# Toward Collective Privacy Using Coordinative Path Planning

Haoran Xu and Yuqing Sun

School of Computer Science and Technology, Shandong University, Jinan, China
hr_xu1990@163.com, sun_yuqing@sdu.edu.cn

**Abstract.** With the increasing importance of location based services in people's daily lives, location related privacy becomes a critical issue. Most of the current solutions protect users' privacy by cloaking users' exact positions when they invoke requests on the location based service. In this paper, we tackle the location privacy problem in navigation applications in a different way. Based on the trusted third-party architecture and the $k-anonymity$ criterion, we propose a coordinative path planning algorithm for collective privacy. The novelty resides on two folds. One fold is from the predication perspective rather than the current solutions' focusing on on-site users. A user would be recommended a privacy-preserving path when he sends a navigation request. Another fold is by intentionally collective privacy rather than the traditionally independent calculation. The planned path for each user would be adjusted such that a set of users could be provided more privacy without degrading each user's privacy. To evaluate the proposed solution, we perform a set of experiments on both synthesis data and practical data. The experimental results show the efficiency and effectiveness of our method.

**Keywords:** location privacy, path predication, coordination.

## 1 Introduction

The development and integration of wireless communication and positioning technologies have promoted the generation and advances of location based services (LBS). By obtaining an individual's location information, location-based service providers can help him/her for navigation, friend-finder, point of interest (POI), or emergency rescue, etc. Although the wide variety of location-based services provide convenience to people's lives, the accessed location information may reveal personal habits, social customs, religious or other privacy information of individuals, which is a potentially serious threat to people's privacy.

The most representative technique for preserving privacy in LBS is to introduce a trusted third party anonymity server and the $k-anonymity$ criterion [1–6]. After getting a LBS request from a user, the anonymity server would send a cloaking region instead of the user's exact position to a LBS server [7] such that there are other $k-1$ users in the same region. Under this scheme, the uncertainty, which an adversary has in matching each exact user to a known location-identity

Q. Zu et al. (Eds.): ICPCA/SWS 2013, LNCS 8351, pp. 698–710, 2014.

association, depends on the number of actual users in the same cloaking region at the same time. However, in practice, there may not exist enough users in a considered region all the time, which makes the $k-anonymity$ criterion unsatisfied. Although some methods consider this case by enlarging the cloaking region or delaying the response [4], they degrade the quality-of-service (Qos) on spatial inaccuracy or response time.

Another representative scheme is based on peer-to-peer architecture, where the location privacy is protected by users' sharing LBS information with each other or by generating a cloak region among them. However, these solutions are based on the assumption that all users are benign and wiling to participate for collaboration. In case there is a malicious user in the group participating the collaboration, the cloaking result as well as the privacy would not be controlled[8, 9]. Shokri et al. propose an approach that a user can get the required LBS information from his neighbours, who send the same quires to a LBS server recently and have stored the results in their buffers. Otherwise, the user has to query the LBS server himself. The main drawback of this proposal is that each user has to store large-scale data and the provided information may not be applicable to the real-time context. Therefore, it is a challenging problem to protect users' location privacy without degrading the quality of LBS.

In this paper, we tackle the location privacy problem in the navigation applications in a different way. Based on the trusted third-party architecture and the $k-anonymity$ criterion, we propose a coordinative path planning algorithm for multiple users. The novelty resides on two folds. One fold is from the predication perspective rather than the current solutions' focusing on on-site users. Another fold is by intentionally collective privacy rather than the traditional independent calculation. When a user sends a navigation request, he is recommended a privacy-preserving path according to his preference, on which the k-anonymity is satisfied. The planned path may be adjusted when he moves ahead such that a set of users could be provided more privacy without degrading each user's privacy. The purpose of this mechanism is to benefit multiple individuals via their collective activities. This idea is inspired by the collective intelligence, where a consensus decision can be made by sharing or grouping intelligence of many individuals aiming to improve the group conditions. To evaluate the proposed solution, we perform a set of experiments on both synthesis data and practical data. The experimental results show the efficiency and effectiveness of our method.

The rest of this paper is organized as follows. Section 2 summarizes the related works. We present the details on the framework and the proposed algorithm in Section 3. In Section 4, we discuss the experimental evaluation of our proposed method. Finally, section 5 concludes the paper with a discussion on future work.

## 2 Related Work

**K-anonymity Based Privacy Protection.** Location k-anonymity is first proposed by Gruteser et al. in [3], which requires not less than a number $k$ of users

in a considered region covering the LBS query senders. They develop a quadtree-based cloaking algorithm to construct cloaking area. However, this method protects location privacy based on the assumption of enough users in a cloaking region, which can not be satisfied all the time in real life. Gedik et al. develop the method of enlarging the cloaking region to satisfy the $k - anonymity$ criteria [4]. But this approach would degrade the quality-of-service (QoS) of LBS applications at the same time.

Different from above solutions considering the snapshot LBS query, Chow et al. solve the privacy problem in continuous query scenario [10]. It requires that, for a user's sequence of queries, the cloaking region for each time must consist the same set of users in the previous cloaking region. Wang proposes an efficient solution to cloak not only a user's accurate location but also speed and direction [6], which is an improved version of cloaking box. These approaches also degrade the quality-of-service (QoS) of LBS applications. Ji et al. propose a privacy-preserving path predication algorithm so as to provide more guarantee on privacy as well as less detour [11]. As a continuous work, we adopt the similar idea in this work that predicates a secure path for the requester so as to avoid insecure cases in advance. Differently, we consider a more critical scenario with less users, where there may not exist a secure path, and promote cooperation among multiple users for collective privacy.

**Collaborative Location Privacy.** The collaborative location privacy mechanism in a peer-to-peer environment is also related with our work. To protect location privacy, a specific software should be installed on users' portable intelligent positioning devices. In Ferrer et al. proposed work [8], the software perturbs each user' location with Gaussian noise and broadcasts this fake position to others. When a user invokes a LBS query, it collects other $k-1$ nearest neighbours' perturbed positions and computes the centroid of these positions as the request location for LBS. Since the position of centroid can not be determined in advance and may be far from the requester, the quality of service can not be guaranteed. A little difference with the above method, Chow et al. propose a P2P spatial cloaking algorithm [9] that computes a cloaking region via peer-to-peer users' cooperation instead of sending the exact location to a LBS server. After forming a group with $k-1$ nearest peers, a user regards the region covering these peers as the cloaking area. In case there is a malicious user in the group participating the collaboration, the cloaking result would not be safe.

Shokri et al. propose a user-collaborative privacy preserving approach, MobiCrowd, to avoid disclosing user's location information to the LBS server [12]. A user can get the required LBS information from his neighbours, who send the same quires to a LBS server in recent time and have stored the results in their buffers. Otherwise, the user has to query the LBS server himself. The main drawback of this proposal is that each user has to store large-scale data and the provided information may not be applicable to the real-time context. Our approach could overcome these drawbacks above and provide a higher guarantee on location privacy.

**Location Based Recommendation.** Our work is also related to the location based recommendation, which focuses on calculating the most popular route. For example, Wei et al. propose a route inference framework to mine the most popular route by use of a large number of coarse GPS trajectories [13]. By modelling each individual's personal preferences and referencing opinions from local experts, the location-based and preference-aware recommender system is proposed to offer top-k ranked of routes for a particular user [14]. However, they do not consider the real-time context and the privacy issue.

Taking into account the real-time taxi trajectories, Yuan et al. propose a cloud-based driving direction system to provide a user with a fastest route to a destination [15]. Given a user-location matrix, the inference model based on HITS predicts the significance ranking of a physical location [16] by considering the correlation between locations. This work does not consider the privacy issue either.

## 3 The Coordinative Path Planning Algorithm

### 3.1 The Privacy Model

In this work, we adopt the widely used trusted third-party architecture, shown as Figure 1, and the $k-anonymity$ criteria for preserving users' location privacy. There are three parties in this architecture: LBS users, anonymity server, and LBS server.

*Mobile Users (Users)*: Users are associated with mobile positioning devices such as mobile phones, PDA, or laptops etc. They can send LBS requests to the anonymity server with their current physical locations. According to users' privacy preferences, users can be classified into two categories: privacy-sensitive users and ordinary users. We would only consider privacy-sensitive users in the following discussion and adopt *users* for short. Each user is allowed to move in a different speed.

*Anonymity Server (A-Server)*: is a fully trusted third-party who is responsible for protecting LBS users' privacy while maintaining a certain extent of quality-of-services (QoS). The channel connecting to users is assumed secure. When users ask for LBS, the anonymity server receives their exact positions and anonymizes them to get a cloaking region including at least $k$ users before submitting to the

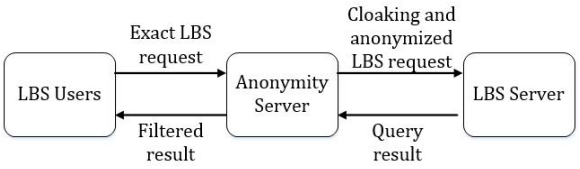



**Fig. 1.** The Trusted Third-party System Architecture

LBS server. Even though an adversary gets these anonymized information, he could not identify a specific user. For example, under $k-anonymity$ criteria, the adversary's confidence of linking a real user to a requester is less than $1/k$.

*Location Based Service Provider (LBS server)*: A LBS server provides location based services to its subscribed mobile users. When it receives a LBS request, it processes the query according to the associated geographic position, either exact point or cloak region. Then it returns the response to the requested user. LBS server is not responsible for preserving privacy of the mobile users.

In this work, based on the above architecture, we propose the predication based collective privacy solution. This solution focuses on the navigation applications, which are very popular in practice, and can be adaptive to other related LBS with the characteristics of continuous LBS queries while users moving ahead. Under this scenario, a user may invoke a LBS request when going along the path from his start to the destination. For a set of navigation requests, the purpose of our approach is to provide more privacy guarantee without degradation of QoS by holistically considering these requests. Firstly, the anonymity server invokes the path planning algorithm to find a privacy-preserving path for each request according to its preferences on privacy and distance, such as an integer $k$ as the $k-anonymity$ criterion. In case no privacy-preserving path can be found or when a user's next step becomes insecure, A-Server would coordinate multiple users to improve their collective privacy. This idea is inspired by collective intelligence, where multiple individuals would all benefit more from their collective activities than any one individual undertakes and solves it alone.

### 3.2 Problem Definition

In this subsection, we would first present the basic concepts and notions that would be used in the following sections and give a formal definition of the problem studied in this paper.

*Map and Map Situation* : The considered geographical region is called a map in this paper, denoted as $Map$, generally referring to a city or a borough. For a given map $Map$, map situation reflects the distribution of the dummy users [17] and active users, who all participate for collective privacy.

*User Request* : Given a map $Map$, a Privacy-Preserving Navigation request (PPN request for short) is in the form of 4-tuple $< u_{id}, Start, Destination, k >$, where $u_{id}$ is the identifier of the requester, $Start \in Map$, and $Destination \in Map$ respectively represent the user's start and destination for navigation, $k$ is an integer denoting the user's privacy preference for $k-anonymity$ criteria.

*Problem Definition* : For a set of PPN requests, donated as $R=\{r_1, r_2, \cdots, r_n\}$, where $r_i$ represents a PPN request, our purpose is to find a $k-anonymity$ secure path for each request according to his privacy preference, donated by $kA-Path=\{p_1, p_2, \cdots, p_l\}$, where $p_i \in Map$, such that no matter a LBS request

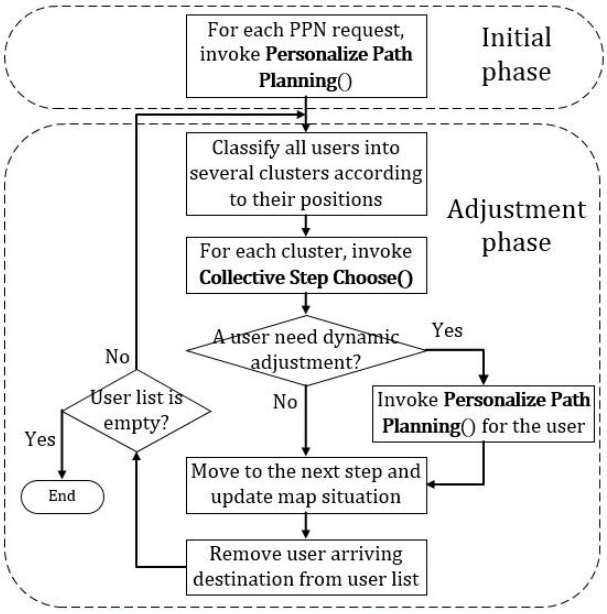



**Fig. 2.** The path planning process

occurs at any point of the path, the cloaking region is $k - anonymity$ secure. Each edge$< p_i, p_{i+1} >$ connecting $p_i$ and $p_{i+1}$ maps to a street between two neighboring blocks.

### 3.3 The Coordinative Path Planning Algorithm

In this subsection, we propose the coordinative path planning algorithm among multiple users so that they can benefit from each other. The main process of our solution is shown in Figure 2. It includes two phases: the initial phase and the adjustment phase.

In the initial-planning phase, for each PPN request, an anonymity server firstly invokes the **Personalized Path Planning** method to find a privacy preserving path according to the requester's preferences on privacy, namely $k$ for $k - anonymity$ criterion. We adopt the algorithm proposed in [11] to perform the **Personalized Path Planning**, which is based on the D* path planning algorithm for solving the robot movement problem. It integrates location privacy with distance as the criteria for choosing a path. In our problem, for a PPN request $< u_{id}, Start, Destination, k >$, this method will find a privacy preserving path from *Start* to *Destination*.

If there is not a secure path or an insecure cloaking region on next step during user movement, the second phase of dynamic adjustment is invoked. The anonymity server then coordinates all LBS users' movement for higher collective

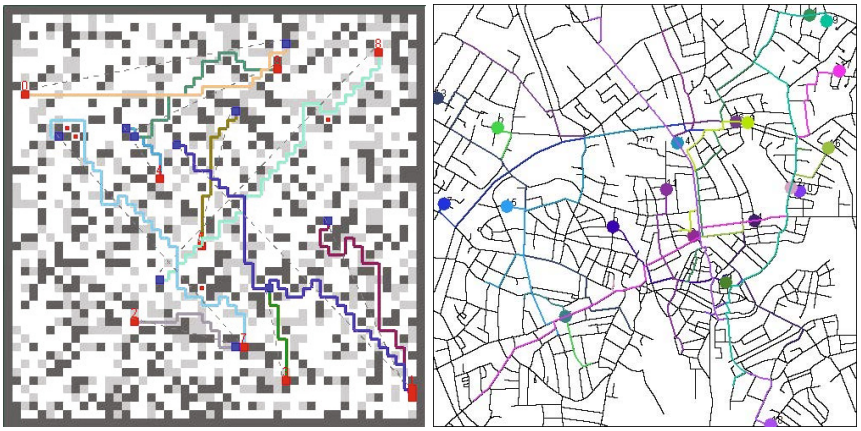
privacy by finding an optimal next step for each user according to the current map situation. We consider collective privacy from several aspects. The most important aspect is to check whether a user is k-A secure. A user is said to be secure if he is in a $k-anonymity$ secure area satisfying his preference. Considering a set of requests, the more k-A secure users, the more collective privacy. When the number of secure users is the same in two different choices, we consider a higher entropy aspect. This is because user privacy is protected by confusing a set of users in a cloaking area, the more users in a region, the more entropy. Another aspect is distance since it is a very important issue in navigation applications. When the privacy under two choices is the same, the shorter one is better. Besides, other user concerned aspects can be taken into account, such as distance according to practical requirements.

The key of this process is the **Collective Step Choose** algorithm, shown as Algorithm 1, it recursively searches the overall optimal choices. The recursive part discusses the general situation that more than one users need to be taken into account for path planning. It is recursively solved by the reduction to its sub-problem on the number of users from $n$ to $n-1$. To integrated above considerations on collective privacy, we introduce the **Dominate()** function to make a comparison between two choices and adopt $BestOption$ to store the optimal options for all users. $OPT$=$\{opt_1, opt_2, \cdots, opt_n\}$ is used to store the candidate choices for all users during each recursion, where $opt_i$ is represented as current choice of user $u_i$. During each recursion, we will replace the $BestOption$ with an dominant $OPT$.

**Algorithm 1.** The Collective Step Choose

```
function The Collective Step Choose(cluster, n)
  if (n = 1) then
    for user u_1's each step option opt_1 do
      collective privacy calculation
      if Dominate(BestOption,OPT) then
        update BestOption with OPT
      end if
    end for
  else
    for user u_n's each step option opt_n do
      The Collective Step Choose(cluster, n − 1)
    end for
  end if
end function
```

To reduce the practical complexity of this algorithm, we will classify all LBS users on the user list into several clusters according to their positions. This is based on the fact that only the geographically close users have opportunities to be coordinated for collective privacy. The **Collective Step Choose** algorithm would be invoked in each cluster and the map situation is updated periodically.



(a)on simulated map (b) on real map

**Fig. 3.** The Coordinative path planning for multiple users

There are many efficient cluster algorithms can be used, such as DBSCAN [18], which is omitted here.

## 4 Experimental Study

We prototype a system and perform a series of experiments to verify the proposed solution on both simulated data and real data. The purpose of these experiments includes two folds. One is to better understand the effectiveness on location privacy protection after applying our method. Another is to study how well the efficiency scales with different parameters.

### 4.1 Experiment Setting

The experiments were conducted on a desktop with Intel 3.10GHz CPU, 3.16G memory and 500GB disk space. Its operating system is Windows 7.All experimental results are the average values of more than ten times of program running. LBS users are randomly distributed on the map, which is represented as a mesh or grid and each cell of the mesh is regarded as a cloak region when a user requests a location-based service.

To simulate practical instances in real life, we select the mesh size ranging from 40*40 to 500*500. A set of *dummy* users are generated and distributed in the mesh. Let $N_{dummy\ user}^{each\ cell}$ denotes the number of dummy LBS users in each cell. In Figure 3 (a), for illustrative purpose, we denote the cells with few dummy users by the deeper colour. User requests are randomly distributed over the mesh. The number of total PPN requests is denoted as $|R|$. The abbreviations of these parameters would be used in the rest of this section.

The practical map is selected from the OldenBurg County generated by the well-known Thomas Brinkhoff Network-based Generator, which is widely adopted in mobile application based research works. We clip one part from the map with the size of 500x500 cells, approximately representing a geographical square of $4000m * 4000m$. The LBS users are randomly distributed in the this area. Each user had its own start and destination in this region with a random speed. Different with the color setting in the simulated map, the while color denotes the obstacles that can not be go through and the black lines denote roads or streets in the county.

### 4.2 Evaluation Metric

In this subsection, we introduce some metrics to evaluate a path so as to overall consider what a requester desires. In the experiment where these metrics are used, the map is a mesh $Mesh$ and a path is regarded as a sequence of cells from a start position to a destination. Based on these metrics, we would evaluate the proposed location privacy protection solution.

*Success Rate:* Success rate describes how well the $k-anonymity\ criteria$ is satisfied on this path. Given a map $Mesh$, an integer $k$ for the $k-anonymity$ criteria and a path *path* on $Mesh$, the success rate is defined as the k-anonymity private proportion of the whole path length.

*$K-$anonymity path privacy:* $K-anonymity$ Path Privacy is a more fine-grained evaluation of path privacy. Especially when there does not exist a $k-anonymity$ secure path, a user may choose a path with higher privacy. We define the privacy for a given cloaking region $c_i$ as: $\mathcal{H}(c_i) = -\sum_{i=1}^{\eta_i}(1/\eta_i * log_2(1/\eta_i)) = log_2(\eta_i)$, where $\eta_i$ is the number of users in cloaking region $c_i$(namely the size of the anonymous request set). The $K-anonymity$ Path Privacy ($kA-Privacy$) of the path, donated as $\mathcal{H}^k(path)$), is the average of all cell k-anonymity privacy $\mathcal{H}(c_i)$ on the path.

*Detour rate:* Detour rate expresses the tolerance a user would like to accept on path detour. Given a mesh $Mesh$, and a path *path*, the detour rate is the ratio of the number of additive cells that secure path is more than shortest path to the shortest path length.

### 4.3 Experiments and Analysis

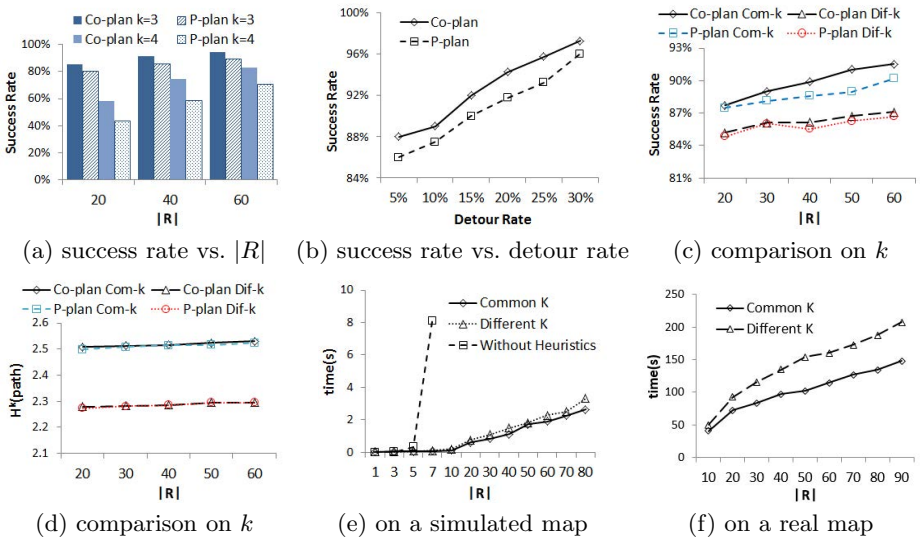
Since the primary purpose of our solution is to solve the collaborative privacy-preserving problem in critical situations, we pay attention on the overall privacy of all users. Figure 3 shows the path planning results of multiple users and each dog-leg line denotes a recommended path for a user, where (a) is evaluated on a simulated map and (b) is on a real map. In a color display, we can see that each color line denotes a planned path for an individual.

**Table 1.** The experimental settings of Figure 4

| figure | mesh size | $\|R\|$ | $N^{each\ cell}_{dummy\ users}$ | k |
|---|---|---|---|---|
| (a) | 400 | [20..60] | [0..2] | [3..4] |
| (b) | 400 | 10 | [0..9] | 4 |
| (c) | 1600 | [20..60] | [0..9] | [3..9] |
| (d) | 1600 | [20..60] | [0..9] | [3..9] |
| (e) | 3600 | [1..80] | [0..9] | [2..6] |
| (f) | 250000 | [10..80] | [0..9] | [2..6] |

Then we perform a series of experiments to quantitatively assess the effectiveness of our method. The experiments settings are listed in table 1 and results are shown in Figure 4. Since the semantics of success rate and path privacy are different under various $k$ values, we first evaluate the effectiveness on privacy protection by setting a common $k$ for collaborative users. Figure 4 (a) exams the relationship between success rate on path privacy and the number of collaborative users. The selected number of dummy users in each cell ranges with [0,2] so as to reflect critical situations considered in the coordinative path planning algorithm, namely a user's $k$ criteria is difficult to satisfy without collaboration. The results show that the coordinative path planning algorithm ($co-plan$ for short in the Figure) generates better privacy preserving paths for users than the personalized path planning algorithm ($p-plan$ for short), especially with the increase of $k$. Also we can see that the more users attend collaboration, the higher the success rates it gets. This implies that in a critical privacy environment, users could together reach a high guarantee on privacy via collaborating with each other on path planning. In Figure 4 (b), we exam the relationship between the average success rate and user detour rate. The results show that if users would like to detour more, they could be provided more guarantee on privacy by both personalized path planning and coordinative path planning. But the later has a higher success rate.

Considering in practice users may have different $k$ as their preferences, we then make comparison on common $k$ cases and different $k$ cases when applying the coordinative path planning algorithm, shown in Figure 4 (c) and (d). The $k$ values for users are randomly distributed from 3 to 9 and the common $k$ is set their average value $k = 6$. The results in (c) show that the success rate in common $k$ cases is better than in the different $k$ cases in coordinative path planning. It is easy to understand that some large $k$ is difficult to satisfy in most cases. And they are all better than non-collaboration. The results in (d) show that there is less difference between the coordinative planning and the personalized planning. This is because some high $k$ value may choose high privacy cell by path planning and in a well situation a privacy preserving path can be found via personalized planning. This again illustrates that the coordinative planning algorithm targets the critical situations. Overall, the common $k$ cases are better than different $k$ cases such that users can acquire more privacy via collaboration. In next section, we would further discuss what a common $k$ value is appropriate under a given mesh situation for a high guarantee on users' collective privacy.




(a) success rate vs. $|R|$ (b) success rate vs. detour rate (c) comparison on $k$

(d) comparison on $k$ (e) on a simulated map (f) on a real map

**Fig. 4.** The coordinative path planning algorithm

Finally, we exam the efficiency of our algorithm. Figure 4 (e) shows the relationship between the execution time and the size of the PPN request set $|R|$ on a simulated map. Since the computational complexity of the algorithm without any heuristic is exponential with the size of the PPN request set, the running time increases fast with $|R|$. However, the running time of the optimized algorithm with clustering heuristics is highly reduced on both common $k$ cases and different $k$ cases. We also exam the efficiency on a practical map and present the results in Figure 4(f). The results show that the running time is acceptable for about 100 PPN requests. Actually, the selected resolution is very fine, in which the size of a cell is less than 100 square meter. A practical application can accept a much coarser resolution. The reason of selecting $500 * 500$ is to test the computing pressure under such resolution since it can be applied to a whole city.

## 5 Conclusion

In this paper, we tackle the problem of collective privacy preserving in the navigation applications. Based on the trusted third-party architecture and the $k-anonymity$ criterion, we propose a coordinative path planning algorithm for multiple users' collective privacy, which is suitable for the less secure situation. For a set of user navigation requests, each user is recommended a privacy-preserving path according to his preference, on which the k-anonymity is satisfied. The planned path may be adjusted when a user moves ahead such that the total privacy of users is improved without degrading his privacy. This mechanism

can benefit multiple individuals via their collective activities. A set of experiments are conducted to evaluate the effectiveness and efficiency of our proposed method on both simulated data and real data. Experimental results show that our method provides higher privacy than users' random movement without degrading the Quality of Service. In the future, we would consider to optimize the collective path planning algorithm in directed graphs as well as to explore more efficiency algorithms combined with classical algorithm in graph theory .

**Acknowledgements.** This work is supported by the National Natural Science Foundation of China (61173140) and the National Science & Technology Pillar Program (2012BAF10B03-3).

## References

1. Merrill, S., Basalp, N., Biskup, J., Buchmann, E., Clifton, C., Kuijpers, B., Othman, W., Savas, E.: Privacy through uncertainty in location-based services. In: MDM 2013, pp. 67–72 (2013)
2. Yang, K.-T., Chiu, G.-M., Lyu, H.-J., Huang, D.-J., Teng, W.-C.: Path privacy protection in continuous location-based services over road networks. In: WiMob 2012, pp. 435–442 (2012)
3. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: MobiSys 2003, pp. 31–42 (2003)
4. Gedik, B., Liu, L.: Protecting location privacy with personalized k-anonymity: Architecture and algorithms. In: TMC, pp. 1–18 (2008)
5. Xu, T., Cai, Y.: Exploring historical location data for anonymity preservation in location-based services. In: INFOCOM 2008, pp. 547–555 (2008)
6. Wang, Y., Wang, L., Fung, B.C.M.: Preserving privacy for location-based services with continuous queries. In: ICC 2009, pp. 1–5 (2009)
7. Chow, C.-Y., Mokbel, M.F.: Trajectory privacy in location-based services and data publication. ACM SIGKDD 13(1) (2011)
8. Domingo-Ferrer, J.: Microaggregation for database and location privacy. In: Etzion, O., Kuflik, T., Motro, A. (eds.) NGITS 2006. LNCS, vol. 4032, pp. 106–116. Springer, Heidelberg (2006)
9. Chow, C.Y., Mokbel, M.F., Liu, X.: A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In: GIS 2006, pp. 171–178 (2006)
10. Chow, C.-Y., Mokbel, M.F.: Enabling private continuous queries for revealed user locations. In: Papadias, D., Zhang, D., Kollios, G. (eds.) SSTD 2007. LNCS, vol. 4605, pp. 258–275. Springer, Heidelberg (2007)
11. Ji, G., Sun, Y., Ma, X.: Path planning for privacy preserving in location based service. In: CSCWD 2011, pp. 162–167 (2011)
12. Shokri, R., Papadimitratos, P., Theodorakopoulos, G., Hubaux, J.-P.: Collaborative location privacy. In: MASS 2011, pp. 500–509 (2011)
13. Wei, L.-Y., Zheng, Y., Peng, W.-C.: Constructing popular routes from uncertain trajectories. In: KDD 2012, pp. 195–203 (2012)
14. Bao, J., Zheng, Y., Mokbel, M.F.: Location-based and preference-aware recommendation using sparse geo-social networking data. In: SIGSPATIAL 2012, pp. 199–208 (2012)

15. Yuan, J., Zheng, Y., Xie, X., Sun, G.: T-drive: Enhancing driving directions with taxi drivers' intelligence. IEEE Transactions on Knowledge and Data Engineering 25(1), 220–232 (2013)
16. Cao, X., Cong, G., Jensen, C.S.: Mining significant semantic locations from gps data. Proc. VLDB Endow. 3(1-2), 1009–1020 (2010)
17. Kato, R., Iwata, M., Hara, T., Suzuki, A., Xie, X., Arase, Y., Nishio, S.: A dummy-based anonymization method based on user trajectory with pauses. In: SIGSPATIAL 2012, pp. 249–258 (2012)
18. Ester, M., Kriegel, H.P., Jörg, S., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD (1996)