directly used to solve the budgeted k-FR problem, we tuned kLNB algorithm to choose the optimal 1-FR pair each time the budget allows and made it as a baseline, referred to as kLNB-B. Therefore, we report the performance of the following three algorithms.

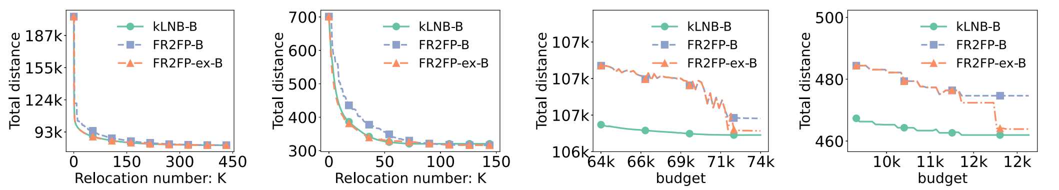
- kLNB-B: The adjusted kLNB algorithm.
- FR2FP-B: The solution proposed in Algorithm 3.
- FR2FP-ex-B: The solution proposed in Algorithm 4.

6.2.2 *Experimental Results for the Budgeted k-FR.*

Comparison of effectiveness. First, we compared the performance of different algorithms in solving the budgeted k-FR problem. The volume of the facility set scales from 100 to 1,000. They all have similar results, so we only show results when the scale is 1,000, which is shown in Figures 9(a) and 9(b). We can find that this result is very similar to that of the k-FR problem, which is shown in Figure 6. Although kLNB-B cannot provide a theoretical approximation ratio, it works well in practice. When the number of relocation is small, the effect of FR2FP-B is relatively poor. With the increase of the number of relocation, its performance continues to improve. The reason for this is the same as described earlier. FR2FP-ex-B has a good practical effectiveness while providing a theoretical approximation rate guarantee. In practice, we recommend the FR2FP-ex-B algorithm. If running time is sensitive, then the kLNB-B algorithm can also be used. However, in some cases, the kLNB-B algorithm may fail, we shall describe this in detail in Section 7.

Effect of budget. In previous experiments, we found that the relative performance of the various algorithms is similar to their performance in solving the k-FR problem. One reason for this result is that the budget is so large that this constraint has no effect. Therefore, we verify the impact of budget on different algorithms. In previous experiments, the budget is uniformly and randomly generated between the lower and upper bounds. In this part, the budget varies continuously between the lower and upper bounds, and the number of facilities to be relocated is constant, and then we examine the results obtained by the algorithm under different budgets. The result is shown in Figures 9(c) and 9(d).

As the budget gradually increases, the results obtained by different algorithms gradually decrease and eventually become stable. This is in line with our intuition. However, there are also some unexpected and interesting results. First, as the budget increases, the results are not steadily decreasing, but contains vibrations, and this phenomenon happen in all the three algorithms. This is because all algorithms utilize greedy methods, which consider only the local solution each time, not the global one. As the budget increases, there exist cases that a certain selection becomes better, but the overall result becomes worse.



(a) CA:1000 park (b) BJ:1000 bank (c) different budget (CA) (d) different budget (BJ)

Fig. 9. Experimental results of budgeted k-FR.

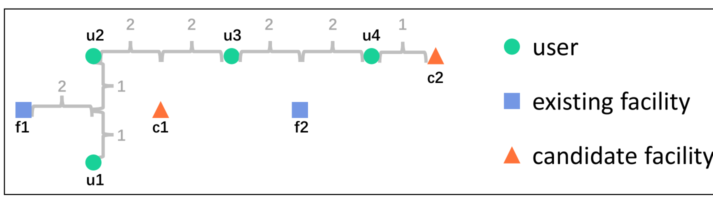


Fig. 10. A failure scenario of kLNB.

Second, different algorithms have different degrees of vibration, the kLNB-B algorithm is more stable, and the FR2FP-B algorithm vibrates more significantly. By deep investigation over the execution process of each algorithm, we observe that when the budget changes, only the last few steps of the kLNB-B algorithm will be affected, while the FR2FP-B algorithm is easily affected by the early steps. Since the FR2FP-B algorithm has many steps to perform after being affected by a budget change, it is more susceptible. FR2FP-ex-B is based on FR2FP-B and it of course is also susceptible.

Finally, when the budget is small, the kLNB-B algorithm is slightly better than the other two algorithms. When the budget is increased, the results of the three algorithms are roughly the same. Therefore, if the budget is not sufficient, the kLNB-B algorithm can be a good choice.

7 EXTENSIVE STUDY OF kLNB IN REFERENCE [27]

In this part, we introduce a scenario where the kLNB will fail. Through the above experiments, we found that although the kLNB does not have an approximation ratio, its performance is surprisingly superior to the FR2FP solution when k is small. However, as kLNB fails to guarantee the results quality while FR2FP does, there should exist cases that kLNB fails to work. Driven by that, we conduct extensive study, and finally find the cases that kLNB fail to perform well: *when multiple facilities need to be relocated at the same time and the relocation of any one of the facilities alone cannot bring the distance reduction.*

Figure 10 shows an example of this scenario. For ease of discussion, we use the Euclidean distance, which does not affect our conclusion in this example. According to the distance (the gray number) marked on the figure, we can calculate the distance between each user and the facilities, shown in Table 4. Suppose we are relocating two facilities, according to Table 5, no matter which pair is selected, it cannot bring a positive distance reduction. Therefore, kLNB will output nothing. Nevertheless, if we relocate f_1 and f_2 at the same time, then the distance will decrease eventually. Suppose FR2FP is adopted in this scenario, it will selects c_1 first, and then c_2, which is obviously better than kLNB.

Theorem 8. *The kLNB algorithm fails to provide any approximation ratio for the problem in Definition 2.*

Table 4. Distances between Users and Facilities

user \ facility	f_1	f_2	c_1	c_2
u_1	$\sqrt{5}$	$\sqrt{37}$	$\sqrt{5}$	$\sqrt{85}$
u_2	$\sqrt{5}$	$\sqrt{37}$	$\sqrt{5}$	9
u_3	$\sqrt{37}$	$\sqrt{5}$	$\sqrt{5}$	5
u_4	$\sqrt{101}$	$\sqrt{5}$	$\sqrt{37}$	1

Table 5. Distance Reductions for all 1-FR Pairs

pairs	reductions
$\langle f_1, c_1 \rangle$	0
$\langle f_1, c_2 \rangle$	$3\sqrt{5} - 1 - 2\sqrt{37} < 0$
$\langle f_2, c_1 \rangle$	$\sqrt{5} - \sqrt{37} < 0$
$\langle f_2, c_2 \rangle$	$2\sqrt{5} - 6 < 0$

Proof. We can prove this by contradiction. Suppose that the kLNB has an approximation ratio $\epsilon > 0$ for the problem. That is, given that the ideal solution of the problem is $\langle F_{k'}, C_{k'} \rangle_{OPT}$, then the solution of kLNB, say $\langle F_{k'}, C_{k'} \rangle_{LNB}$, should satisfy

$$\Delta(\langle F_{k'}, C_{k'} \rangle_{LNB}) \geq \epsilon \cdot \Delta(\langle F_{k'}, C_{k'} \rangle_{OPT}). \tag{11}$$

However, we can easily find an instance shown in Figure 10, where $\Delta(\langle F_{k'}, C_{k'} \rangle_{OPT}) = \sqrt{5} - 1$. However, the distance reduction of the result of kLNB is 0, i.e., $\Delta(\langle F_{k'}, C_{k'} \rangle_{LNB}) = 0$, which conflicts with Equation (11).

Therefore, the assumption does not hold, and kLNB algorithm fails to provide any approximation ratio for the problem. □

Although the kLNB algorithm does not have an approximate rate guarantee, it performs good in many cases, even better than FR2FP when k is small. However, it may encounter some awkward situations as shown in Figure 10. Due to that, we recommend to use FR2FP-ex algorithm to solve the k-FR problem, as it not only has the same result quality guarantee as FR2FP but also exhibits the best practical performance among the solutions.

8 CONCLUSION

Although k-FR problem has a wide range of applications in real life, the NP-completeness makes it difficult to find an algorithm reliable in real application. In this article, we proposed a group of approximate solutions to k-FR by transforming the problem into an equivalent facility placement one and a solution using linear programming. To the best of our knowledge, they are the first approximate solutions. We also found that the state-of-the-art heuristic solution, kLNB proposed by Reference [27], is effective in most cases, although it does not provide an approximation ratio. Results of exhaustive experiments show that if we want to guarantee the approximation ratio, we need to spend more time. Hence, in practice, we need to make a choice between efficiency and effectiveness. We also extensively study the budgeted k-FR problem and provide the corresponding algorithms to solve it.

REFERENCES

[1] Zitong Chen, Yubao Liu, Raymond Chi-Wing Wong, Jiamin Xiong, Ganglin Mai, and Cheng Long. 2014. Efficient algorithms for optimal location queries in road networks. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. 123–134.
[2] Gerard Cornuejols, Marshall L. Fisher, and George L. Nemhauser. 1977. Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Manage. Sci.* 23, 8 (1977), 789–810.
[3] Gérard Cornuéjols, George Nemhauser, and Laurence Wolsey. 1983. *The Uncapicitated Facility Location Problem*. Technical Report. Cornell University Operations Research and Industrial Engineering.
[4] Jiangtao Cui, Meng Wang, Hui Li, and Yang Cai. 2018. Place your next branch with MILE-RUN: Min-dist location selection over user movement. *Info. Sci.* 463–464 (2018), 1–20.

[5] Marshall L. Fisher, George L. Nemhauser, and Laurence A. Wolsey. 1978. An analysis of approximations for maximizing submodular set functions-II. In *Polyhedral Combinatorics*. Springer, 73–87.
[6] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Mauro Sozio. 2015. Scalable facility location for massive graphs on pregel-like systems. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM'15)*. ACM, 273–282.
[7] Arthur M. Geoffrion. 2010. Lagrangian relaxation for integer programming. In *50 Years of Integer Programming 1958–2008: From the Early Years to the State-of-the-Art*. Springer, 243–281.
[8] Parisa Ghaemi, Kaveh Shahabi, John P. Wilson, and Farnoush Banaei-Kashani. 2010. Optimal network location queries. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 478–481.
[9] Russell Halper, S. Raghavan, and Mustafa Sahin. 2015. Local search heuristics for the mobile facility location problem. *Comput. Oper. Res.* 62 (2015), 210–223.
[10] Richard M. Karp. 1972. Reducibility among combinatorial problems. In *Complexity of Computer Computations*. Springer, 85–103.
[11] Jon Lee, Vahab S. Mirrokni, Viswanath Nagarajan, and Maxim Sviridenko. 2009. Non-monotone submodular maximization under matroid and knapsack constraints. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*. 323–332.
[12] Feifei Li, Dihan Cheng, Marios Hadjieleftheriou, George Kollios, and Shang-Hua Teng. 2005. On trip planning queries in spatial databases. In *Proceedings of the International Symposium on Spatial and Temporal Databases*. Springer, 273–290.
[13] Yuhong Li, Yu Zheng, Shenggong Ji, Wenjun Wang, Leong Hou U, and Zhiguo Gong. 2015. Location selection for ambulance stations: A data-driven approach. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 1–4.
[14] Zhenhui Li, Bolin Ding, Jiawei Han, Roland Kays, and Peter Nye. 2010. Mining periodic behaviors for moving objects. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1099–1108.
[15] Gino J. Lim and Ayse Durukan Sonmez. 2013. γ-robust facility relocation problem. *Eur. J. Oper. Res.* 229, 1 (2013), 67–74.
[16] Kyriakos Mouratidis, Dimitris Papadias, and Spiros Papadimitriou. 2005. Medoid queries in large spatial databases. In *Proceedings of the International Symposium on Spatial and Temporal Databases*. Springer, 55–72.
[17] Subhash C. Narula, Ugonnaya I. Ogbu, and Haakon M. Samuelsson. 1977. Technical note—An algorithm for the *P*-median Problem. *Oper. Res.* 25, 4 (1977), 709–713.
[18] George L. Nemhauser and Laurence A. Wolsey. 1978. Best algorithms for approximating the maximum of a submodular set function. *Math. Oper. Res.* 3, 3 (1978), 177–188.
[19] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions-I. *Math. Program.* 14, 1 (1978), 265–294.
[20] Dimitris Papadias, Yufei Tao, Kyriakos Mouratidis, and Chun Kit Hui. 2005. Aggregate nearest neighbor queries in spatial databases. *ACM Trans. Database Syst.* 30, 2 (2005), 529–576.
[21] Jianzhong Qi, Rui Zhang, Lars Kulik, Dan Lin, and Yuan Xue. 2012. The min-dist location selection query. In *Proceedings of the IEEE 28th International Conference on Data Engineering*. IEEE, 366–377.
[22] Jianzhong Qi, Rui Zhang, Yanqiu Wang, Andy Yuan Xue, Ge Yu, and Lars Kulik. 2014. The min-dist location selection and facility replacement queries. *World Wide Web* 17, 6 (2014), 1261–1293.
[23] Ayse Durukan Sonmez and Gino J. Lim. 2012. A decomposition approach for facility location and relocation problem with uncertain number of future facilities. *Eur. J. Oper. Res.* 218, 2 (2012), 327–338.
[24] Jan Vondrák. 2008. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*. 67–74.
[25] Feng Wang, Li Chen, and Weike Pan. 2016. Where to place your next restaurant?: Optimal restaurant placement via leveraging user-generated reviews. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management (CIKM'16)*. ACM, 2371–2376.
[26] Hu Wang, Hui Li, Meng Wang, and Jiangtao Cui. 2021. Addressing the hardness of k-facility relocation problem: A pair of approximate solutions. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM'21)*. ACM, 1919–1928.
[27] Meng Wang, Hui Li, Jiangtao Cui, Sourav S. Bhowmick, and Ping Liu. 2020. FROST: Movement history-conscious facility relocation. *ACM Trans. Intell. Syst. Technol.* 11, 1, Article 9 (Jan. 2020), 26 pages.
[28] Meng Wang, Hui Li, Jiangtao Cui, Ke Deng, Sourav S. Bhowmick, and Zhenhua Dong. 2016. Pinocchio: Probabilistic influence-based location selection over moving objects. *IEEE Trans. Knowl. Data Eng.* 28, 11 (2016), 3068–3082.
[29] Qian Wang, Rajan Batta, Joyendu Bhadury, and Christopher M. Rump. 2003. Budget constrained location problem with opening and closing of facilities. *Comput. Oper. Res.* 30, 13 (2003), 2047–2069.

[30] R. A. Whitaker. 1983. A fast algorithm for the greedy interchange for large-scale clustering and median location problems. *INFOR: Info. Syst. Oper. Res.* 21, 2 (1983), 95–108.

[31] Brian J. Worton. 1989. Kernel methods for estimating the utilization distribution in home-range studies. *Ecology* 70, 1 (1989), 164–168.

[32] Xiaokui Xiao, Bin Yao, and Feifei Li. 2011. Optimal location queries in road network databases. In *Proceedings of the IEEE 27th International Conference on Data Engineering*. IEEE, 804–815.

[33] Da Yan, Zhou Zhao, and Wilfred Ng. 2011. Efficient algorithms for finding optimal meeting point on road networks. *Proc. VLDB Endow.* 4, 11 (2011), 968–979.

[34] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 325–334.

[35] Man Lung Yiu, Nikos Mamoulis, and Dimitris Papadias. 2005. Aggregate nearest neighbor queries in road networks. *IEEE Trans. Knowl. Data Eng.* 17, 6 (2005), 820–833.

[36] Donghui Zhang, Yang Du, Tian Xia, and Yufei Tao. 2006. Progressive computation of the min-dist optimal-location query. In *Proceedings of the 32nd International Conference on Very Large Data Bases*. 643–654.

[37] Yu Zhang, Wei Wei, Binxuan Huang, Kathleen M. Carley, and Yan Zhang. 2017. RATE: Overcoming noise and sparsity of textual features in real-time location estimation. In *Proceedings of the ACM on Conference on Information and Knowledge Management (CIKM'17)*. ACM, 2423–2426.

[38] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. 2009. Mining interesting locations and travel sequences from GPS trajectories. In *Proceedings of the 18th International Conference on World Wide Web*. 791–800.

[39] Liang Zhu, Yinan Jing, Weiwei Sun, Dingding Mao, and Peng Liu. 2010. Voronoi-based aggregate nearest-neighbor query processing in road networks. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 518–521.

Received 8 August 2022; revised 13 February 2023; accepted 15 February 2023