MC^{2}LS: Towards Efficient Collective Location Selection in Competition

Meng Wang, Mengfei Zhao, Hui Li, Jiangtao Cui, Bo Yang, and Tao Xue

***Abstract*—Collective Location Selection (CLS) has received significant research attention in the spatial database community due to its wide range of applications. The CLS problem selects a group of k preferred locations among candidate sites to establish facilities, aimed at collectively attracting the maximum number of users. Existing studies commonly assume every user is located in a fixed position, without considering the competition between peer facilities. Unfortunately, in real markets, users are mobile and choose to patronize from a host of competitors, making traditional techniques unavailable. To this end, this paper presents the first effort on a CLS problem in competition scenarios, called MC^{2}LS, taking into account the mobility factor. Solving MC^{2}LS is a non-trivial task due to its NP-hardness. To overcome the challenge of pruning multi-point users with highly overlapped minimum boundary rectangles (MBRs), we exploit a position count threshold and design two square-based pruning rules. We introduce IQuad-tree, a user-MBR-free index, to benefit the hierarchical and batch-wise properties of the pruning rules. We propose an $(1-\frac{1}{e})$-approximate greedy solution to MC^{2}LS and incorporate a candidate-pruning strategy to further accelerate the computation for handling skewed datasets. Extensive experiments are conducted on real datasets, demonstrating the superiority of our proposed pruning rules and solution compared to the state-of-the-art techniques.**

***Index Terms*—Collective location selection, peer competition, spatial index, moving objects, pruning rule.**

I. Introduction

THANKS to the proliferation of mobile internet and GPS-equipped devices, location-based services (LBS) fuel access and convenience for restaurant recommendation and online taxi services, etc., making moving users increasingly inclined towards them [1]. The large amount of geo-tagged data accumulated from LBS platforms, which depict the positions of users' movements, provide unprecedented support for business location decision-making. In general, chains or large corporations prioritize overall market share rather than the impact of an individual facility. For this reason, these companies use **C**ollective **L**ocation **S**election (CLS) [2] to mine a group of k preferred locations among candidate sites to establish facilities, aimed at collectively attracting the maximum number of customers. Budget is commonly the primary factor of k.

As critical strategic planning, a spectrum of CLS problems have been widely studied, ranging from chains expanding store networks [3], [4], urban authorities optimizing the distribution of public service facilities [5], [6], business planning for warehouse and advertising billboard sites [7], [8], [9], [10], and so on.

Traditionally, spatial proximity [11], [12] has been used to measure the attractiveness of locations to customers, each of which is considered at a single fixed position and patronizes to the nearest service facility. However, a study on **L**ocation **S**election (LS) for moving users or vehicles, PRIME-LS [13], has shown that the single-point model is inappropriate for objects with a series of activity positions. They verified each position of an object plays a role in the overall attractiveness and proposed a probability-based multi-point cumulative influence model. The influence relationship is defined as a cumulative probability of all positions instead of the nearest-neighbor-based yes-no binary logic. The probability of each position being influenced by a facility corresponds to a utility function negatively correlated with the distance between them [14]. Recently, a moving object-oriented CLS problem k-**C**ollective **I**nfluential **F**acility **P**lacement (referred to as k-CIFP) [15] has been studied based on the aforementioned multi-point model. The solution of k-CIFP is attained by progressively picking k candidates with the maximum marginal utility.

Most literature on CLS, especially regarding users as moving objects [10], [15], [16], focused solely on influencing potential customers without considering peer competition. However, a company in a market environment can hardly operate independently of its competitors. Ignoring the competition from peers will inevitably limit the applicability and effectiveness of existing CLS models. To illustrate the impact of competitors, let us consider the following scenario over moving users.

Example 1: Wendy's intends to select two of the three candidate locations to open new fast-food restaurants in the hope of better capturing the market. For the sake of discussion, each

Received 29 January 2024; revised 7 September 2024; accepted 26 November 2024. Date of publication 2 December 2024; date of current version 12 January 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62272369 and under Grant 62372352, in part by Natural Science Basic Research Program of Shaanxi under Grant 2023-JC-YB-558 and under Grant 2024JC-YBMS-473, in part by Scientific Research Program Funded by Education Department of Shaanxi Provincial Government under Grant 23JS028, in part by Scientific and Technological Program of Xi'an under Grant 24GXFW0016, in part by Xi'an Major Scientific and Technological Achievements Transformation Industrialization Project under Grant 23CGZH-CYH0008. Recommended for acceptance by K. Zheng. *(Corresponding author: Hui Li.)*

Meng Wang, Mengfei Zhao, Bo Yang, and Tao Xue are with the School of Computer Science & The Shaanxi Key Laboratory of Clothing Intelligence, Xi'an Polytechnic University, Xi'an 710048, China (e-mail: wangmeng@xpu.edu.cn; 961311016@qq.com; yangboo@stu.xjtu.edu.cn; xuetao@xpu.edu.cn).

Hui Li is with the School of Computer Science and Technology, Xidian University, Xi'an 710071, China, and also with the Shanghai Yunxi Technology, Shanghai 200051, China (e-mail: hli@xidian.edu.cn).

Jiangtao Cui is with the School of Computer Science and Technology, Xidian University, Xi'an 710071, China (e-mail: cuijt@xidian.edu.cn).

Digital Object Identifier 10.1109/TKDE.2024.3510100

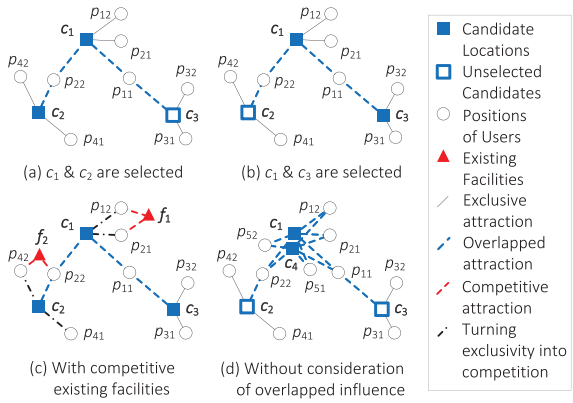


Fig. 1. Motivation example. The hollow squares c_3 and c_2 means they are excluded from the result sets, respectively.

user in Fig. 1 only records two moving positions (e.g., user o_i contains positions p_{i1} and p_{i2}). We assume that c_1 influences $\{o_1, o_2\}$, c_2 influences $\{o_2, o_4\}$, and c_3 attracts $\{o_1, o_3\}$. As shown in Fig. 1(a) and (b), choosing $\{c_1, c_2\}$ and $\{c_1, c_3\}$ as two result sets of 2-CIFP will derive the same quality (i.e., three-quarters) of market share, where they attract $\{o_1, o_2, o_4\}$ and $\{o_1, o_2, o_3\}$, respectively. However, the situation will become complicated once existing facilities are involved as competitors, such as f_1 and f_2 for *McDonald's*, which upset the equilibrium between the initial solutions $\{c_1, c_2\}$ and $\{c_1, c_3\}$. We assume that f_1 and f_2 in Fig. 1(c) influence $\{o_1, o_2\}$ and $\{o_2, o_4\}$, respectively. This results in candidate facilities having to compete with peers for o_1, o_2 and o_4, except for o_3, which is still exclusively captured by c_3. Intuitively, the fact that c_3 can monopolize o_3 and help c_1 consolidate the influence over o_1 against f_1 makes it appear that set $\{c1, c3\}$ has a competitive advantage over $\{c1, c2\}$.

Despite the progress made by LS studies on competition, they are not available in solving the aforementioned scenario due to at least one of the following two drawbacks. On one hand, their techniques are single-facility oriented [17], [18], [19] which cannot apply to the collective scenario. As in Fig. 1(d), both candidates c_1 and c_4 influence the most three users $\{o_1, o_2, o_5\}$, and thus $\{c_1, c_4\}$ is chosen as the optimal result. However, a serious issue was suffered that the method neglected the overlap of influence among selected facilities. Due to the overlapped influence between c_1 and c_4, it prevents maximizing the utility of influence (e.g., selecting $\{c_1, c_3\}$ can attract more users $\{o_1, o_2, o_3, o_5\}$). On the other hand, users are assumed to be static [4], [14], [20], making their methods unavailable to the multi-point moving model.

To this end, we present a novel CLS problem in this paper, namely **M**obility-oriented **C**ompetitive-based **C**ollective **L**ocation **S**election (**MC** ^{2} **LS**), which considers both mobility and competition factors to address the limitations that existing CLS techniques and LS studies on competition suffer from.

A naive solution to MC²LS is to exhaustively check all the candidate-user and facility-user pairs to derive their influence relationships, where each position of a user needs to be calculated. Then the optimal result with the maximum collective influence can be returned by enumerating all possible combinations of k candidates. Unfortunately, the NP-hardness of MC²LS (proven in Section IV) leads to exponential complexity. Given the overlap of influence among candidate locations and the impact from competitors, it is non-trivial to propose an efficient algorithm for massive datasets. As stated in works [13], [21], the activity regions of moving users, i.e., minimum bound rectangles (MBRs) of positions, are highly overlapped, which makes classical pruning rules fail to trim unnecessary users. To solve the efficiency issue, a previous study, PRIME-LS [13], sidestepped the challenge of pruning users by instead focusing on eliminating candidate locations.

Since the number of users is significantly larger than candidates, there is more potential for performance improvement by pruning users. Accordingly, this paper designs a novel spatial index structure, IQuad-tree, for pruning irrelevant multi-point users. The main idea is to use the cumulative probability model (detailed in Section III-A) as a bridge for backward derivation, transforming the measure for determining the influence relationship from distance to the number of a user's positions. By counting the positions around a facility or candidate, two new pruning rules are proposed regardless of highly overlapped MBRs. Integration with the hierarchical structure of Quad-tree [22] is helpful to batch-wise handle facilities/candidates in the same node. In brief, the IQuad-tree is a user-MBR-free index structure with superior performance compared to the existing pruning techniques for multi-point model.

Based on IQuad-tree, we design a four-stage framework to answer MC²LS. First, the pruning rules are utilized to identify the influence relationships between users and facilities/candidates, without the need for evaluating cumulative probability. Second, validation calculations are performed for users whose influence relationships remain undetermined. Third, we calculate competitiveness of candidates based on the evenly split competition model [14], [18], [23], where users are simultaneously influenced by multiple facilities. Finally, a heuristic is used to gradually select k candidates with the maximum marginal benefits to the solution set. This process leads to an approximate result of the MC²LS problem and can guarantee an $(1 - \frac{1}{e})$-approximation ratio. Extensive experiments on two real-world datasets demonstrate that our proposed method improves the efficiency by at least an order of magnitude compared to the baseline. In summary, our contributions in this paper are outlined as follows.

- To the best of our knowledge, this is the first effort to formalize the collective location selection problem taking both users' mobility and peer competition factors into account. Compared to existing works, MC²LS is a problem more in line with the market economy scenario.
- A novel user-MBR-free index structure, IQuad-tree, and two new pruning rules are designed to address the pruning hardness due to the overlapping activity regions between moving users.
- The IQuad-tree is superior to the state-of-the-art pruning strategies for the probability influence model of multi-point users. It is flexible and can not only prune users by itself but also be combined with existing pruning rules to improve the performance further.

- We theoretically prove the NP-hardness of the MC^{2}LS problem and present an IQuad-tree-based solution, which can guarantee an $(1 - \frac{1}{e})$-approximation ratio.
- We conduct empirical studies on real-world datasets to demonstrate the superiority of our proposed solution compared to a method adapted from the most relevant and state-of-the-art study.

The rest of this paper is organized as follows. We give a brief overview of related work in the next section. In Section III, we discuss the cumulative probability influence model and formally define the MC^{2}LS problem. We prove the NP-hardness of MC^{2}LS and propose two baseline algorithms in Section IV. Section V presents an efficient solution based on IQuad-tree and new pruning rules. Section VI analyzes the cost of the proposed methods. Section VII reports the empirical study and the last section concludes this paper.

II. Related Work

In this section, we discuss the related efforts for solving the collective and competitive location selection problems.

A. Collective Location Selection

Xu et al. [24] introduced the first work on a Group Location Selection (alias for CLS) query to select the minimum number of locations that collectively influence all the uncertain users. Given an influence distance range, a two-stage greedy framework was devised. Chen et al. [2] relaxed the constraint and aimed to cover the maximum number of users. Clustering-based and grid partitioning-based policies were developed to tighten the approximate search space. Wang et al. [25] suggested that a user was influenced by k nearest facilities (kNN) and designed a branch-and-bound greedy method with dynamic programming. Following the kNN spatial metric, Cai et al. [26] considered social influence and keyword similarity factors. They accelerated the greedy process by leveraging a R-tree-based index structure. Mir et al. [6] used k-medians clustering with an extra Min-dist constraint (minimizing the average distance between users and facilities). Following Min-dist, Wang et al. [27] studied a variant that focused on relocating k inferior facilities with alternatives on road network. Ning et al. [28] expanded the CLS problem where users and facility locations could vary over time slots.

Zhang et al. [9] defined a user trajectory-based CLS problem but only considered the nearest point on a trajectory to a facility, regardless of influence from other points. The authors [10] further developed a branch-and-bound paradigm upon hash-based aggregate trajectory index. Ali et al. [16] evaluated the influence score by the proportion of a user trajectory within a distance range and exploited a divide-and-conquer greedy algorithm based on a trajectory index. However, their assumption that closer and farther distance thresholds did not impact results deviated from the distance-related influence preference [29]. Following the distance-based preference semantics, Wang et al. [13] verified that the cumulative influence probability model outperformed the nearest neighbor (e.g., [9]) and range (e.g., [16]) influence semantics. The authors [15] devised a greedy solution to CLS with pruning rules and FM sketches. Zhang et al. [30] described influence as if a user trajectory passed through a certain number of facilities. An upper-bound progressive pruning and a tangent line-based approximation were developed. A time-aware CLS problem [3] was addressed by a greedy heuristic and an online streaming model leveraging grid-trajectory and time-trajectory index structures.

Aside from the above works by the database community, problems with various constraints from the operations research field were studied. Chakrabarty et al. [31] made effort on the continuous k-medians problem with facility opening costs. They solved the problem using a linear program with the ellipsoid algorithm. Chen et al. [7] studied a capacitated Min-dist CLS problem. Hierarchical clustering were applied to a Mixed Integer Nonlinear Programming. Wang et al. [32] modeled the CLS problem a combinatorial optimization with traffic flow and proposed a divide-and-conquer mechanism. Saha et al. [8] utilized the Delphi technique based on the Fermetean fuzzy sets and double normalized MARCOS approach.

The above studies contributed to CLS and its variants. However, none of them considered the competitive factor from peer facilities. Hence, their techniques cannot be easily adapted to address the MC^{2}LS problem.

B. Competitive Location Selection

The classical competitive LS [18] was defined as opening a new facility to attract the maximum number of customers against competing facilities. A user's demand was evenly split between all the facilities that can attract the user. Following the same split metric, Liu et al. [17] reduced computational cost by utilizing an influence pruning policy to filter out users corresponding to inferior candidates. Zeng et al. [19] considered only the nearest facility as the competitor. For the distances from a user to a candidate and a competitor, the ratio was defined as a relative distance influence weight. Two pruning rules and a reverse influence sampling algorithm were proposed. They [33] extended the problem by restricting the affected users with a distance threshold and an extra clustering-based pruning was devised to prevent redundant computations. Study [34] focused on refining the influence utility model of shopping center, where Huff's model was applied. They evaluated the vitality of locations according to surrounding points of interest, transportation situations, and social interests.

The above studies focused on stationary users and only selected a single optimal location. As their problem settings were orthogonal to ours, their efforts cannot be applied.

Aboolian et al. [14] devised a non-linear knapsack model for competitive LS. The influence of a facility was depicted as the ratio of its utility, which was negatively correlated with the distance, to the total utility of all rivals. They employed an adaptive grid policy based on Tangent Line Approximation. Kung et al. [20] incorporated network benefit to the model, which indicated extra gains brought by nearby chain sites. They approximated the problem by a kink function, and the linear integer program was decomposed into sub-problems. Shan et al. [4] further integrated price as a component beyond the distance factor. They used a bi-level model where the lower model updated the price until a

Nash equilibrium was reached with the competitor. Drezner et al. [35] analyzed an evenly-split-based model between only two competitive firms. One optimized its store locations to capture market share, while the other adjusted its strategy to fight back until the equilibrium. Qi et al. [36] mathematically transformed a two-layer optimization of a similar problem into a single-layer mixed-integer nonlinear programming.

Although these studies considered the collective factor, unfortunately, they were still based on static single-point users, ignoring the impact of users' moving behaviors on the patronage of facilities. As a result, their influence models also struggle to cope with the MC^{2}LS problem.

III. Problem Formulation

In this section, we formally describe the MC^{2}LS problem. We begin by introducing the preliminary of the probability-based influence model over moving users [13] and terminologies that are necessary for the definition.

A. Preliminaries

Following the mobility-aware influence criterion [15], a moving user o (or vehicle, animal, etc.) is described as a series of r positions $o = \{p_1, p_2, \ldots, p_r\}$ in two-dimensional Euclidian space $\mathbb{R}^2$, each of which represents a geographical coordinate $p_i = \langle latitude, longitude \rangle$. Given two positions p_1 and p_2, the distance between them is denoted by $d(p_1, p_2)$. Besides the set $C = \{c_1, c_2, \ldots, c_n\}$ of candidate locations for opening new facilities in traditional CLS, we include an extra set $F = \{f_1, f_2, \ldots, f_m\}$ of existing facilities as competitors. Any facility or candidate is located at a stationary position. As both facilities and candidates can influence users, to simplify the discussion, we introduce a concept of *abstract facility* v, where $v \in C \cup F$. For a moving user o, the probability that o is influenced by v at position $p_i \in o$ is denoted as $Pr_v(p_i)$, which is independent of those at other positions. Taking into account the attraction received by a moving user o at all positions, the overall influence probability is defined as follows.

Definition 1: Given an abstract facility $v \in C \cup F$ and a moving user o, the *cumulative influence probability* of o being influenced by v, denoted as $Pr_v(o)$, is computed as $Pr_v(o) = 1 - \prod_{i=1}^{r}(1 - Pr_v(p_i))$ [13].

In practice, some companies prioritize the individuals more certain to be attracted, referring to the *quality* of influence, while others consider a broader range of customers, referring to the *quantity* of influence. To this end, a probabilistic threshold τ for making trade-offs between quality and quantity is presented to filter the users of interest to the business.

Definition 2: Given an abstract facility $v \in C \cup F$, a moving user o and a probability threshold τ, v can *influence* o if and only if $Pr_v(o) \geq \tau$. Further, given a set $\Omega = \{o_1, o_2, \ldots, o_n\}$ of moving users, the subset consisting of users influenced by v is defined as Ω_v, whose cardinality, denoted by $inf(v)$, is defined as the *influence* value of v [13].

The influence model can be applied to various scenarios based on different distance-based probability functions that depict diverse behavior patterns of influence [29]. Specifically, $Pr_v(p_i) = PF(d(v, p_i))$, where $PF(\cdot)$ denotes a distance-based probability (utility) function that describes the monotonically decreasing influence preference with distance $d(v, p_i)$ [13], [14]. Then $Pr_v(o)$ in Definition 1 can be calculated as $Pr_v(o) = 1 - \prod_{i=1}^{r}(1 - PF(d(v, p_i)))$.

Example 2: For user $o_1 = \{p_{11}, p_{12}\}$ in Fig. 1(c), we assume $Pr_{c_1}(p_{11}) = 0.6$, $Pr_{c_1}(p_{12}) = 0.3$, $Pr_{f_1}(p_{11}) = 0.65$ and $Pr_{f_1}(p_{12}) = 0.2$. Then we have $Pr_{c_1}(o_1) = 1 - (1 - 0.6)(1 - 0.3) = 0.72$ and $Pr_{f_1}(o_1) = 1 - (1 - 0.65)(1 - 0.2) = 0.72$. If a business set $\tau = 0.7$, both c_1 and f_1 can influence o_1.

B. Problem Definition

The probabilistic influence in Definition 2 means a moving user can be attracted by multiple abstract facilities simultaneously, only if their probabilities of influencing the user are greater than or equal to τ.

Definition 3: Given a set F of existing facilities, the *set of competitive facilities that can influence a moving user* o is denoted as $F_o = \{f | Pr_f(o) \geq \tau \wedge f \in F\}$.

According to the *evenly split* competition model [14], [18], [23], all facilities that attract the same user o will equally capture the influence. In other words, each facility in F_o obtains $1/|F_o|$ influence on user o. Taking into account competitors in F_o, the *competitive influence of a candidate* c *on* o, denoted as $cinf(c, o)$, can be calculated as

$$cinf(c, o) = \frac{1}{|F_o| + 1}, \tag{1}$$

As a result, the total influence of a candidate location in a competitive environment is no longer determined solely by the number of objects it attracts. Instead, it is summarized by the captured parts of influences on all the attracted users.

Definition 4: Given a set Ω_c of moving users influenced by c and a set F of existing facilities, the *competitive influence of a candidate* c, denoted by $cinf(c)$, is computed as

$$cinf(c) = \begin{cases} 0 & \Omega_c = \emptyset, \\ \sum_{o \in \Omega_c} cinf(c, o) & otherwise. \end{cases} \tag{2}$$

The above definition is consistent with our intuition that the more users a candidate location attracts or the fewer existing facilities competing with it, the higher its influence.

Below, we choose a set of candidates to collectively enlarge their influence, which is more complicated than selecting a single one. Consider the scenario of opening fast-food chain restaurants or clothing stores that are affiliated with the same corporation. The stores of an organization are not in competition with each other, but rather operate as a cohesive entity to compete against existing facilities in the market. Any store that successfully draws in customers (i.e., where customers make purchases) contributes to the market share. Even if multiple stores in the organization meet the criteria of Definition 2 and can influence a specific user, the user will access at most one of these stores for service. Obviously, the overlapped influences of stores are unhelpful in increasing market share and result in wasted construction costs. Consequently, when choosing a set of candidate locations, it is advisable to avoid this situation to optimize their overall cost-effectiveness.

Definition 5: Given a set Ω of moving users and a set G of candidate locations, where $G \subseteq C$, the *set of influenced users by candidates in* G is defined as $\Omega_G = \{o | Pr_c(o) \geq \tau \wedge c \in G \wedge o \in \Omega\}$.

Definition 6: Given a set Ω of moving users, a set F of existing facilities and a set G of candidate locations, where $G \subseteq C$, the *competitive collective influence of* G, denoted by $cinf(G)$, is computed as $cinf(G) = \sum_{o \in \Omega_G} \frac{1}{|F_o|+1}$.

Definitions 5 and 6 regard the candidate set as a whole that offers services to users while preventing the repeated accumulation of overlapping influences from multiple candidates on the same users. By comparing the competitive collective influence of the sets composed of different candidate combinations, their ranks can be evaluated. Now, we are ready to define the MC^{2}LS problem addressed in this paper.

Definition 7: Given a set Ω of moving users, a set F of existing facilities, a set C of candidate locations and the number $k \in \mathbb{Z}^+$ of desired candidate locations, the **M**obility-oriented **C**ompetitive-based **C**ollective **L**ocation **S**election (**MC^{2}LS**) problem aims to find an optimal candidate subset $G \subseteq C \wedge |G| = k$ to maximize its competitive collective influence, i.e., for $\forall G' \subseteq C \wedge |G'| = k$, we have $cinf(G) \geq cinf(G')$.

Example 3: Let us revisit Example 1 with the peer competitors f_1, f_2 which attract $\{o_1, o_2\}$ and $\{o_2, o_4\}$, respectively. For the two candidate sets $G_1 = \{c_1, c_2\}$ and $G_2 = \{c_1, c_3\}$, when k is set to 2, we can derive their competitive collective influences according to Definition 6 as follows:

$$cinf(G_1) = \sum_{o \in \Omega_{G_1} = \{o_1, o_2, o_4\}} \frac{1}{|F_o|+1} = \frac{4}{3},$$

$$cinf(G_2) = \sum_{o \in \Omega_{G_2} = \{o_1, o_2, o_3\}} \frac{1}{|F_o|+1} = \frac{11}{6}.$$

As $cinf(G_2) > cinf(G_1)$, G_2 is the superior result set, which is consistent with our previous intuition in Example 1.

IV. Baseline Solutions to MC^{2}LS

In light of Definition 7, a straightforward and exact solution to the MC^{2}LS problem is to exhaustively check all the combinations of candidate locations and find the optimal group. Given a set of n candidate locations, the number of possible combinations consisting of k candidates is $\binom{n}{k}$. Unfortunately, computing factorials in polynomial time is exceedingly challenging, particularly when the cardinality n is fairly large (e.g., *McDonald's* operates dozens of chains in every major city in the U.S.[1]).

In this section, we first prove the NP-hardness of the MC^{2}LS problem and then propose two baselines, a heuristic algorithm and an adapted CLS method, to address MC^{2}LS.

Theorem 1: The MC^{2}LS problem in Definition 7 is NP-hard.

Proof: As discussed in [37], the *Maximum k-Coverage* problem is NP-hard. Specifically, given a universal U set of elements $(|U| = m)$ and a collection of subsets $S = \{S_1, S_2, \ldots, S_i, \ldots, S_n\}$ $S_i \subseteq U$, where each element $u \in U$ has an associated weight $w(u)$, the *Maximum k-Coverage* problem selects k $(k \in \mathbb{Z}^+)$ subsets from S such that the total weight of elements in their union is maximized.

According to Definition 6, we have $\Omega_C = \bigcup_{c_i \in C} \Omega_{c_i} = \{o | o \in \Omega \wedge Pr_{c_i}(o) \geq \tau \wedge c_i \in C\}$. Then we construct a mapping $C \rightarrow \Omega_C$, where each candidate $c_i \in C$ corresponds to the set Ω_{c_i} of users influenced by c_i. Let Ω_C be treated as another universal set U', where each $o \in \Omega_C$ corresponds to an element $u' \in U'$. Then the set $\{\Omega_{c_1}, \Omega_{c_2}, \ldots, \Omega_{c_n}\}$ can be regarded as the counterpart collection of subsets S' where $S'_i = \Omega_{c_i} \subseteq U'$. Given a definite facility set F as competitors, $\forall c_i \in C$, if $Pr_{c_i}(o) \geq \tau$ holds, we have $cinf(c_i, o) = \frac{1}{|F_o|+1}$ based on (1). In other words, $\frac{1}{|F_o|+1}$ can be the associated weight $w(u')$ of element $u' \in U'$. Hence, the MC^{2}LS problem is identical to the *Maximum k-Coverage* problem in nature, and so is its NP-hardness.

Note that there might be users in Ω who are not influenced by any candidate, i.e., $\Omega_C \subset \Omega$. Nevertheless, since they are not involved in the calculation of total weight, this impacts nothing on the above conclusion. ■

A. Baseline Greedy Algorithm

Due to the NP-hardness of MC^{2}LS, a heuristic is more promising approach to find a near optimal solution in a reasonable time. Definition 4 enables the exact competitive influence of every candidate location, which means the greedy algorithm employed for the approximate solution to *Maximum k-Coverage* can also be applicable for MC^{2}LS. It works in three steps. First, we initialize the result set G as empty and compute $cinf(c)$s based on users Ω for all the candidates in C against competitive facilities in F. Second, the candidate with the maximum competitive influence is selected as the current optimal candidate c_{opt} and involved in set G. In order to maximize the overall influence coverage utility of the k preferred candidates, users already attracted by c_{opt} need not be considered in the subsequent candidate selection, i.e., avoiding unprofitable influence overlap between the selected candidates. Third, we iteratively use $\Omega = \Omega \setminus \Omega_G$ and $C = C \setminus G$ to compute $cinf(c)$s and put the next optimal candidate into G. The iterations continue until k candidates have been chosen.

Example 4: We illustrate the greedy process with $k = 2$ based on Example 1. According to Definition 4, we have $cinf(c_1) = \sum_{o \in \Omega} cinf(c_1, o) = 1/(|F_{o_1}| + 1) + 1/(|F_{o_2}| + 1) = 1/2 + 1/3 = 5/6$. Similarly, $cinf(c_2) = 5/6$ and $cinf(c_3) = 3/2$, respectively. Following the criteria of step two, c_3 is selected and involved in G. As o_1 and o_3 have been influenced by G, they are filtered out from Ω, leaving only o_2 and o_4. We continue to compute $cinf(c_1) = 1/2$ and $cinf(c_2) = 5/6$ in step three, and candidate c_2 is selected as the second optimum. Finally, we obtain the result set $\{c_2, c_3\}$.

B. Algorithm Adapted From k-CIFP

The aforementioned baseline greedy algorithm can offer an approximate solution to the MC^{2}LS problem, while the computation overhead is excessive (detailed in Section VII). To our

[1][Online]. Available: https://www.scrapehero.com/location-reports/McDonalds-USA/

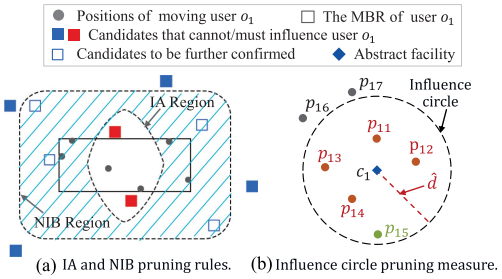


Fig. 2. Illustrations of pruning concepts.

knowledge, there is no method directly available to deal with MC²LS. Considering the similarity between the *k*-CIFP [15] and MC²LS problems in terms of collectivity and mobility, we incorporate the competition factor into the *k*-CIFP method, which was based on the state-of-the-art PINOCCHIO technique [13] for multi-point cumulative influence model.

The main idea is to eliminate redundant computations of influence for candidate locations with the help of two regions, Influence Arcs (IA) and Non-Influence Boundary (NIB) (shown in Fig. 2(a)). These regions were designed by a previous mobility-aware influence model [13] based on a distance metric called *influence radius*, $minMaxRadius(\tau, r)$ (hereinafter referred to as $mMR(\tau, r)$), which is used to measure multi-point-based influence relationship. Specifically, given the number r of user o's positions, the probability function PF and threshold τ, $mMR(\tau, r)$ is a transformation of $Pr_v(o) = \tau$ and computed as $mMR(\tau, r) = PF^{-1}(1 - (1 - \tau)^{1/r})$. Two corollaries of the $mMR(\tau, r)$ were derived in [13]:

Corollary 1: If all r positions of a user are located in the circle centered on a given candidate location with $mMR(\tau, r)$ as the radius, the candidate necessarily attracts the user;

Corollary 2: If none of the positions is within the circle, the user cannot be influenced.

By combining the enclosed MBR of o's positions, IA and NIB regions were derived based on the above two corollaries, respectively. Accordingly, IA region identifies candidate locations that necessarily influence user o, while NIB excludes candidates that cannot. The remaining candidates inside the interstitial region (the blue shaded area in Fig. 2(a)), whose influence relationships are yet to be confirmed, require refinement according to Definition 2. By evaluating all users' IA and NIB regions, we can efficiently obtain the set of users attracted by each candidate.

The adaptation to *k*-CIFP is extended by computing the set of existing facilities that can influence each user by the IA and NIB pruning rules. Based on that, we evaluate $cinf(c)$ for each candidate against the competitive facilities and choose the candidate with the largest $cinf(c)$ into the result set. Following the similar idea of the baseline greedy algorithm, we iteratively perform its second and third steps k times to obtain the final result. Algorithm 1 shows the detailed steps.

We initialize the optimal result set G and two key-value sets, Ω_c and F_o with every candidate and existing facility as the key, respectively. To benefit from the IA and NIB pruning rules, we employ two separate R-trees [38], RT_C and RT_F, to index sets C and F, which can be applied for the efficient

Algorithm 1: Adapted *k*-CIFP.

Input: Ω, F, C, k, τ and $PF(\cdot)$
Output: the optimal set G of candidate locations
1 initialize $G := \emptyset$, $\Omega_c := \emptyset$ $(\forall c \in C)$ and $F_o := \emptyset$ $(\forall o \in \Omega)$;
2 construct R-trees RT_C of C and RT_F of F;
3 **foreach** $o \in \Omega$ **do**
4 $\quad C_{IA} := RangeQuery(RT_C, IA(o))$;
5 $\quad$ **foreach** $c \in C_{IA}$ **do**
6 $\quad\quad \Omega_c := \Omega_c \cup \{o\}$;
7 $\quad$ **foreach** $c \in RangeQuery(RT_C, NIB(o)) \setminus C_{IA}$ **do**
8 $\quad\quad$ **if** $Pr_c(o) \geq \tau$ **then**
9 $\quad\quad\quad \Omega_c := \Omega_c \cup \{o\}$;
10 **foreach** $o \in \Omega' (\Omega' = \bigcup_{c \in C} \Omega_c)$ **do**
11 $\quad F_{IA} := RangeQuery(RT_F, IA(o))$;
12 $\quad F_o := F_o \cup F_{IA}$;
13 $\quad$ **foreach** $f \in RangeQuery(RT_F, NIB(o)) \setminus F_{IA}$ **do**
14 $\quad\quad$ **if** $Pr_f(o) \geq \tau$ **then**
15 $\quad\quad\quad F_o := F_o \cup \{f\}$;
16 **while** k *times* **do**
17 $\quad$ pair $P_{op} := \langle NULL, 0 \rangle$;
18 $\quad$ **foreach** $c \in C$ **do**
19 $\quad\quad$ compute or re-compute $cinf(c)$;
20 $\quad\quad$ **if** $cinf(c) > P_{op}.second$ **then**
21 $\quad\quad\quad P_{op} := \langle c, cinf(c) \rangle$;
22 $\quad G := G \cup \{P_{op}.first\}$, $C := C \setminus G$;
23 $\quad$ **foreach** $c \in C$ **do**
24 $\quad\quad \Omega_c := \Omega_c \setminus \Omega_{P_{op}.first}$;
25 return G;

range query operation (lines 1-2). Functions $IA(o)$ and $NIB(o)$ retrieve the IA and NIB regions of a user o, respectively. We recursively conduct the $RangeQuery()$ function on both the R-trees for each user. The process retrieves the candidates and facilities that can influence each user and adds them to sets C_{IA} and F_{IA}, respectively (lines 4,7,11,13). The candidates and facilities within NIB but not inside IA need to be verified based on Definition 2 (lines 7-8,13-14). Then we update Ω_c and F_o (lines 6,9,12,15). Equation (1) requires that c influences user o. Hence, evaluating only users influenced by at least one candidate reduces computational costs for existing facilities (line 10). Next, it iterates the greedy steps k times. We use a key-value pair P_{op}, in the form of $\langle c, cinf(c) \rangle$, to record the current optimal candidate (lines 17,21). Notably, $cinf(c)$ needs to be re-computed in the next iteration after G and Ω are updated (lines 19-20, 22-24). Once the iteration is complete, G will comprise the optimal k collective candidates (line 25).

V. IQUAD-TREE-BASED SOLUTION TO MC²LS

Based on pruning unnecessary abstract facilities, adapted k-CIFP outperforms the baseline greedy algorithm. In practice, the number of moving users greatly exceeds that of facilities, which means the computational cost would be reduced if unnecessary users could also be eliminated. However, the challenge lies in the high overlap among moving users' MBRs [21], which renders classical index-based pruning rules unable to distinguish which

users can be filtered out. That is why the PINOCCHIO technique used in k-CIFP pruned candidates in the opposite direction [13].

In this section, we proposed a *user-MBR-free* strategy via theoretical deduction, which transforms the way of determining the influence relationship by counting the number of a user's positions in a specific region instead of computing $Pr_v(o)$. Accordingly, we derive two novel pruning rules that are not subject to the limitation of MBR overlap. Integrating these pruning strategies into Quad-tree leads to a new index structure, which enables us to propose a more efficient solution with guaranteed approximation ratio.

A. Pruning Measure Based on Position Count

As discussed in Section IV-B, the two corollaries use a circle with radius $mMR(\tau, r)$ to filter candidates. The circles correspond to users' MBRs, which are highly overlapped, making it infeasible to prune users. Therefore, to enable pruning unnecessary computations for users, we adopt reverse thinking on the $mMR(\tau, r)$ formula to derive a new threshold based on position count to determine the influence relationship. We perform the following constant transformation of $mMR(\tau, r)$.

$$mMR(\tau, r) = PF^{-1}\left(1 - (1-\tau)^{1/r}\right)$$
$$\Rightarrow \quad r = 1/\log_{1-\tau}\left(1 - PF(mMR(\tau, r))\right). \quad (3)$$

Definition 8: Given a probability threshold τ, a probability function PF and a distance $\hat{d}$, we define the *position count threshold of* $\hat{d}$ as $\eta(\tau, PF, \hat{d}) = 1/\log_{1-\tau}(1 - PF(\hat{d}))$.

To facilitate the discussion, we define the circle centered on an abstract facility v with a given radius d_{radius} as an *influence circle*, denoted by $\phi(v, d_{radius})$.

Lemma 1: Given a position count threshold $\eta(\tau, PF, \hat{d})$, an abstract facility v and a moving user o with r positions, if the influence circle $\phi(v, \hat{d})$ encloses $\lceil \eta(\tau, PF, \hat{d}) \rceil$ positions of o, where $\lceil \eta(\tau, PF, \hat{d}) \rceil \leq r$, abstract facility v must influence o.

Proof: Without loss of generality, let a user o' consist of the exactly $\lceil \eta(\tau, PF, \hat{d}) \rceil$ positions, then v necessarily influences o' according to the aforementioned Corollary 1 with $mMR(\tau, r)$. As o' is equivalent to a subset of o, we have $Pr_v(o') \leq Pr_v(o)$, which means v must influence o. ■

As shown in Fig. 2(b), assuming that $\eta(\tau, PF, \hat{d}) = 5$, then c_1 must influence user o_1 as five positions of o_1 are covered by $\phi(c_1, \hat{d})$. If p_{15} were not a position of o, i.e., only four points were covered by $\phi(c_1, \hat{d})$, Lemma 1 would not meet, and the influence relationship needs to be confirmed by Definition 2.

B. Pruning Rules

According to Lemma 1, given a distance $\hat{d}$, we can effectively filter out users that are necessarily influenced by an abstract facility v. Next, we further try to prune users for a batch of candidates at once.

Lemma 2: Given a square $ABCD$ with diagonal lengths $\hat{d}$, any abstract facility v located within square $ABCD$ can influence a user o, if at least $\lceil \eta(\tau, PF, \hat{d}) \rceil$ positions of o are also covered by square $ABCD$.

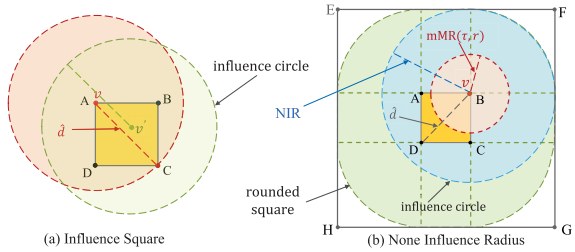


Fig. 3. IS and NIR pruning rules.

Proof: Since at least $\lceil \eta(\tau, PF, \hat{d}) \rceil$ positions of o are inside square $ABCD$, any abstract facility v can influence o if $\phi(v, \hat{d})$ can cover the square, i.e., $\phi(v, \hat{d})$ encloses $\eta(\tau, PF, \hat{d})$ positions of o (e.g., the light green influence circle in Fig. 3(a)). On the other hand, let v be an abstract facility located on one of the four corners of $ABCD$, say on A in Fig. 3(a), then $\phi(v, \hat{d})$ (the red influence circle) covers $ABCD$. As the diagonal is the longest distance between two points inside a square, then $\phi(v', \hat{d})$ of every abstract facility v' located within $ABCD$ can cover the square. In summary, the lemma is proved. ■

Users probably have different numbers of positions, and so do the corresponding $mMR(\tau, r)$s. We define the maximum $mMR(\tau, r)$ of all the users as *Non-influence Radius*, denoted by $NIR = mMR(\tau, r_{\max})$ where $r_{\max} = \max\{r | r = |o| \wedge o \in \Omega\}$, and it is applied to prune users that cannot be influenced.

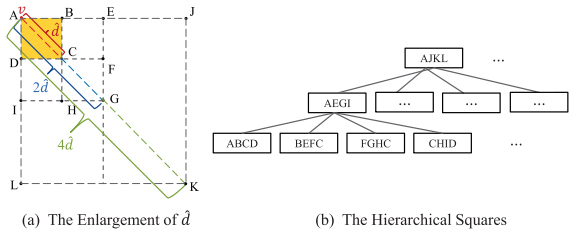
Lemma 3: As illustrated in Fig. 3(b), given a square $ABCD$ with diagonal lengths $\hat{d}$, we define a *NIR rounded square*, whose rounded corners are centered on the corners of $ABCD$ with NIR as radii. Any abstract facility v located within square $ABCD$ cannot influence user o, if none of o's positions is inside the MBR (i.e., $EFGH$) of the *NIR rounded square*, denoted by $\square_{NIR}(ABCD)$.

Proof: For a user o with r positions, we can draw an influence circle $\phi(v, mMR(\tau, r))$ centered at abstract facility v. For abstract facilities enclosed by square $ABCD$, $\phi(v, mMR(\tau, r))$ must be inside the NIR rounded square as NIR is the upper bound of all the mMRs (e.g., the red influence circle in Fig. 3(b) whose radius is shorter than NIR). As there is no position of user o inside $\square_{NIR}(ABCD)$, i.e., $EFGH$, influence circle $\phi(v, mMR(\tau, r))$ cannot cover any position. According to Corollary 2 of the $mMR(\tau, r)$ discussed above, the lemma can be proved. ■

According to Lemmas 2 and 3, we can efficiently filter out users who are necessarily influenced or not influenced in a given square, referred to as the *Influence Square (IS) and NIR pruning rules*, respectively.

C. IQuad-Tree

Compared to the IA and NIB pruning regions, which cannot be utilized for pruning users or batch processing subject to the MBR and distance metric related to a user's positions, the IS and NIR pruning rules offer two distinct advantages. First, they are only required to validate the count of a user's positions within a square of any given size $\hat{d}$ instead of concerning the user's MBR. Second, the rules can handle a batch of abstract facilities



(a) The Enlargement of $\hat{d}$ (b) The Hierarchical Squares

Fig. 4. The idea of the square hierarchy index.

within the square in one go. The batch-wise and fixed-size characteristics make them ideal for designing a spatial index structure to optimize computation.

Beyond Lemma 2, a possible case arises when a user o fails to meet the position count threshold within a square $ABCD$: o may have additional positions outside and around $ABCD$. Then o could possibly satisfy a new position count threshold by enlarging the metric $\hat{d}$. In Fig. 4(a), if the square $ABCD$ with $\hat{d}$ as the diagonal cannot meet Lemma 2, we extend $\hat{d}$ to $2\hat{d}$ and verify the condition in $AEGI$. And if $2\hat{d}$ does not work, it can be further increased to $4\hat{d}$ for validation, and so forth. This manner constitutes the hierarchy of squares depicted in Fig. 4(b), which motivates us to incorporate the pruning rule into Quad-tree to develop an *IQuad-tree* (Influence Quad-tree) for indexing users and their positions.

The leaf node N^L of an IQuad-tree comprises entries in the form of $\langle rect, \mathcal{P}, \Omega_{inf}, \Omega_{vrf} \rangle$. Spatial region $rect$ denotes the square of node N^L, whose diagonal is a predefined distance $\hat{d}$. The set $\mathcal{P}$ consists of *key-value pairs* each of which is of the form $o_{id}(o_{posIn})$, where the key o_{id} is the ID of a moving user who has at least one position within $N^L.rect$, and the value o_{posIn} associates only the subset of o_{id}'s positions located inside $N^L.rect$. The sets Ω_{inf} and Ω_{vrf} hold users that are necessarily and possibly influenced by abstract facilities within $N^L.rect$, respectively. Initially, both Ω_{inf} and Ω_{vrf} are empty. By eliminating users who are necessarily unaffected based on Lemma 3, we can obtain users in Ω_{vrf} of a leaf node. The users in $\Omega_{vrf} \setminus \Omega_{inf}$ require to be further verified.

For a non-leaf node N^N, the entry takes the form of $\langle rect, \mathcal{P}, \Omega_{inf}, visited \rangle$. Component $rect$ is the area enclosed by its four child squares. The set $\mathcal{P}$ represents the unions of the $N^L.\mathcal{P}$ or $N^N.\mathcal{P}$ sets in the four child nodes. The set Ω_{inf} of N^N follows the idea of the enlargement as Fig. 4(a). We verify users via twice the diagonal length of the child node. If users meet the condition of Lemma 2, they are affected by the abstract facilities inside $N^N.rect$ and appended to N^N. The binary value $visited$ indicates whether N^N has been traversed or not (i.e., 1 or 0).

For both the N^L and N^N nodes, users in Ω_{inf} are achieved according to Lemma 2. To facilitate verification, we construct an attached *Hash* structure $\langle d_{diagonal}, \eta(\tau, PF, d_{diagonal}) \rangle$ for the IQuad-tree, which stores the position count threshold corresponding to the diagonal of the square at each level of the tree. Then the computation of position count threshold can be done by $O(1)$ look-up operation. The IQuad-tree in Fig. 5(a) is an illustration of moving users shown in Fig. 5(b).

The main benefits of using IQuad-tree are twofold. First, for an abstract facility in a specific node, the users influenced by it are recorded in Ω_{inf}. This allows us to instantly retrieve the influence relationships of other abstract facilities located in the same node without calculation. Second, for moving users who have been identified to be influenced by candidates in a small region (e.g., leaf nodes), there is no need to evaluate the influence relationship in larger ones (i.e., parent or ancestor nodes), as evidenced by the following lemma. In other words, the former batch-wise avoids redundant computation, and the latter tightens the search space.

Lemma 4: Moving users who are influenced by an abstract facility in a small region are affected in the larger region that encloses the small one.

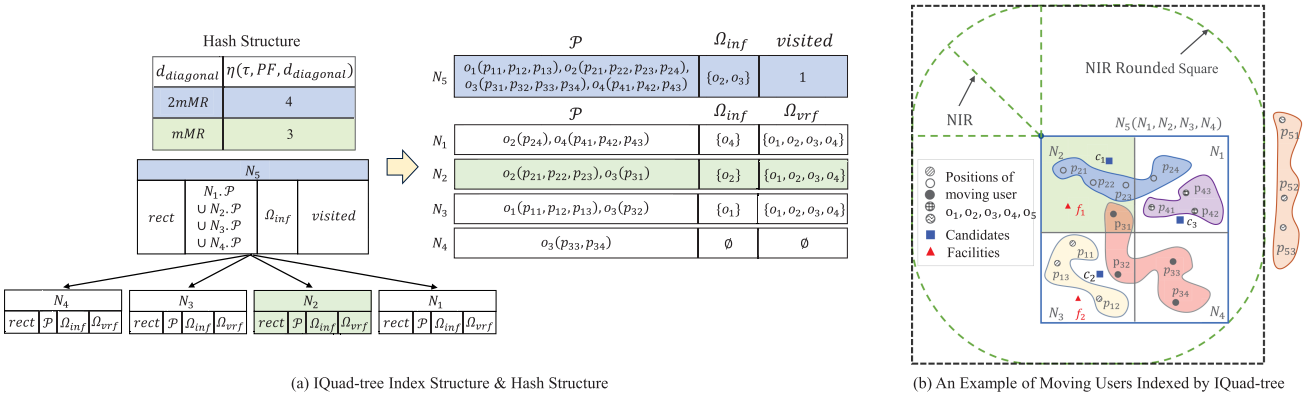
Proof: For a user $o = \{p_1, p_2, \ldots, p_i, p_{i+1}, \ldots, p_r\} \wedge |o| = r$, we assume o is influenced by an abstract facility v based on partial positions $\{p_1, p_2, \ldots, p_i\}$ which are located inside a small region. Let $(1 - Pr_v(p_1)) \cdot (1 - Pr_v(p_2)) \cdot \ldots \cdot (1 - Pr_v(p_i)) = Pr_v^{(1,i)}$, and we have $1 - Pr_v^{(1,i)} \geq \tau$ according to Definition 2. Suppose there is a larger region that encloses the small one, and a position $p_j \in \{p_{i+1}, \ldots, p_r\}$ is located outside the small region and inside the larger one. As $0 \leq (1 - Pr_v(p_j)) \leq 1$, we have $1 - Pr_v^{(1,i)} \cdot (1 - Pr_v(p_j)) \geq 1 - Pr_v^{(1,i)} \geq \tau$. Obviously, the inequality applies to larger regions containing more positions. Even if larger regions contain no more position, the conclusion still holds. ■

D. IQuad-Tree-Based Solution

In this section, we are ready to present the **IQ**uad-**T**ree-based (IQT) solution to MC²LS. The IQT solution comprises four phases: index-based pruning, verification, competitive influence computation and influenced users updating.

In the *pruning phase*, the IQuad-tree indexes all the users first. The pruning process of an abstract facility v begins at the leaf node in which it is located. According to the IS and NIR pruning strategies, we can retrieve the users inevitably influenced by v and those not. The process is then computed recursively from small to larger regions, checking the influence relationships of users in the tree until it reaches the root. There is no need to recalculate the batch of other abstract facilities in the same region as v. It is worth mentioning that both IS and NIR pruning rules are oriented to trim users, and the search space can be further tightened if we can eliminate unnecessary abstract facilities simultaneously. Fortunately, the IQuad-tree can seamlessly integrate IA and NIB pruning rules discussed in Section IV-B, as these two strategies are not coupled with the traversal process of the IQuad-tree. Thus reducing redundant abstract facilities can further improve the efficiency. In the *verification phase*, Definition 1 is employed to assess the remaining users and abstract facilities for which the influence relationships still need to be confirmed. The *third phase* calculates the competitive influence value for each candidate according to (1). The *updating phase* aims to handle the issue of overlapping influence. After a candidate c is selected into the final solution set, the users influenced by c are removed from the sets of attracted users of other candidates. The final result is obtained through k iterations of this phase. Algorithm 2 details the four-phase procedure.

Similar to Algorithm 1, we first initialize the result set G and two *key-value* sets F_o and Ω_v. An additional *key-value* set



(a) IQuad-tree Index Structure & Hash Structure

(b) An Example of Moving Users Indexed by IQuad-tree

Fig. 5. The IQuad-tree index structure and an example.

Algorithm 2: IQuad-Tree-Based (IQT) Algorithm.

Input: Ω, F, C, k, τ and $PF(\cdot)$
Output: the optimal set G of candidate locations

1 initialize $G := \emptyset$; $F_o := \emptyset$ ($\forall o \in \Omega$) and $\Omega_v := \emptyset$, $\Omega'_v := \emptyset$ ($\forall v \in C \cup F$);
2 construct IQuad-tree IQT of Ω and record NIR;
3 **foreach** $v \in C \cup F$ **do**
4 $\quad \langle \Omega_v, \Omega'_v, F_o \rangle := Traverse(IQT.root, v)$;
5 initialize $\Omega_v^{NIB} := \emptyset$ ($\forall v \in C \cup F$);
6 construct R-trees RT_C of C and RT_F of F;
7 **foreach** $o \in \Omega$ **do**
8 $\quad$ **foreach** $v \in RangeQuery(RT_C, NIB(o))$ **or** $v \in RangeQuery(RT_F, NIB(o))$ **do**
9 $\quad\quad$ **if** $o \notin \Omega_v$ **then**
10 $\quad\quad\quad \Omega_v^{NIB} := \Omega_v^{NIB} \cup \{o\}$;
11 **foreach** $v \in C \cup F$ **do**
12 $\quad \Omega'_v := \Omega'_v \cap \Omega_v^{NIB}$;
13 $\quad$ **foreach** $o \in \Omega'_v$ **do**
14 $\quad\quad$ **if** $Pr_v^{r-r'}(o) \leq 1 - \tau$ **then** // Early Stopping
15 $\quad\quad\quad \Omega_v := \Omega_v \cup \{o\}$;
16 $\quad\quad\quad$ **if** $v \in F$ **then**
17 $\quad\quad\quad\quad F_o := F_o \cup \{v\}$;
18 Execute lines 16-24 of Algorithm 1;
19 return G;

Ω'_v is used to hold users whose influence relationships between v need further validation. The IQuad-tree IQT indexes all the users, and the maximum number of positions, i.e., the NIR, is recorded accordingly (lines 1-2). We traverse IQT recursively to prune unnecessary calculations, and the three key-value sets are updated (line 4, detailed in the following paragraph). As discussed above, the IQuad-tree can be integrated with the IA and NIB pruning rules. Based on two R-trees of C and F, it utilizes the NIB rule to hold potentially attracted users, denoted by a *key-value* set Ω_v^{NIB}, where the key is v and the value corresponds to users requiring verification (lines 5-12). As NIR and NIB pruning strategies trim uninfluenced relationships between users and abstract facilities from different aspects, we only need to check the intersection of users retrieved by them (line 12). Two points need to be noted: first, the NIB rule may not always have an effect and hardly has an impact in scenarios where the spatial density of users' positions is low; second, the algorithm excludes the IA rule since the IS strategy almost overrides its effect. We employ *Early Stopping Strategy* proposed by study [13] to accelerate the computation of Definition 1 (line 14). Next, Ω_v and F_o are updated (lines 15-17). Finally, it executes the greedy iteration as Algorithm 1 and obtains the result (lines 18-19).

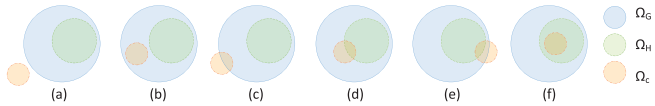


Fig. 6. Cases when adding $\{c\}$ to G or H.

Algorithm 3 outlines the process of the IQuad-tree traversal. For non-leaf nodes, we recursively call function $Traverse()$ with their child nodes that enclose a given abstract facility (lines 1-3). For the current node N, we first check whether it has been evaluated via any abstract facility before (lines 4,5 or lines 4,11). If not, we calculate the position count threshold or directly retrieve it from the *Hash* structure according to $N.rect$'s diagonal (lines 5-6,12-13). Utilizing the IS pruning rule, users necessarily influenced by abstract facilities in N are obtained and put into Ω_{IS} (lines 8,14). Since the NIR strategy only needs to be performed at leaf nodes, we use $RangeQuery()$ to achieve the set Ω_{vrf} of users that need further verification, which is also the initial Ω'_v of v (lines 9-10). Finally, we update sets Ω_v, Ω'_v and F_o (lines 16-19).

Theorem 2: The IQT algorithm can achieve $(1 - \frac{1}{e})$-approximation ratio to the MC^{2}LS problem.

Proof: According to Definition 6, for $G \subseteq C$ we have $cinf(G) \geq 0$, which means $cinf(\cdot)$ is a *non-negative* function. Given a candidate $c \in C \setminus G$, $cinf(\{c\}) \geq 0$ holds based on Definition 4. There are three cases for $cinf(G \cup \{c\})$: 1) $\Omega_G \cap \Omega_c = \emptyset$, then $cinf(G \cup \{c\})$ has the maximal value $cinf(G) + cinf(\{c\}) \geq cinf(G)$; 2) $\Omega_c \subset \Omega_G$, then $cinf(G \cup \{c\}) = cinf(G)$; 3) $\Omega_G \cap \Omega_c \neq \emptyset \wedge \Omega_c \not\subset \Omega_G$, we have $cinf(G) + cinf(\{c\}) > cinf(G \cup \{c\}) > cinf(G)$. Hence, $cinf(\cdot)$ is *monotone*. Let $H \subset G \subset C$, then $\Omega_H \subset \Omega_G$. When adding $\{c\}$ to G or H, we can evaluate $cinf(\cdot)$ by the set relationship, and all the six cases are illustrated in Fig. 6. For cases

Algorithm 3: $Traverse(N, v)$.

Input: node N, v
Output: $\langle \Omega_v, \Omega'_v, F_o \rangle$ if N is root node
if N *not leaf node* **then**
  foreach *child node of* N *encloses* v **do**
    $Traverse(child, v)$;
if $N.\Omega_{inf} = \emptyset$ **then**
  if N *is leaf node* ***and*** $N.\Omega_{vrf} = \emptyset$ **then**
    if $\eta(\tau, PF, Diagonal(N.rect)) \notin Hash$ **then**
      calculate $\eta(\tau, PF, Diagonal(N.rect))$;
    retrieve $N.\Omega_{inf}$ via IS pruning rule (Lemma 2);
    retrieve $N.\Omega_{vrf}$ via NIR pruning rule (Lemma 3);
    $\Omega'_v := N.\Omega_{vrf}$;
  if N *not leaf node* ***and*** $N.visited = 0$ **then**
    if $\eta(\tau, PF, Diagonal(N.rect)) \notin Hash$ **then**
      calculate $\eta(\tau, PF, Diagonal(N.rect))$;
    retrieve $N.\Omega_{inf}$ via IS pruning rule (Lemma 2);
    $N.visited := 1$; // for Non-leaf node
$\Omega_v := \Omega_v \cup N.\Omega_{inf}$;
foreach $o \in \Omega_v$ ***and*** $v \in F$ **do**
  $F_o := F_o \cup \{v\}$;
$\Omega'_v := \Omega'_v \setminus N.\Omega_{inf}$;

(a) and (f), $cinf(\{c\})$ does not impact $cinf(G \cup \{c\})$ and $cinf(H \cup \{c\})$, then $cinf(G \cup \{c\}) - cinf(G) = cinf(H \cup \{c\}) - cinf(H)$. For other cases, as Ω_c all brings more influenced users for Ω_H than for Ω_G, which means $cinf(G \cup \{c\}) - cinf(G) < cinf(H \cup \{c\}) - cinf(H)$. In summary, $cinf(G \cup \{c\}) - cinf(G) \leq cinf(H \cup \{c\}) - cinf(H)$, and thus $cinf(\cdot)$ satisfies the condition of a *submodular non-decreasing* function. As IQT algorithm is based on greedy heuristic, it guarantees a $(1 - \frac{1}{e})$-approximation ratio. ■

VI. Complexity Analysis

This section conducts theoretical study over all proposed methods: Baseline, Adapted k-CIFP and IQT.

The Baseline method greedily picks candidates based on Definition 7. The time complexity of computing the sets of influenced users for abstract facilities is $O((n+m)ur)$, where u, n and m are the cardinalities of Ω, C and F, r is the average number of positions of each moving user. To compute $cinf(c)$s and update Ω_cs for candidates, the complexity is $O(k \cdot 2n)$. Hence, the total complexity is $O((n+m)ur + 2kn)$.

For the Adapted k-CIFP algorithm, the study [13] proved in their Theorem 3 that the IA and NIB pruning rules can tighten the search space of abstract facilities[2] to $\lambda(m+n)$, where $\lambda \ll 1$. Hence, together with the cost of computing $cinf(c)$s and updating Ω_cs, the overall time complexity of Adapted k-CIFP is $O(\lambda(m+n)ur + 2kn)$.

Given a square of side length $2NIR + \frac{\sqrt{2}}{2}\hat{d}$, let random events X and Y denote that the square intersects a user's MBR and a user has at least one position falling in the square.

[2]This paper uses the parameter λ to represent the complicated form coefficient in [13].

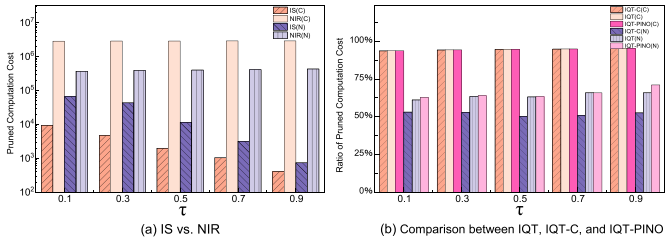


Fig. 7. Effect of pruning rules.

Let random event Z depicts a user has more than or equal to $\eta(\tau, PF, \hat{d})$ positions in a square of side length $\frac{\sqrt{2}}{2}\hat{d}$. The verification cost after the IS and NIR pruning rules is $O(1 - P(\bar{X}) - P(\bar{Y}|X) - P(Z|XY))ur)$, referred to as δur, where probabilities $P(\bar{X})$ and $P(\bar{Y}|X)$ are quite small (as discussed in Section VII). In addition, as the IQuad-tree is batch-wise, we probe only one abstract facility in each leaf node for pruning, thus the search space of abstract facilities $O(m+n)$ is ideally reduced to $O(\frac{2S}{\hat{d}^2})$, where S is the area of the whole spatial region. Typically, the pruning strategies cannot work on all users, so the actual effect $O(m'+n')$ falls between the ideal value $O(\frac{2S}{\hat{d}^2})$ and $O(m+n)$. Moreover, the early stopping strategy needs only partial $r' < r$ positions. Hence, the IQT algorithm has the time complexity of $O(\lambda(m'+n')\delta ur' + 2kn)$. Next, we examine the space complexity of IQuad-tree. We have $\frac{2S}{\hat{d}^2}$ squares as leaf nodes, and all $\frac{1}{3}(\frac{8S}{\hat{d}^2} - 1)$ nodes derived by summing geometric series. According to the form of IQuad-tree node, at most $3urP(Y)$ positions are stored in each node. Thus the space complexity is $O(\frac{8}{\hat{d}^2}P(Y)urS)$.

In summary, in the aspect of query efficiency, we have IQT > Adapted k-CIFP ≫ Baseline. Our experimental study will also validate the analysis.

VII. Performance Study

A. Experimental Setup

Datasets and Settings: We use two real-world datasets, California (abbr., C) [15] and New York (abbr., N),[3] in the experiments. They collected users' moving positions from *Gowalla* and *Brightkite* websites, respectively. The dataset C (*resp.*, N) contains 10,162 (*resp.*, 2,725) moving users and corresponding 381,165 (*resp.*, 34,024) positions in total, where users with only one position are trimmed. In C, each user's average number of positions per km^2 is approximately 80% of that in N. The ratio of the area of every user's MBR to that of the entire region averages about 0.085 in C, and it is 0.029 in N. These data indicate that users in C have wider spots of activities with a sparser distribution. Experiments are conducted following the probability function $PF(d(v,p)) = \rho/(1 + e^{d(v,p)})$ defined in [13], and the maximum probability parameter ρ is set to 1. In line with the previous study, we vary probability thresholds τ to 0.1, 0.3, 0.5, 0.7 and 0.9. For both candidates and existing facilities, we randomly choose 100,200,300,400 and 500 distinct locations from real points of interest in the two datasets. The diagonal $\hat{d}$

[3][Online]. Available: http://snap.stanford.edu/data/loc-Brightkite.html.

of the IQuad-tree leaf node ranges from 1 km to 2.5 km. Unless specified otherwise, the default values of $|\Omega|, |C|, |F|, k, \tau$ and $\hat{d}$ are 10,162 (C)/2,725 (N),100, 200, 10, 0.7 and 2 km, respectively. For all experiments, we compare the total time involving both indexing and querying.

Algorithms: Following algorithms are evaluated in the experiments. They are implemented in Python and tested on a 2.1GHz eight-core machine with 16GB RAM, running Linux.

- Baseline: The straightforward method discussed in Section IV-A that computes $cinf(c)$s for all abstract facilities and greedily selects k optimal candidates.
- k-CIFP: The solution adapted from study [15] and described in Algorithm 1.
- IQT: The IQuad-based solution described in Algorithm 2.
- IQT-C: A version of the IQT algorithm that excludes the classical NIB pruning rule [13].

B. Experimental Results

Effect of pruning rules: We first investigate the performance of the proposed IS and NIR pruning rules. As illustrated in Fig. 7(a), both the IS and NIR rules work effectively, where the square of an IQuad-tree node and the corresponding NIR rounded square extended by the NIR value determine their pruning effects, respectively. The former rule is relatively strict for the position count in a small area, while the latter prunes users in a larger spatial region due to the small area ratio of users' MBRs (about 3% to 8%). This is why the NIR pruning performs more prominently than the IS rule. Furthermore, according to (3), given a specific diagonal $\hat{d}$ of the IQaud-tree node, the $\eta(\tau, PF, \hat{d})$ value grows with the increase of τ, which results in a decrease in the performance of the IS pruning strategy; on the contrary, the NIR value declines as τ increases, which leads to better efficiency of the NIR rule.

The two pruning strategies perform differently on datasets C and N. As stated in Section VII-A, the activity regions of user moving positions are relatively smaller and denser in N compared to those in C. As the IS rule depends on the position count in a given area, the higher density of positions in N benefits satisfying the $\eta(\tau, PF, \hat{d})$ threshold, and so does the IS pruning effect. The side length of the NIR rounded square theoretically determines the NIR pruning effect, and their difference is only around 2 km in the two datasets. This implies that the uncovered area of the NIR rounded square is not the cause of the distinct pruning efficiencies in C and N.

In order to reveal the reason why the proportion of users pruned by the NIR rule is more than 90% in C but less than 40% in N, we further examine the datasets. Fig. 9(a) and (b) illustrate the two datasets, where gray, green and red dots indicate user positions, existing facilities and candidates, and the blue diamonds denote the k selected candidate results. Obviously, the distribution of users and abstract facilities is uniform in C, while it is highly skewed in N. It is more evident in the zoomed-in regions of Fig. 9(b), where some existing facilities even overlap. The phenomenon reflects the real-world situation that facilities are usually gathered in the regions where customers frequently

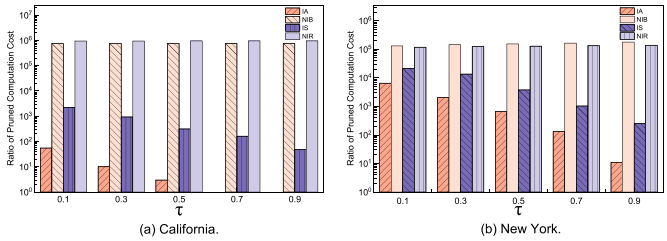


Fig. 8. Comparison with IA and NIB pruning rules.

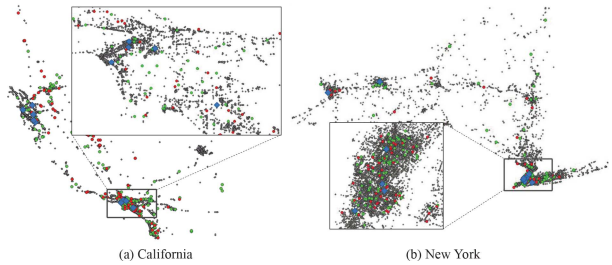


Fig. 9. The distribution of the two datasets.

appear. Due to the skewness, the number of users outside the NIR rounded square decreases, degrading the NIR pruning effect.

Next, we compare our pruning rules with the classical strategies for the multi-point model (i.e., IA and NIB as mentioned in Section IV-B). The IS and IA rules filter out objects (users or candidates, respectively) that are necessarily influenced. Fig. 8 reports that IS is more effective than IA. The reason is twofold. One is the batch-wise property of IS; the other is that the denser positions satisfy the $\eta(\tau, PF, \hat{d})$ condition more easily. The NIR and NIB rules exclude the unaffected objects. In C, the NIR rule can prune more than 20% of the computation cost than NIB. In N, however, NIB is slightly better than NIR (less than 10%). Batch-wise utility is degraded due to the skewed and concentrated distribution. NIB has the opposite effect. Smaller users' MBRs lead to larger areas of the external regions to prune. In N, the slightly inferior pruning effect of the NIR rule does not contradict its superior computational efficiency for two reasons. First, IA and NIB must conduct range query for each abstract facility, which is more costly than IQuad-tree traversal. Second, users with more positions are more likely to be trimmed by the IS and NIR rules. In contrast, k-CIFP cannot benefit after pruning since IA and NIB are irrelevant to position count.

The two sets of strategies offer different pruning perspectives: IA and NIB aim to prune abstract facilities, while IS and NIR focus on filtering out users. Thus, they can be combined for better efficiency. Fig. 7(b) compares the pruning effects of our proposed methods (IQT-C) with those of appending NIB to IQT-C (i.e., IQT) and appending both NIB and IA (referred to as IQT-PINO). In C, the pruning effects of the IS and NIR rules dominate, while NIB and IA struggle to provide utility only less than 1%. Conversely, for skewed and concentrated distribution in N, it can prune 8% to 15% computation and reduce time cost by 17% to 23% when incorporating NIB pruning. However, we observe that the pruning effects of IQT-PINO are very close to those of IQT. Only when $\tau = 0.9$ does IQT-PINO exhibit a

TABLE I
EXECUTION TIME COMPARISON WHEN CONSIDERING IA

# of abstract facilities	300	500	700	900	1.1k
IQT	276s	461s	646s	797s	976s
IQT-PINO	279s	469s	650s	809s	981s

TABLE II
EXECUTION TIME COMPARISON OF INDEX STRUCTURES

Dataset	IQuad-tree	IQT	R-tree	k-CIFP
C	47.70s	1,761.97s	0.04s	4,090.26s
N	3.88s	226.68s	0.04s	322.04s

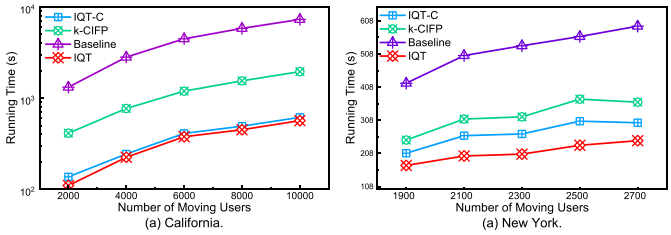


Fig. 10. Effect of $|\Omega|$.

slightly higher (about 5%) pruning gain than IQT. Accordingly, we compare the actual time costs of IQT-PINO and IQT under this setting, varying the number of abstract facilities from 300 to 1,100. Table I reports that the running time for IQT-PINO even exceeds that of IQT. This indicates that the time spent on executing range queries by the IA region outweighs its efficiency gains, making it unprofitable. Therefore, integrating only NIB is beneficial, especially when the dataset distribution is unknown.

Comparison of indexing time: This part compares the execution costs of the indexing structures employed in the IQT and k-CIFP algorithms, where IQuad-tree and R-tree index moving users and abstract facilities, respectively. As shown in Table II, compared to the time taken by R-tree to index 300 abstract facilities, the IQuad-tree appears to spend more time indexing users (more than 380k in C and 34k in N). However, if we compare the indexing time for a unit object, the IQuad-tree only requires 0.125ms (C) and 0.114ms (N), while the R-tree requires 0.133ms, indicating that the per-unit time cost is even lower for IQuad-tree. Meanwhile, the use of IQuad-tree brings significant performance improvements to the overall algorithm, making the incurred cost worthwhile.

Effect of $|\Omega|$*:* In this part, we study the scalability varying the cardinality of moving users. As illustrated in Fig. 10, the running time for all the algorithms grows with the increase of the number of users. The linear scanning-based Baseline is the least effective. The adapted k-CIFP method is designed based on the classical pruning strategies for multi-point probability model, IA and NIB, to reduce computation of unnecessary abstract facilities. As discussed in Section V, the limited number of facilities compared to users restrains the performance improvement of k-CIFP. From the user-pruning perspective, the IQT algorithm performs the best, reducing the running time by more than an order of magnitude compared to Baseline. Its superiority is due to the batch-wise operation on abstract facilities in IQuad-tree nodes. In N, the reduction in computation achieved by IQT is

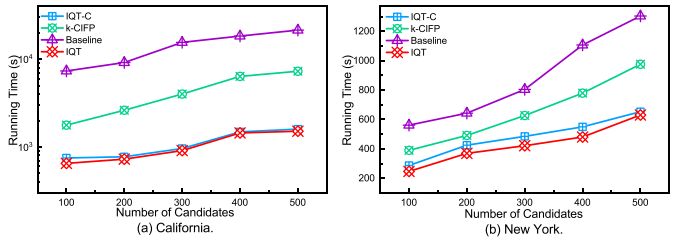


Fig. 11. Effect of $|C|$.

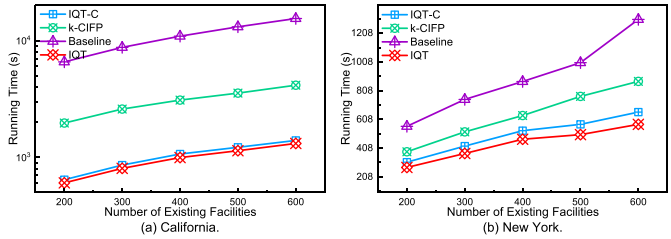


Fig. 12. Effect of $|F|$.

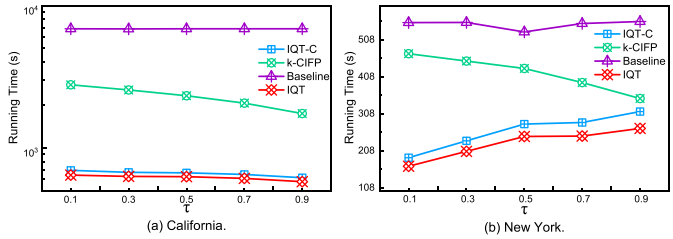


Fig. 13. Effect of τ.

not as significant as in C due to the degraded pruning effect discussed above. Nevertheless, IQT still improves efficiency by $30\% \sim 37\%$ over k-CIFP.

Effect of $|C|$*:* Fig. 11 demonstrates that the IQT algorithm remains the top performer and widens its lead as the number of candidates increases. Considering the experimental findings of the pruning rules above, the batch-wise effect of the IS rule is enhanced in scenarios with more concentrated candidates. In contrast, k-CIFP shows the opposite trend, decreasing performance as $|C|$ rises. The classical candidate-oriented pruning strategies IA and NIB suffer from the defect of being unable to batch process. This conclusion is illustrated well by the reverse trends of IQT-C and k-CIFP in N, where the gain of NIB pruning narrows with the increase of $|C|$.

Effect of $|F|$*:* Below, we test the performance when varying the number of existing facilities. As shown in Fig. 12, the results are qualitatively similar to the effect of $|C|$, where IQT exhibits the best efficiency, followed by IQT-C, k-CIFP and Baseline. Similar to the impact of $|C|$, the increase in $|F|$ weakens the effectiveness of the NIB rule. However, the change in performance with $|F|$ is relatively smooth. The reason mainly lies in the similar distribution of facilities and independent of their counts.

Effect of τ*:* This part discusses the impact of parameter τ. As shown in Fig. 13, since the complexity of Baseline is not directly affected by τ, its cost remains constant. k-CIFP exhibits

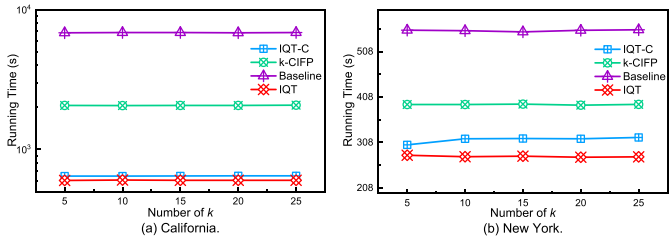


Fig. 14. Effect of k.

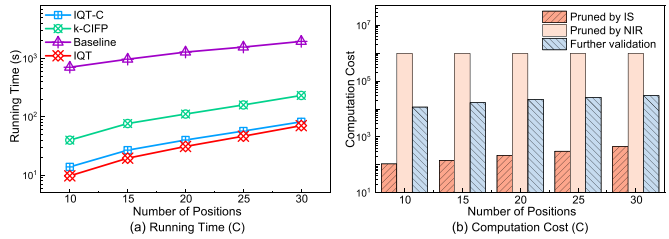


Fig. 15. Effect of r (in C).

a very significant upward trend due to the decline of $mMR(\tau, r)$ caused by the increase of τ, which enhances its pruning effect. Differing from the facility-oriented pruning of k-CIFP, IQT is affected by the data distribution, making the pruning more complex. Since the NIR value decreases with increasing τ, the relatively uniform distribution of data in C enhances the NIR rule. In N, the skewed and concentrated distribution simultaneously weakens the pruning effects of both IS and NIR rules, leading to performance degradation. Fortunately, even so, IQT still outperforms the other methods.

Effect of k: The results on different k values are shown in Fig. 14, where IQT still outperforms other algorithms. All the algorithms achieve identical k result candidates. For every algorithm, its running costs remain nearly constant regardless of k values, which means the time complexity of handling influence overlapping is negligible compared to that of evaluating competitive influence.

Effect of $\hat{d}$: We report the effect of the diagonal length $\hat{d}$ of the IQuad-tree leaf node. $\hat{d}$ hardly impacts on the pruning effectiveness, as $\hat{d}$ accounts for less than 1% of the entire region's scope and only about 15% of the side length of NIR rounded squares for both datasets. Due to space constraint, we omit the figure varying $\hat{d}$ for brevity. The time required to build the IQuad-tree is only 0.5% of the time cost of Baseline, highlighting the value of the index structure.

Effect of r: Finally, we examine the effect of varying the number of user positions r . To ensure consistency, we choose users with over 30 positions (3,336 in C and 233 in N) and randomly sample 10, 15, 20, 25, and 30 positions from these users. Regardless of r, the advantages of IQT are evident. As position density increases, the pruning effect of IS improves while that of NIR drops, yet NIR remains dominant. As shown in Fig. 15(a) (*resp.,* Fig. 16(a)), the execution time rises with more positions, correlating positively with the verification computation cost in Fig. 15(b) (*resp.,* Fig. 16(b)). This indicates that pruning reduces substantial computation costs with a minimal

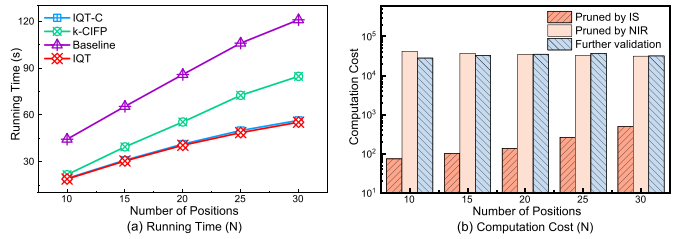


Fig. 16. Effect of r (in N).

overhead, demonstrating its effectiveness. Note that the very small number of eligible users in N, only 233, diminishes the effectiveness of the pruning strategies.

VIII. Conclusion

In this paper, we formalize a novel Collective location selection (CLS) problem called MC^{2}LS that considers, for the first time, peer competition and users' mobility factors simultaneously. We prove the MC^{2}LS problem is NP-hard and propose a greedy method adapted from the candidate-pruning technique. To overcome the challenge that highly overlapped activity areas of moving users make existing techniques for pruning users unavailable, we exploit the relationship between influence probability and the user's position count to design two square-based pruning rules. Then, we present a user-MBR-free index, IQuad-tree, to benefit the square hierarchy property. Based on IQuad-tree, we propose an $(1 - \frac{1}{e})$-approximate solution to MC^{2}LS. The theoretical analysis and experimental results show the efficiency, effectiveness and scalability of our proposed approaches. As part of future work, we plan to study extended solution towards MC^{2}LS in social network scenarios, incorporating social influence and users' interests.

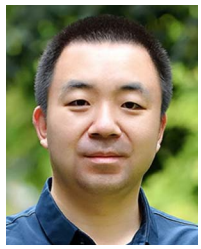
Acknowledgment

The authors would like to thank the editor and anonymous reviewers for their constructive comments on this paper.

References

[1] X. Ding, L. Chen, Y. Gao, C. S. Jensen, and H. Bao, "UlTraMan: A unified platform for big trajectory data management and analytics," in *Proc. VLDB Endowment*, vol. 11, no. 7, 2018, pp. 787–799.
[2] F. Chen, H. Lin, J. Qi, P. Li, and Y. Gao, "Collective-k optimal location selection," in *Proc. Int. Symp. Spatial Temporal Databases*, 2017, pp. 339–356.
[3] Z. Qi, H. Wang, T. He, C. Wang, J. Li, and H. Gao, "TAILOR: Time-aware facility location recommendation based on massive trajectories," *Knowl. Inf. Syst.*, vol. 62, no. 9, pp. 3783–3810, 2020.
[4] W. Shan, Q. Yan, C. Chen, M. Zhang, B. Yao, and X. Fu, "Optimization of competitive facility location for chain stores," *Ann. Operations Res.*, vol. 273, no. 1/2, pp. 187–205, 2019.
[5] Y. Li, J. Bao, Y. Li, Y. Wu, Z. Gong, and Y. Zheng, "Mining the most influential k-location set from massive trajectories," *IEEE Trans. Big Data*, vol. 4, no. 4, pp. 556–570, Dec. 2018.
[6] A. M. Mir et al., "Data driven smart policing: A novel road distance-based *k*-median model for optimal substation placement," *Comput. Hum. Behav.*, vol. 127, 2022, Art. no. 107014.
[7] C. Chen, J. Liu, Q. Li, Y. Wang, H. Xiong, and S. Wu, "Warehouse site selection for online retailers in inter-connected warehouse networks," in *Proc. IEEE Int. Conf. Data Mining*, 2017, pp. 805–810.

[8] A. Saha, D. Pamucar, Ö. F. Görçün, and A. R. Mishra, "Warehouse site selection for the automotive industry using a Fermatean fuzzy-based decision-making approach," *Expert Syst. Appl.*, vol. 211, 2023, Art. no. 118497.
[9] P. Zhang, Z. Bao, Y. Li, G. Li, Y. Zhang, and Z. Peng, "Trajectory-driven influential billboard placement," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2748–2757.
[10] P. Zhang, Z. Bao, Y. Li, G. Li, Y. Zhang, and Z. Peng, "Towards an optimal outdoor advertising placement: When a budget constraint meets moving trajectories," *ACM Trans. Knowl. Discov. Data*, vol. 14, no. 5, pp. 51:1–51:32, 2020.
[11] Z. Chen, Y. Liu, R. C. Wong, J. Xiong, G. Mai, and C. Long, "Efficient algorithms for optimal location queries in road networks," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2014, pp. 123–134.
[12] Y. Gao, S. Qi, L. Chen, B. Zheng, and X. Li, "On efficient k-optimal-location-selection query processing in metric spaces," *Inf. Sci.*, vol. 298, pp. 98–117, 2015.
[13] M. Wang, H. Li, J. Cui, K. Deng, S. S. Bhowmick, and Z. Dong, "PINOC-CHIO: Probabilistic influence-based location selection over moving objects," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 11, pp. 3068–3082, Nov. 2016.
[14] R. Aboolian, O. Berman, and D. Krass, "Competitive facility location model with concave demand," *Eur. J. Oper. Res.*, vol. 181, no. 2, pp. 598–619, 2007.
[15] D. Li, H. Li, M. Wang, and J. Cui, "k-collective influential facility placement over moving object," in *Proc. IEEE 20th Int. Conf. Mobile Data Manage.*, 2019, pp. 191–200.
[16] M. E. Ali, S. S. Eusuf, K. Abdullah, F. M. Choudhury, J. S. Culpepper, and T. Sellis, "The maximum trajectory coverage query in spatial databases," *Proc. VLDB Endowment*, vol. 12, no. 3, pp. 197–209, 2018.
[17] P. Liu, M. Wang, J. Cui, and H. Li, "Top-k competitive location selection over moving objects," *Data Sci. Eng.*, vol. 6, no. 4, pp. 392–401, 2021.
[18] F. Plastria, "Static competitive facility location: An overview of optimisation approaches," *Eur. J. Oper. Res.*, vol. 129, no. 3, pp. 461–470, 2001.
[19] Q. Zeng, M. Zhong, Y. Zhu, and J. Li, "Business location selection based on geo-social networks," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2020, pp. 36–52.
[20] L. Kung and W. Liao, "An approximation algorithm for a competitive facility location problem with network effects," *Eur. J. Oper. Res.*, vol. 267, no. 1, pp. 176–186, 2018.
[21] J. Chen et al., "Towards efficient MIT query in trajectory data," in *Proc. IEEE Int. Conf. Data Eng.*, 2023, pp. 2194–2206.
[22] R. A. Finkel and J. L. Bentley, "Quad trees: A data structure for retrieval on composite keys," *Acta Informatica*, vol. 4, pp. 1–9, 1974.
[23] C. Revelle, "The maximum capture or sphere of influence location problem: Hotelling revisited on a network," *J. Regional Sci.*, vol. 26, no. 2, pp. 343–358, 1986.
[24] C. Xu, Y. Gu, R. Zimmermann, S. Lin, and G. Yu, "Group location selection queries over uncertain objects," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 12, pp. 2796–2808, Dec. 2013.
[25] S. Wang, Y. Zhang, X. Lin, and M. A. Cheema, "Maximize spatial influence of facility bundle considering reverse k nearest neighbors," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2018, pp. 684–700, collective.
[26] T. Cai, J. Li, A. Mian, R. Li, T. Sellis, and J. X. Yu, "Target-aware holistic influence maximization in spatial social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 4, pp. 1993–2007, Apr. 2022.
[27] H. Wang, H. Li, M. Wang, and J. Cui, "Toward balancing the efficiency and effectiveness in k-facility relocation problem," *ACM Trans. Intell. Syst. Technol.*, vol. 14, no. 3, pp. 52:1–52:24, 2023.
[28] W. Ning, X. Yan, and B. Tang, "Towards efficient MaxBRNN computation for streaming updates," in *Proc. IEEE Int. Conf. Data Eng.*, 2021, pp. 2297–2302.
[29] B. Liu, Y. Fu, Z. Yao, and H. Xiong, "Learning geographical preferences for point-of-interest recommendation," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2013, pp. 1043–1051.
[30] Y. Zhang, Y. Li, Z. Bao, S. Mo, and P. Zhang, "Optimizing impression counts for outdoor advertising," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1205–1215.
[31] D. Chakrabarty, M. Negahbani, and A. Sarkar, "Approximation algorithms for continuous clustering and facility location problems," in *Proc. Eur. Symp. Algorithms*, 2022, pp. 33:1–33:15.
[32] L. Wang, Z. Yu, D. Yang, H. Ma, and H. Sheng, "Efficiently targeted billboard advertising using crowdsensing vehicle trajectory data," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1058–1066, Feb. 2020.
[33] Q. Zeng, M. Zhong, Y. Zhu, T. Qian, and J. Li, "Business location planning based on a novel geo-social influence diffusion model," *Inf. Sci.*, vol. 559, pp. 61–74, 2021.
[34] T.-K. Wang, X. Gu, K. Li, and J.-H. Chen, "Competitive location selection of a commercial center based on the vitality of commercial districts and residential emotion," *J. Urban Plan. Develop.*, vol. 147, no. 1, 2021, Art. no. 04021001.
[35] T. Drezner, Z. Drezner, and P. J. Kalczynski, "A leader-follower model for discrete competitive facility location," *Comput. Operations Res.*, vol. 64, pp. 51–59, 2015.
[36] M. Qi, R. Jiang, and S. Shen, "Sequential competitive facility location: Exact and approximate algorithms," *Operations Res.*, vol. 72, pp. 300–316, 2022.
[37] D. S. Hochbaum and A. Pathria, "Analysis of the greedy approach in problems of maximum k-coverage," *Nav. Res. Logistics*, vol. 45, no. 6, pp. 615–627, 1998.
[38] A. Guttman, "R-trees: A dynamic index structure for spatial searching," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 1984, pp. 47–57.



Meng Wang received the MS degree from Xi'an Jiaotong University, China, in 2012, and the PhD degree from Xidian University, China, in 2018. He is an associate professor with the School of Computer Science, Xi'an Polytechnic University, China. His research interests include data mining, spatio-temporal data management, and time series analysis. He has published papers in major venues in these areas, such as ICDE, the *IEEE Transactions on Knowledge and Data Engineering*, CIKM, etc.

Mengfei Zhao received the BE degree from Xuchang University, China, in 2021. She is currently working toward the MS degree with the School of Computer Science and Technology, Xi'an Polytechnic University, China. Her research interest include spatio-temporal data management.

Hui Li is currently a professor with the School of Computer Science and Technology, Xidian University, Xi'an, China. His research interests include data mining, database, and privacy-preserving query. He has published many papers in major venues in these areas, such as SIGMOD, VLDB, ICDE, KDD, the *IEEE Transactions on Knowledge and Data Engineering*, *VLDB Journal*, etc. He has been nominated as the best paper award in SIGMOD.

Jiangtao Cui received the MS and PhD degrees in computer science from Xidian University, Xi'an, China, in 2001 and 2005, respectively. Between 2007 and 2008, he has been with the Data and Knowledge Engineering Group working on high-dimensional indexing for large scale image retrieval, with the University of Queensland, Australia. He is currently a professor with the School of Computer Science and Technology, Xidian University, China. His current research interests include data and knowledge engineering, data security, and high-dimensional indexing.

Bo Yang received the PhD degree in computer science and technology from Xi'an Jiaotong University, China, in 2017. He was a postdoctoral fellow with Northwestern Polytechnical University, China, and with the University of Toronto, Canada, from 2017 to 2020 and from 2021 to 2022. He has been an associate professor with Xi'an Polytechnic University, China. His current research interests include machine learning, data mining, and bioinformatics.

Tao Xue received the BS degree in mechanical engineering from Xi'an Jiaotong University (XJTU), China, in 1995, the MS degree in computer science from Northwest University, China, in 2000, and the PhD degree in computer software and theory from the XJTU, in 2005. He is currently a professor with the School of Computer Science, Xi'an Polytechnic University, China. His current research interests include knowledge graph, large language model, computer vision, and Big Data.