



Top- k Competitive Location Selection over Moving Objects

Ping Liu¹ · Meng Wang² · Jiangtao Cui¹ · Hui Li³

Received: 15 October 2020 / Revised: 22 February 2021 / Accepted: 17 April 2021 / Published online: 31 May 2021
© The Author(s) 2021

Abstract

The location selection (LS) problem identifies an optimal site to place a new facility such that its influence on given objects can be maximized. With the proliferation of GPS-enabled mobile devices, LS studies have made progress for moving objects. However, the state-of-the-art LS techniques over moving objects assume the new facility has no competitor, which is too restrictive and unrealistic for real-world business. In this paper we study Competitive Location Selection over Moving objects (CLS-M), which takes into account competition against existing facilities in mobile scenarios. We present a competition-based influence score model to evaluate the influence of a candidate. To solve the problem, we propose an influence pruning algorithm to prune objects who are either influenced by inferior candidates or affected by no candidate. Experimental study over two real-world datasets demonstrates that the proposed algorithm outperforms state-of-the-art LS techniques in terms of efficiency.

Keywords Competitive location selection · Moving objects · Pruning strategy · Spatial data

1 Introduction

Location selection (LS) problems have been extensively studied in spatial databases. Given a set of objects Ω and a set of candidate locations C , the LS problem aims to find an optimal candidate location $c \in C$, such that c can influence a maximum number of objects. LS problems are widely used in many fields such as marketing [11], urban planning [18], monitoring wildlife [6], scientific research.

Ping Liu and Meng Wang contributed equally to this work and should be considered co-first authors.

✉ Ping Liu
pingxiao_liu@163.com

Meng Wang
wamengit@sina.com

Jiangtao Cui
cuijt@xidian.edu.cn

Hui Li
hli@xidian.edu.cn

¹ School of Computer Science and Technology, Xidian University, Xi'an, China

² School of Computer Science, Xi'an Polytechnic University, Xi'an, China

³ School of Cyber Engineering, Xidian University, Xi'an, China

With the proliferation of GPS-enabled mobile devices, the location data can be easily collected. This enables us to consider the mobility of objects in the LS problems [10, 11]. Specifically, the authors in [11] investigate a generalized LS problem called PRIME-LS in which the influence between a facility and a moving object is modeled by a probabilistic relationship, instead of the traditional deterministic binary criterion (i.e., either influence or not). The probabilistic feature even makes PRIME-LS coincide with a common phenomenon that an object can be influenced by multiple facilities simultaneously. However, the PRIME-LS does not involve competition factor, which significantly limits its applications in the real world.

Consider that someone plans to open a convenience store and needs to choose an optimal location for it hoping that it can attract the maximum number of potential customers. Taking Fig. 1a as an example, there are two moving objects O_1, O_2 and two candidates c_1, c_2 . According to PRIME-LS, c_2 is chosen as the optimal answer. Unfortunately, in many real-world scenarios, there are existing facilities of the same type (e.g., *7-eleven*, *FamilyMart*) which makes it a competitive market instead of an ideal condition without competitor.¹ As illustrated in Fig. 1b, there are two existing facilities near candidate c_2 , while there is no competitor around c_1 . In

¹ In this paper, we assume that all the facilities, including existing ones and the top- k candidate ones to be built, have the same type.

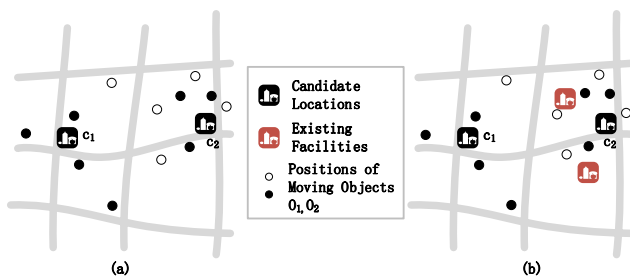


Fig. 1 Location model

that case, *how can we choose the optimal result to gain better economic benefits?*

Recall that studies [3, 4] considered the impact of competition among existing facilities nearby in the LS problem. However, their influence model is based on Bichromatic Reverse Nearest Neighbor (BRNN) criterion [5] and static single-point objects without considering the mobility. Hence, their competition-based techniques are unsuitable for solving the aforementioned problem.

To address the limitations of existing LS techniques, we study the competition-based LS problem in moving scenarios, called Competitive Location Selection over Moving objects (CLS-M), which takes into account both mobility and competition. We face the following two challenges: (1) It is not straightforward to evaluate and model the impact of existing facilities on candidates in moving scenarios. (2) The large amount of positions will incur substantial overhead for the evaluation of influence.

In this paper, we define the competitive influence based on two key concepts inspired by existing works. On the one hand, in the influence relationship model in [11], moving objects can be affected by multiple facilities simultaneously, which implies we can identify which facilities (or/and candidates) will join the competition for a specific moving object. On the other hand, according to the competition model in [8, 9], if n facilities all capture (i.e., influence) the same object, the capture is divided into n equal parts. Hence, incorporating the two above aspects, we design a competition-based influence score model, which is detailed in Sect. 3.

To solve the problem efficiently, we propose an Influence Pruning Algorithm (IPA) which prune objects who are either influenced by inferior candidates or not affected by any candidate. Experiments show that IPA is superior in efficiency and it is at least one order of magnitude better than the baseline algorithm.

The contributions of this paper can be summarized as follows:

- We introduce a more practical location selection task, namely CLS-M, which takes both mobility and competition factors into consideration.
- We propose an efficient algorithm called IPA to solve the proposed problem. Two pruning strategies are designed to reduce the computational complexity.
- Comprehensive experiments are conducted on real-world datasets from two cities. The results demonstrate that, comparing to the baseline and the state-of-the-art algorithm in [11], our proposed solution significantly improves the efficiency.

The rest of the paper is organized as follows. In Sect. 2, the related works are reviewed. In Sect. 3, we give out the definition of CLS-M. In Sect. 4, we present our solution. In Sect. 5, the result of those experiments is reported. We conclude this paper in Sect. 6.

2 Related Works

In this section, we discuss related efforts in LS problems under various scenarios.

One direction is the maximum influence-based LS (Max-inf). In Max-inf problems, *influence* refers to the number (or probability) of objects (e.g., persons, vehicles) that may visit (or be affected by) a particular location if some facility is placed there. Most of these studies assume that an object's location is a static single point and only one facility will exhibit influence on it exclusively. The Max-inf-based LS problem is closely related to the BRNN concept [5]. Specifically, Max-inf LS aims to find a location with the maximum influence. Xia et al. [13] defined the influence of a location as the total weight of its RNNs (reverse nearest neighbors) and developed a distance metric, called minExistDNN, to prune search space based on R-tree. Yan et al. [14] relaxed the assumption from NN facility to $(1 + \alpha)NN$, where α was a user-specified value. Wong et al. [12] studied a similar problem, called MaxBRkNN, in which all the kNN facilities exhibited influence on objects. Yiu et al. [15] further extended the LS problem, in which they focused on the total distance-weighted qualities of surrounding facilities of query locations. In these studies, the object location is a certain single point, which is not consistent with reality.

Cheema et al. [1] studied the problem of probabilistic reverse nearest neighbor based on the possible-world semantics. Following the same setting, Zhan et al. [16] aimed to find top- k most influential facilities over uncertain objects. Zheng et al. [19] proposed a partition-based algorithm and many pruning techniques to solve a similar problem. Although these studies modeled an object as multiple position instances, the multiple positions are very near around the actual position of the object, while locations of a moving object cover a large

geographic area. The wide-range region of a moving object leads to overlapping with other objects, which makes pruning techniques and location selection approaches designed for uncertain model are not available for our problem. Besides, in a possible world, each object was still represented by a single position and was limited to be influenced by only one facility based on NN metric.

Wang et al. [11] introduced a generalized LS problem called PRIME-LS, which utilized mobility and probability factors. The authors presented a criterion that used cumulative probability for all positions along the moving object to judge the impact. Compared with uncertain model [1], the experiment in [11] proved that the moving object model was better in both efficiency and effectiveness. As it is more relevant to real scenarios, we will adopt that criterion to judge whether a candidate $c \in C$ influences a moving object or not. Zhang et al. [17] study a maximum coverage-based LS problem, which finds a set of facilities that can cover the maximum number of trajectories. However, the competitive impact of existing facilities in these studies was not considered.

Traditional competitive location selection problems [7] assumed that a firm entered in a market where some existing firms had been operating, aiming at choosing the optimal location to attract the maximum market share under competition. In these problems, the spatial location information of competitors was obtained in advance. Studies [8, 9] introduced a new type of competitive LS problem. It added new facilities to the existing ones so that the new facilities had the maximum influence. They assumed if two facilities captured the same object, they equally shared the influence on it. This model can be used to evaluate the situation that multiple facilities affect the same object, and thus, it can be a basis of our competitive influence model.

Huang et al. [3, 4] considered the impact of existing facilities on location selection under the Max-inf-based model. The definition of influence relationship was based on nearest neighbor. It pruned the computation by establishing a minimum facility circle for objects to find candidate locations with the greatest influence. Because they all assume that the object is a static single point. The solution of this problem is not applicable to the mobile scenarios. Therefore, in Sect. 3, we build a more general model for the competitive relationship in moving scenarios.

3 Problem Definition

In this section, we first introduces some preliminaries about the competition-based influence score and then formulate the CLS-M problem.

3.1 Preliminary

A location p is a point in a two-dimensional Euclidean space, denoted by its geographical coordinate (i.e., latitude and longitude). Given two locations p_1 and p_2 , the distance between them is denoted by $\text{dist}(p_1, p_2)$. In this paper, we use a set of discrete positions $O = \{p_1, p_2, \dots, p_r\}$ to represent a moving object. We denote candidate locations for new facilities to deploy as $C = \{c_1, c_2, \dots, c_n\}$ and existing facilities as $F = \{f_1, f_2, \dots, f_m\}$. The probability that an object at location p is influenced by a facility $v \in C \cup F$ is denoted by $Pr_v(p)$. Following the setting of distance-based probability function $PF(\cdot)$ in [11], $Pr_v(p)$ can be computed as $Pr_v(p) = PF(\text{dist}(v, p))$. To put it another way, a moving object O is influenced by v if and only if there is at least a position p_i of O that is influenced by v . Assume that, the probability that O is influenced by a facility v at any position $p_i \in O (i \in [1, r])$ is independent of those at other positions; we have $Pr_v(p_i) = PF(\text{dist}(v, p_i))$. Considering all the positions of object O , the *cumulative probability* that O is influenced by v is defined as $Pr_v(O) = 1 - \prod_{i=1}^r (1 - Pr_v(p_i))$.

Definition 1 (*Influence value*): Given a moving object O and a probability threshold τ , candidate c (resp., facility f) can influence O if and only if $Pr_c(O) > \tau$ (resp., $Pr_f(O) > \tau$). Further, given a set Ω of moving objects, the **influence value** of c (resp., f), denoted as $\text{inf}(c)$ (resp., $\text{inf}(f)$), is the number of moving objects in Ω that are influenced by c (resp., f).

According to Definition 1, the influence value of f or c indicates the maximum number of moving objects which might be influenced by f or c under the constraint of a user-specified probability condition.

3.2 Competition-based Influence Score

In this section, we take into account existing facility competitors against the new sites to design a novel competitive influence relationship over moving objects, which is based on two concepts from existing studies.

On the one hand, according to the competition model in [8, 9], if two facilities are located in equal distance from an object based on BRNN, it is regarded that they both capture the object and their capture is divided into equal parts. In other words, if some facilities capture the same object, they equally share the influence on it.

On the other hand, the PRIME-LS model for moving objects [11] assumes each object can be affected by multiple facilities simultaneously. Specifically, for a moving object, if the influence values of multiple facilities are all beyond the given probability threshold, the facilities will all exhibit influence on the object.

Incorporating the two aforementioned aspects, we design a competition-based influence score model which extends the competition concept in static LS problems to moving scenarios. To facilitate exposition and understanding, we first introduce some necessary notations.

We denote $\sigma(c)$ as the set of objects that are influenced by candidate c , and $\sigma(c)$ can be formalized as $\sigma(c) = \{O \mid Pr_c(O) \geq \tau, c \in C, O \in \Omega\}$.

Similarly, a set of facilities that influence moving object O , denoted by $\sigma^{-1}(O)$, can be defined as $\sigma^{-1}(O) = \{f \mid Pr_f(O) \geq \tau, f \in F, O \in \Omega\}$.

If an object is influenced by a facility or candidate, we call there is an **influence relationship** between them. The above formulas indicate that, for a candidate, it is required to consider not only the objects influenced by itself, but also the existing facilities which have influence relationships with some of the influenced objects. At this time, there will be a competitive relationship between candidates and the existing facilities for objects that they both influence. *Then how do we quantify the competition-based influence score?*

Intuitively, we have the two following observations that have impact on the influence score. First, the more objects are influenced by c , the higher score c might achieve. Second, the less competitors (i.e., existing facilities) which scramble the same objects influenced by c also raise the score. Accordingly, we define the influence score as follows.

Definition 2 (*Influence score*): Given a set $\sigma(c)$ of moving objects influenced by c and a set $\sigma^{-1}(O_i)$ of existing facilities that influence O_i , where $O_i \in \sigma(c)$, the **competition-based influence score** for candidate c can be described as follows:

$$score(c) = \sum_{O_i \in \sigma(c)} \frac{1}{|\sigma^{-1}(O_i)| + 1}. \quad (1)$$

For an object O_i that is influenced by candidate c , the fraction $\frac{1}{|\sigma^{-1}(O_i)| + 1}$ means that c and all the existing facilities, which influence O_i , equally share the influence on O_i . In other words, the influence probability on O_i is equally split by c and the facilities. Thus $score(c)$ indicates the sum of the influence probabilities each of which corresponds to an object influenced by c .

Example 1 We assume that candidates c_1, c_2, c_3 , facilities f_1, f_2 and moving objects O_1, \dots, O_4 have the influence relationships shown in Table 1, which follow the cumulative probability influence criterion. Supposing that we intend

Table 1 An example of influence relationships

Candidate	Existing facilities	Influenced objects
c_1	f_1, f_2	O_1, O_2
c_2	f_2	O_3
c_3		O_4

to select a candidate to place the new facility, c_1 will be the optimal result by directly applying PINOCCHIO-VO [11]. Unfortunately, two competitors f_1, f_2 will share the influence, and thus $score(c_1) = 0.6$. Hence, if we take into account the competition from existing facilities, c_3 has the highest influence score, i.e., $score(c_3) = 1$, and c_3 is picked as the optimum.

Notably, facilities may have different ratings, which are usually based on comprehensive evaluation, e.g., service qualities, price, environment, etc. In that case, our competitive influence model can be easily adapted and applied. Specifically, by normalizing the ratings of facilities (or/and candidates) which influence the same object, facilities will capture different influence probabilities in proportion to their ratings.

3.3 Problem Definition

We are now ready to define the top-k Competitive Location Selection over Moving objects (CLS-M) problem addressed in this paper. We return top-k results based on which further decisions can be made for other factors, such as rental [2].

Definition 3 (*CLS-M*): Given a set of candidate locations C , a set of existing facilities F , a set of moving objects Ω each of which has a series of positions $\{p_1, p_2, \dots, p_r\}$, and a user-specified number k ($k \leq |C| \wedge k \in \mathbb{Z}$), the **CLS-M** problem aims to mine a subset $C' \subset C \wedge |C'| = k$, such that for each $c_i \in C'$, we have $score(c_i) \geq score(c_j)$ where $c_j \in (C - C')$.

4 Solution to CLS-M

According to Definitions 1 and 2, a straightforward solution to the CLS-M problem is to exhaustively check all candidates. Specifically, for each candidate c , we compute the

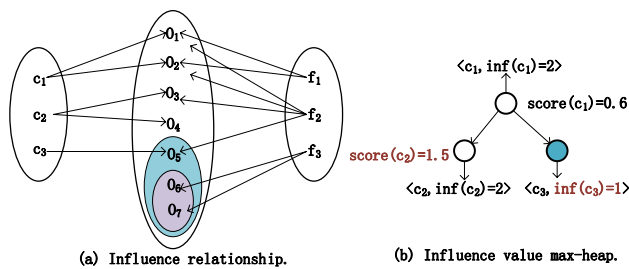


Fig. 2 An illustration of pruning rules

cumulative influence probabilities over moving objects to derive $\sigma(c)$ and further to obtain the influence relationships between objects and existing facilities. Then, we compute influence score for every candidate and the top- k ones with superior influence scores are the optimal answer. Although PINOCCHIO-VO [11] can be used to reduce the computational complexity for evaluating influence probabilities between object-facility pairs, it is still very costly to calculate all the influence relationships of objects that are related to all the candidates.

4.1 Pruning Rules

We notice that not all objects are influenced by candidates. Then before computing the influence score, the objects that are not influenced by candidates can be pruned to avoid the influence relationship calculation with existing facilities. We call it ***influence relationship pruning rule***. In Fig. 2a, each arrow indicates an influence relationship from a candidate/facility to an object. To solve CLS-M, we firstly need to identify $\sigma(c)$ s for candidates. Then it is required to compute the influence relationships between objects in $\sigma(c)$ s and the corresponding existing facilities. With the help of the above pruning rule, the influence relationship calculations for O_6 and O_7 with the corresponding facilities are avoided, as neither of them is influenced by any candidate. Furthermore, we can reduce the computation according to the following theorem.

Theorem 1 For $\forall c \in C$, we have $score(c) \leq inf(c)$.

Proof As shown in Equation (1), for each $O_i \in \sigma(c)$, we have $|\sigma^{-1}(O_i)| \geq 0$. The equal sign, as well as the maximum value of $\frac{1}{|\sigma^{-1}(O_i)|+1}$, holds when there is no competitor, i.e., O_i is

exclusively influenced by c . It means if every $O_i \in \sigma(c)$ is not influenced by any existing facility, c obtains the maximum influence score, i.e., $score(c) \leq inf(c)$.

According to Theorem 1, the influence value of c is the upper bound of $score(c)$, which can be used for pruning inferior candidates. Ordered by influence score, the k -th largest score of candidates can be used as a threshold. Once the upper bound of a candidate is below the threshold, other candidates with less upper bounds do not need to compute the exact influence score. We call it ***influence value pruning rule***.

The pruning strategy can be implemented using a max-heap sorted by $inf(c)$. As illustrated in Fig. 2b, $score(c_1)$ and $score(c_2)$ are derived and the current maximum score is $score(c_2) = 1.5$, which is set to be the current threshold. For candidate c_3 , its upper bound of influence score is $max(score(c_3)) = inf(c_3) = 1 < score(c_2) = 1.5$, which means the calculation of $score(c_3)$ is redundant. Therefore, as shown in Fig. 2a, there is no need to compute the influence relationship for O_3 with existing facilities, as O_3 is influenced by c_3 .

Below, incorporating the ***influence relationship pruning rule*** and ***influence value pruning rule***, we present the influence pruning algorithm (IPA) to significantly improve the efficiency for solving the CLS-M problem.

4.2 Influence Pruning Algorithm

Algorithm 1 outlines the IPA algorithm. We use a max-heap HC and a min-heap HM to apply the pruning strategies. The entry of HC and HM is in the form of $\langle c.loc, inf(c) \rangle$ and $\langle c.loc, score(c) \rangle$, and ordered by $inf(c)$ and $score(c)$, respectively.

We pre-calculate and store the corresponding sets of objects in $Tr(C)$ which are influenced by candidates. Candidate locations in HC indicate the upper bound of influence score. We first calculate the score of the top- k candidates in HC and insert the pairs $\langle c.loc, score(c) \rangle$ into HM (lines 2–11). The minimum score is taken as the current threshold (line 12). For each remaining candidate in HC, if $inf(c)$ is less than the current threshold, which means the candidates in HM are results, the algorithm is finished based on the influence value pruning rule (lines 13–15). Otherwise, we further validate their scores (lines 17–22). If the score is greater than the current threshold, the corresponding candidate is inserted into HM. We update the threshold with the

minimum score in HM (lines 23–26). Once $score(c)$ is found to be less than the current threshold, c can be discarded. Finally, we return elements in HM as the top- k answers to the CLS-M problem. Notably, in the process of score calculation, it is unnecessary to perform the repeated traversals for facilities and objects. Key flags are used to judge whether an object O has computed with the facilities which influence O (lines 4–9, 17–22). When the flag equals to 0, we need to both compute and record the number of facilities that affect the object. If the flag is equal to 1, we only compute the score. Key flags avoid the repeated traversal of the influence calculation.

4.3 Theoretical Analysis

In this part, we provide a theoretical study on IPA. The worst case occurs when every candidate influences all the objects and candidates have the same score. In this case, the two pruning rules cannot be used and the complexity is $O(|\Omega| \cdot (|F| + |C|))$. The complexity of the influence relationship calculation between objects and existing facilities is $O(|\Omega| \cdot |F|)$. The complexity of evaluating the influence score of candidates is $O(|\Omega| \cdot |C|)$.

However, the positions of moving objects are with skewed distributions in common cases. Then, the influence relationship pruning rule can dramatically reduce the number of objects that are not influenced by candidates. Similarly, candidates are randomly located, then, the influence value pruning rule will further prune candidates with less scores, as well as the influence relationships of the corresponding objects. Hence, the number of candidates to be calculated is reduced to $|C'|$ and the moving objects are reduced to $|\Omega'|$ ($|C'| \ll |C|$ and $|\Omega'| \ll |\Omega|$). Hence, the average complexity of common cases is $O(|\Omega'| \cdot (|F| + |C'|))$. Experiments in Sect. 5 will validate the analysis. The best case occurs when the top- k candidates have no competitors. In the special case, the performance of IPA is equivalent to that of PINOCCHIO-VO.

Table 2 Description of real-world datasets

	Foursquare (F)	Gowalla (G)
Number of users (objects)	2321	10162
Number of check-ins (positions)	83615	381165
Avg check-ins	36	37
Min check-ins	2	2
Max check-ins	330	780

Algorithm 1: IPA algorithm

Input: $C, F, \Omega, k, Tr(C)$
Output: top- k optimal candidates

```

1 initialize HC, HM;
2 for  $i=1$  to  $k$  do
3   for each  $O$  in  $Tr(HC.top)$  do
4     if  $flag = 1$  then
5        $score := score + \frac{1}{|\sigma^{-1}(O)|+1}$ ;
6     else
7       compute facilities that influence the
        object  $\sigma^{-1}(o)$ ;
8        $flag := 1$ ;
9        $score := score + \frac{1}{|\sigma^{-1}(O)|+1}$ ;
10    insert  $\langle c, score(c) \rangle$  to HM;
11    Pop(HC);
12  $threshold := score(HM.top)$ ;
13 for each  $c$  in  $HC$  do
14   if  $HC.top.inf \leq threshold$  then
15     break;
16   else
17     for each  $O$  in  $Tr(C)$  do
18       if  $flag = 1$  then
19          $score := score + \frac{1}{|\sigma^{-1}(O)|+1}$ ;
20       else
21         compute facilities that influence the
          object  $\sigma^{-1}(o)$ ;
22          $score := score + \frac{1}{|\sigma^{-1}(O)|+1}$ ;
23     if  $score > threshold$  then
24        $threshold := score$ ;
25       Pop(HC);
26       Pop(HM);
27     else
28       Pop(HC);
29 return top- $k$   $c$  in HM;
```

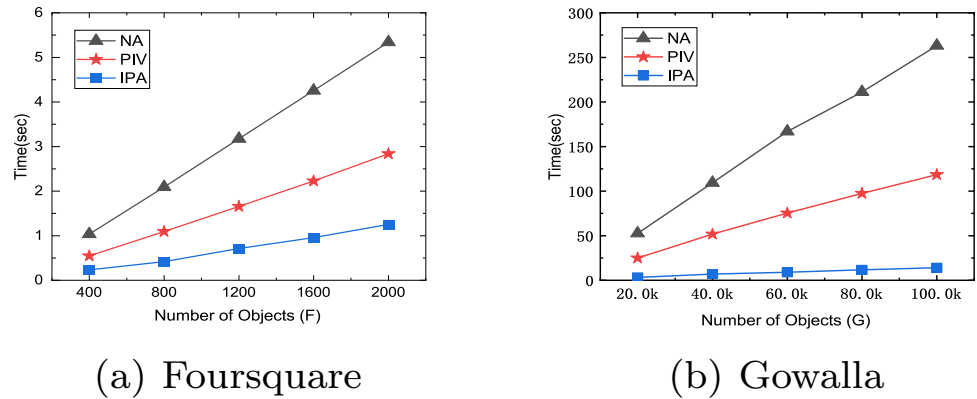
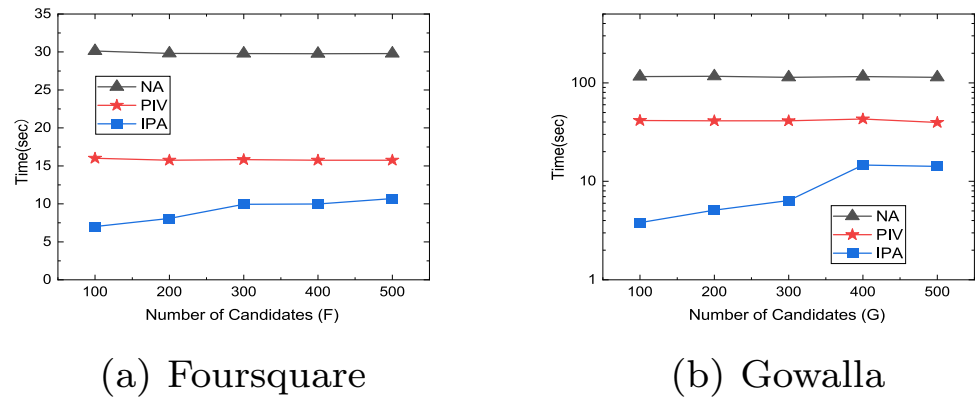
5 Experiment

In this section, we investigate the performance of our solution from a variety of aspects.

5.1 Experiment Setup

1. **Datasets** Table 2 describes two real-world datasets we use in the experiments.² The positions of check-ins in Foursquare are all located in Singapore, while those in Gowalla are mainly in California. The results in [11] show that using 24–48 positions can achieve a trade-off between accuracy and cost. We follow the setting and synthesize larger datasets (100k objects) using normal distribution based on users' positions in Gowalla. We choose positions from check-in

² Both datasets are available from the work [11].

Fig. 3 Effect of $|\Omega|$ **Fig. 4** Effect of $|C|$ 

coordinates as candidate locations by random uniform sampling. The existing facilities use real facility dataset.³

2. Algorithms

- NA: The straightforward method that exhaustively computes the cumulative influence probabilities for all the candidate-object and facility-object pairs.
- PIV: It refers to PINOCCHIO-VO in [11].
- IPA: The algorithm is described in Algorithm 1.

3. *Environment* All the algorithms are implemented in C++, running on a 3.3 GHz machine with 8 GB RAM under Windows 10 (64 bit).

The default values of constant k , probability threshold τ , the numbers of candidates and existing facilities are set to 10, 0.9, 100 and 200, respectively.⁴

5.2 Experiment Results

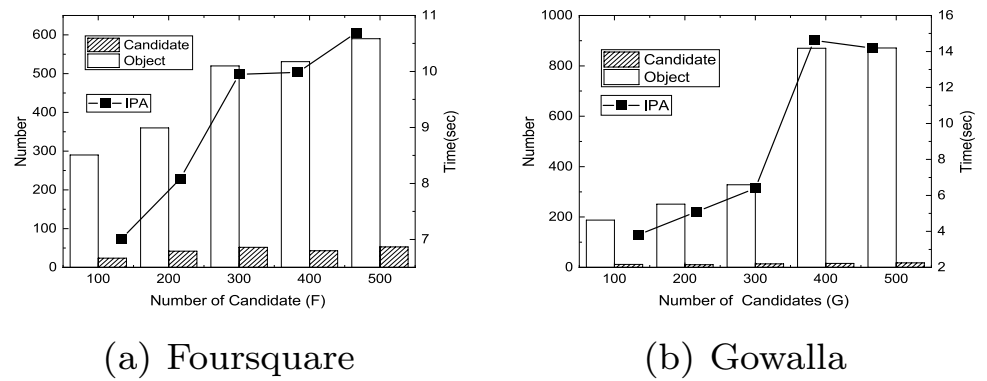
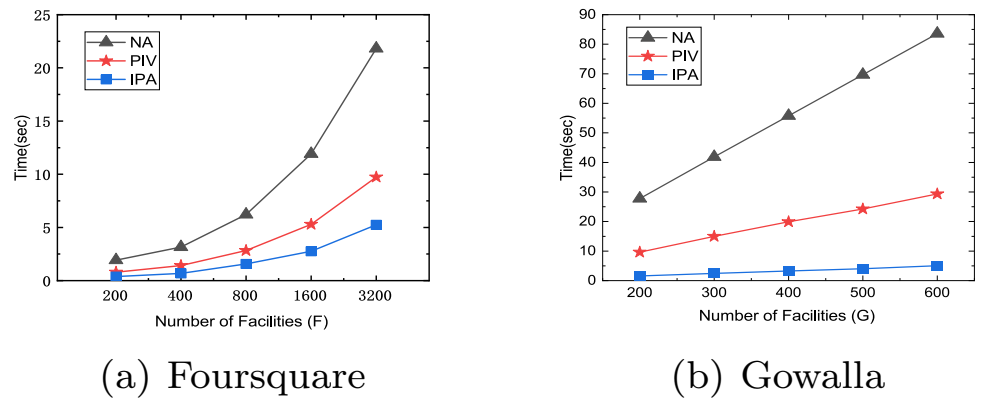
Effect of $|\Omega|$. We first study the performance by varying the number of objects. In order to highlight the scalability with respect to $|\Omega|$ on efficiency, we conduct experiments on Foursquare and Gowalla with real and synthetic user datasets, whose cardinalities are relatively small and large. Figure 3 shows the results. Compared with small $|\Omega|$ s, the pruning effect of IPA is more obvious for larger ones. As illustrated in Fig. 3b, the running time of IPA is remarkably stable with the increase of $|\Omega|$, and the other algorithms grow linearly, which means IPA is more scalable for massive objects.

Effect of $|C|$. We investigate the performance with respect to the number of candidates. The number of existing facilities is set to $1k$. As shown in Fig. 4, IPA exhibits the best performance, followed by PIV and NA. The running cost of IPA is at least one order of magnitude lower than NA. When $|C|$ grows, the costs of NA and PIV are stable. The reason is that, before the score computation, both NA and PIV have to perform a full traversal on the influence relationship

³ Facility dataset is available from the work [10].

⁴ Most of the default values are consistent with PINOCCHIO-VO in [11], which makes experiment results more comparable.

Fig. 5 Effect of pruning

Fig. 6 Effect of $|F|$ 

calculation between all facilities and objects, which means $|C|$ has no effect. For IPA, as the number of candidates to be accessed increases, the number of objects influenced by these candidates also raises. The main overhead of IPA is due to the number of facilities that influence objects.

Figure 5 shows that the running time and the number of objects to be calculated have a similar trend. As $|C|$ increases, the pruning effect of IPA drops. When the number of candidates is very large, IPA will degenerate to PIV. Because candidates will cover almost all the objects. In Figs. 4b and 5b, the processing time of IPA when $|C|$ is set to 500 is less than the case of 400. Affected by the data distribution, there are fewer competing existing facilities near the new added candidate locations, which leads to a decrease in the actual location data involved in the calculation.

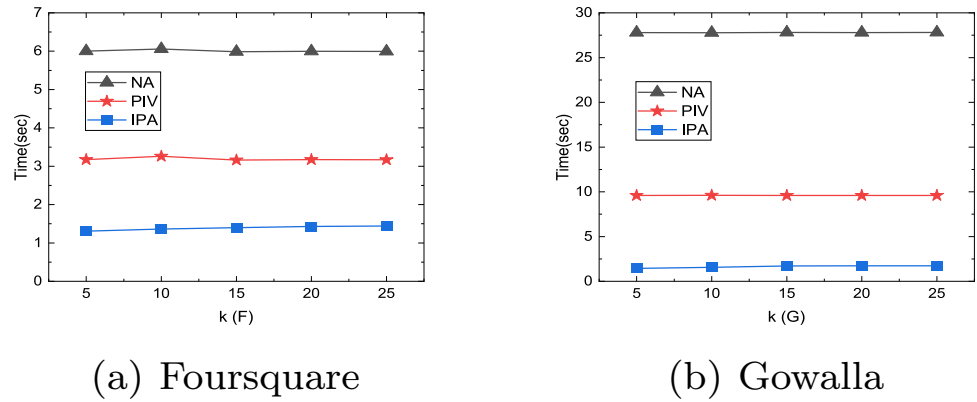
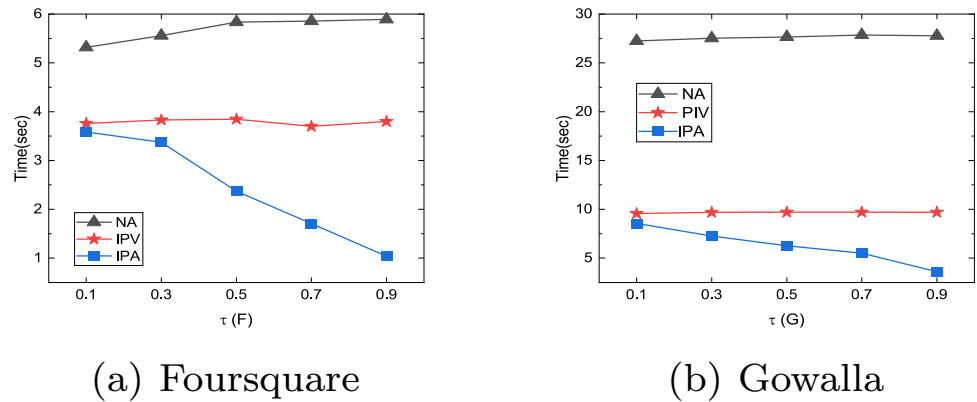
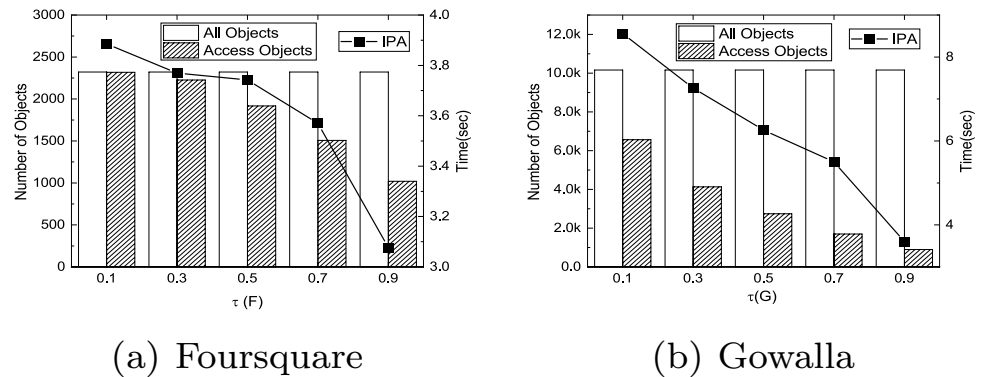
Effect of $|F|$. In this part, we vary the number of existing facilities both exponentially and linearly. Since the results on both datasets are qualitatively similar, due to space constraint, we report the results varying $|F|$ exponentially in Foursquare and linearly in Gowalla, respectively. As shown in Fig. 6, the computation costs of algorithms grow when $|F|$ increases exponentially (in F) and linearly (in G). The main reason is that the number of objects to be accessed is basically stable, and the time increases with $|F|$. The slowest growth rate of IPA shows its superiority.

This is because both pruning strategies work well. The pruning effect is more obvious when $|F|$ is larger.

Effect of k . As illustrated in Fig. 7, the computational time of IPA can be reduced by more than an order of magnitude compared to NA, and it is also significantly better than PIV. As k increases, efficiencies of the three algorithms are all stable. This is because, with the help of max-heap, the number of objects that are accessed does not increase noticeably with k .

Effect of τ . As shown in Fig. 8, the running costs of NA and PIV do not change significantly when varying the threshold, while the computation time of IPA drops when τ increases. This is because NA and PIV need to perform a full traversal calculation on the influence relationships between facilities and objects. When the threshold is set to 0.1, the performance of IPA degrades to PIV. In addition, the definition of influence relationship in [11] shows that τ represents the balance between distance and object quality. The larger threshold value is, the more attention is paid to the contribution of distance to the influence relationship.

Figure 9 reports the effect of τ on the pruning strategies. When τ is set very small, candidates which are accessed will affect almost all the objects. At this time, all algorithms need to traverse all the objects. Moreover, as τ increases, the number of objects to be accessed will

Fig. 7 Effect of k **Fig. 8** Effect of τ **Fig. 9** Effect of pruning

decrease. Hence, the larger τ is, the more effective pruning effect and better performance of IPA is.

6 Conclusions

In this article, we investigate a novel competitive LS problem called CLS-M, which takes into account competition against existing facilities in moving scenes. Specifically,

based on a novel competition-based influence score model, top- k optimal locations are selected. To solve the problem under the large amount of data, we develop an algorithm called IPA which leverages two pruning strategies. Experimental study over two real-world datasets demonstrates significant superiority of our algorithm in comparison with the baseline method and a state-of-the-art LS technique in terms of efficiency. In future work, we will also consider the influence of cooperation.

Funding This work was supported by the National Natural Science Foundation of China (Nos. 61976168, 61972309), Natural Science Basic Research Program of Shaanxi (No. 020JM-575) and the Ph.D. Fund of Xi'an Polytechnic University (No. BS201919), Key Research and Development Plan of Shaanxi Province (No. 2019ZDLGY13-09), Key Scientific Research Program of Shaanxi Provincial Department of Education (No. 20JY014), CCF-Huawei Database System Innovation Research Plan (No. CCF-HuaweiDBIR008B), CCF-Huawei Database System Innovation Research Plan (No. 2020010B).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Cheema MA, Lin X, Wang W, Zhang W, Pei J (2009) Probabilistic reverse nearest neighbor queries on uncertain data. *TKDE* 22(4):550–564
- Chen J, Huang J, Wen Z, He Z, Taylor K, Zhang R (2014) Analysis and evaluation of the top-k most influential location selection query. *Knowl Inf Syst* 43(1):181–217
- Huang J, Wen Z, Pathan M, Taylor K, Xue A, Zhang R (2011) Ranking locations for facility selection based on potential influences. pp 2411–2416
- Huang J, Wen Z, Qi J, Zhang R, Chen J, He Z (2011) Top-k most influential locations selection. *CIKM* 4:2377–2380
- Korn F, Muthukrishnan S (2000) Influence sets based on reverse nearest neighbor queries. In: *SIGMOD*, pp 201–212
- Kritter J, Bréviliers M, Lepagnot J, Idoumghar L (2019) On the optimal placement of cameras for surveillance and the underlying set cover problem. *Appl Soft Comput* 74:133–153
- Plastria F (2001) Static competitive facility location: an overview of optimisation approaches. *Eur J Oper Res* 129(3):461–470
- Revelle C (2006) The maximum capture or sphere of influence location problem: Hotelling revisited on a network. *J Reg Sci* 26(2):343–358
- Serra D, Revelle C (1994) Competitive location in discrete space. *Economics Working Papers*
- Wang M, Li H, Cui J, Bhowmick SS, Liu P (2020) Frost: movement history-conscious facility relocation. *ACM TIST* 11(1):1–26
- Wang M, Li H, Cui J, Deng K, Bhowmick SS, Dong Z (2016) PINOCCHIO: probabilistic influence-based location selection over moving objects. *TKDE* 28(11):3068–3082
- Wong RCW, Özsu MT, Yu PS, Fu AWC, Liu L (2009) Efficient method for maximizing bichromatic reverse nearest neighbor. *VLDB* 2(1):1126–1137
- Xia T, Zhang D, Kanoulas E, Du Y (2005) On computing top-t most influential spatial sites. In: *VLDB*, pp 946–957
- Yan D, Wong RCW, Ng W (2011) Efficient methods for finding influential locations with adaptive grids. In: *CIKM*, pp 1475–1484
- Yiu ML, Lu H, Mamoulis N, Vaitis M (2010) Ranking spatial data by quality preferences. *TKDE* 23(3):433–446
- Zhan L, Zhang Y, Zhang W, Lin X (2012) Finding top k most influential spatial facilities over uncertain objects. In: *CIKM*, pp 922–931
- Zhang P, Bao Z, Li Y, Li G, Zhang Y, Peng Z (2018) Trajectory-driven influential billboard placement. In: *SIGKDD*, pp 2748–2757
- Zhang X, Rey D, Waller ST (2018) Multitype recharge facility location for electric vehicles. *Comput Aided Civ Infrastruct Eng* 33(11):943–965
- Zheng K, Huang Z, Zhou A, Zhou X (2011) Discovering the most influential sites over uncertain data: a rank-based approach. *TKDE* 24(12):2156–2169