IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 30, 2018, accepted November 11, 2018, date of publication November 14, 2018,
date of current version December 18, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2881419

Dynamic Taxi Service Planning by Minimizing Cruising Distance Without Passengers

ZHONGWEN LUO[1], HUIMIN LV[1], FANG FANG[1], (Member, IEEE), YISHI ZHAO[1], YUANYUAN LIU[1], XIUQIAO XIANG[1], AND XIAOHUI YUAN[2], (Senior Member, IEEE)
[1]Faculty of Information Engineering, China University of Geosciences, Wuhan 430074, China
[2]Department of Computer Science and Engineering, University of North Texas, Denton, TX 76210, USA

Corresponding author: Xiaohui Yuan (xiaohui.yuan@unt.edu)

This work was supported in part by the Key Laboratory of Agricultural Electronic Commerce, in part by the Ministry of Agriculture of China under Grant AEC2018003 and Grant AEC2018006, and in part by the National Natural Science Foundation of China under Grant 61602429.

ABSTRACT It is a long-standing challenge to recommend routes to a group of taxis at different locations to minimize the overall mileage spent without customs. This paper studies route recommendation to a group of taxis by minimizing the overall mileage spent without customers. A new recommendation algorithm and evaluation model are proposed. The algorithm recommends the current optimal route and updates both the capacity and the probability along the route after getting a passenger. The new evaluation model is adapted to estimate the performance of each candidate route. A simulator that imitates the recommendation process on the real-world datasets and virtual taxis based on our recommendation system is developed. The experimental results demonstrate the effectiveness of the proposed evaluation model. The proposed method using the potential-cruising-distance model effectively reduces the global cruising distance of multiple taxis, especially in the applications with a large number taxis.

INDEX TERMS Route planning, urban traffic, GPS data analysis, location-based services.

I. INTRODUCTION

Taxi service is important in modern cities. However, it is common that taxicabs cruise around without passengers. A taxi recommendation system offers profitable information for taxi drivers and helps the management of public transportation by mining the historical taxi trajectories [1]–[5].

A practical recommendation system should take into account the change of the capacity and the pickup probability in a pickup location after getting a passenger. There are methods that recommend top locations for taxi drivers to enable drivers to pick up more passengers [6]–[8]. Others arrange routes in a circular list and adopt the Round-Robin method to generate a recommended route for each taxi [9], [10]. However, these studies fail to take into account the changes induced to the other taxicabs after a recommendation is made. The pickup rate varies with the number of passengers. The existing recommendation algorithms for multiple taxicabs need to accommodate the dynamics to achieve global optimization. In addition, most of the existing taxi recommendation systems employ probability distance to calculate the preference of a route [8]–[12]. However, the performance evaluation methods only consider the potential driving distance and ignore the dynamics of passengers taking a ride as well as the number of passengers along the route. Hence, current methods may send a taxi to a route which may have a long distance.

To overcome these limitations, we propose a recommendation algorithm considering the dynamic change of the probability after a taxi responds to a request. The algorithm aims to minimize the overall mileage spent without a fare. A performance evaluation model Potential-Cruising-Distance (PCD) [13] is used for measuring each candidate route. The recommendation algorithm considers the change of the capacity and the probability after getting a passenger. It also integrates the PCD model to measure a candidate route for getting the optimal route.

The main contributions are as follows:

1) A novel recommendation algorithm for multiple taxis at different positions is proposed. Based on the PCD model, the algorithm recommends each taxi to the current optimal path and updates the probability at each node. This means the new algorithm take account of the dynamic change of the pick-up probability of each pick-up point which will influence the next recommendation.
2) To find the optimal route for a target taxi driver, we design a data structure Path-Search-Tree (PS-Tree). Based on the monotonicity of the PCD model,

a pruning algorithm to speed up the searching process is proposed.

3) A proof of the monotonicity of the PCD evaluation model [13]. The PCD model takes account of the probability of taking a passenger along the route and can be used in a more widely situation.

The remainder of this paper is organized as follows. Section II reviews the related works on route recommendation. Section III presents the problem formulation and the system framework. Section IV presents the proposed method for taxis recommendation based on the PCD model. This section discusses the PCD evaluation model and the route recommendation algorithm. Section V discusses our experimental results using real-world taxi tracer data. Section VI concludes the paper with a summary and future work.

II. RELATED WORKS

In recent years, there are many research on taxi route recommendation [11], [12], [14]–[17]. Ge *et al.* [10] propose a taxi recommendation system to find shortest potential travel distance (PTD) for taxi driver to pick up a passenger. The taxi route recommendation system tries to provide an efficient route for taxi drivers using massive trajectory data. The massive taxi trajectory data also used in other area such as predict human mobility style [18] and find fastest route [19] [20].

Substantial attention has also been given to the algorithm of taxi dispatching to improve the performance of transportation systems. Miao *et al.* [21] design a data-driven distributional robust vehicle balancing method to improve the average system performance. Seow *et al.* [22] focus on more globally increasing customer satisfaction by concurrently dispatching multiple taxis to the same number of customers in the same geographical region and allowing taxis to exchange their booking assignments. Miao *et al.* [23] propose a receding horizon control framework for large-scale taxi dispatching considering both current and future demand. Zhang and Pavone [24] develop a closed-loop, real-time rebalancing policy for autonomous vehicles to ensure acceptable quality of service.

Ma *et al.* [25], [26] concern the real-time taxi share system, which tries to save energy by share a taxi among different people who have overlap trajectory. The main problem they try to solve is how to balance the need for taxi drivers and passengers. It is also important to make a real-time decision in the dynamic environment.

The recommendation problem complexity and improve the performance is also important. Statistics show that most recommendation problems are exponentially [27]. So some research [8], [28] try to use a data structure such as KD-tree and variant to minimize search. In this paper, we propose an algorithm for a group of competing taxis at a different position to minimize the overall potential cruising distance. Our previous conference paper [13] propose a PCD evaluate model. The model is correct in a more wide condition than previous widely used PTD model. In this paper, we provide a proof of monotonically of PCD function and based on which we propose a recommendation method which considers the impact of the current recommendation on the next.

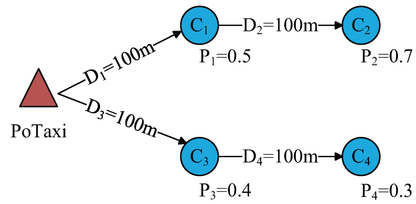


FIGURE 1. Two candidate path at time tp. PoTaxi is the current position of a taxi which require recommendation. node C_i is the recommended pick-up point, P_i pick-up rate at the point, D_i is the distance from node C_{i-1} to node C_i.

III. OVERVIEW

A. PROBLEM FORMATION

The route recommendation problem is to find an optimal route. Here we define the optimal route to be which have a minimal PCD to pick up a customer. And the PCD is defined as a function of a taxi's current position *PoTaxi*, current time *tp*, the route *R* which the taxi will follow. So the PCD function is $PCD(PoTaxi, tp, R)$ where the route R is a serial of pick-up points, i.e., $R = \{C_1 \to C_2 \to \ldots \to C_K\}$. The length of route R denotes as $|\overrightarrow{R}| = K$, the number of pick-up points. In some occation, we also denote the current position of a taxi ($PoTaxi$) as C_0.

Currently, most researchers agree that the conditional probability for getting a passenger at a pickup point and the cruising distance to that point is a good representation of the route property. So, the PCD is $PCD(PoTaxi, tp, P_R, D_R)$ where $P_R = \{P_1, P_2, \ldots, P_K\}$ denotes the picking-up rate at the pick-up points along route R; $D_R = \{D_1, D_2, \ldots, D_K\}$ represents the distance from previous pickup point to the current pickup point along route R.

So, the optimal path recommendation problem is to minimize the route to pick-up points: $\min_{R \in R_{set}} PCD(PoTaxi, tp, P_R, D_R)$ where R_{set} is the set of route R, which is generated from a potential pick-up point set C and which are generated from tracers of experience taxi driver.

The taxis route recommendation problem is to recommend optimal route for each taxi driver in different position to minimize the overall potential cruising distance. Given a set of the position (PoT_{set}), a set of taxis (M_{set}), current time (tp), the taxis route recommendation problem can be formulated as: $\arg\min \sum_{p \in PoT_{set}} \sum_{m \in M_{set}(p)} PCD(p, tp, P_R, D_R)$ Fig. 1 shows an example of two possible paths at a time period *tp*.

The widely used evaluation model PTD considering pickup cruising distance described as: $PTD = \sum_{i=1}^{n} P(C_i|R, tp)D(C_{i-1}, C_i)$ Where $P(C_i|R, tp)$ is the conditional probability of picking up passengers along route R at point C_i. This model is correct only if two routes have the same probability; otherwise, this is not applicable. As an example in Fig. 1, according to PTD definitioan, the PTD of route R_1 would be $P_1D_1 + (1 - P_1)P_2(D_1 + D_2) = 120$

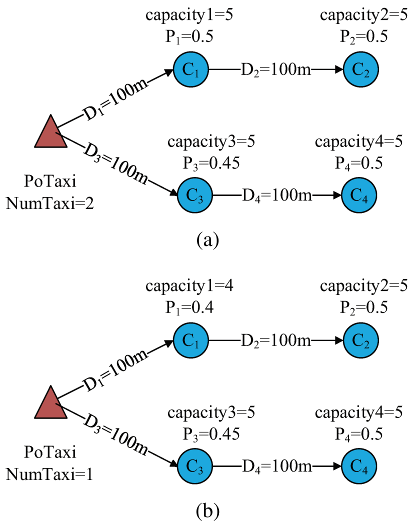


FIGURE 2. Recommendation for two taxis.

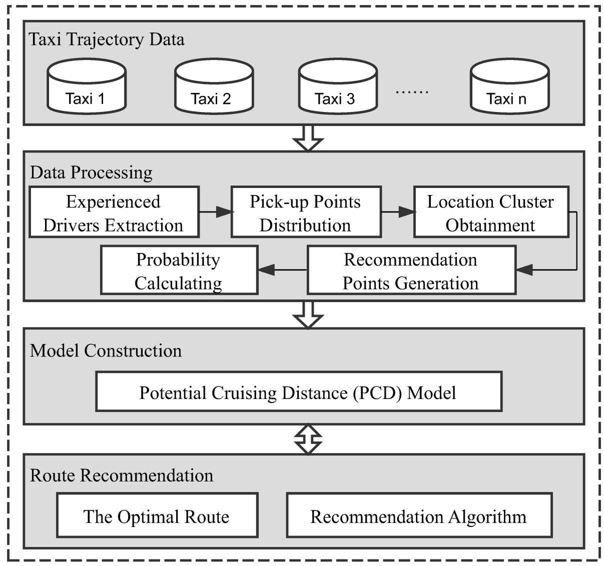


FIGURE 3. Overview of taxi recommendation.

the PTD of route R_2 would be $P_3D_3+(1-P_3)P_4(D_3+D_4) = 76$ The PTD criteria would suggest R_2 as a better choice. However, it is clear that route R_1 is a better choice because these two route has same distance vector, but route R_1 has a higher pickup rate. We have proposed PCD [13] which have solved this problem

Furthermore, the current method of the recommendation for a set of taxis does not take the change of the capacity and the probability of getting a passenger into account.

For example, as shown in Fig. 2(a) there are two empty cabs to be recommended. And there are two candidate routes $R_1 = \{PoTaxi, C_1, C_2\}$ and $R_2 = \{PoTaxi, C_3, C_4\}$. Obviously, the current optimal route is $R_1 = \{PoTaxi, C_1, C_2\}$. We first assign the first cab to route R_1, if that cab catches up a passenger in one of the pickup points, then the capacity at that pickup point will be changed. The picking-up rate will also vary according to the change of the capacity. Therefore, the route R_1 may not better than R_2. We should measure which route is the optimal one again and then recommend it to the second empty cab. So, in this paper, a recommendation algorithm considering these changes is proposed and which will minimize the overall potential cruising distance.

B. FRAMEWORK

The framework of our system is illustrated in Fig. 3. For the acquired taxi trajectory data, we first extract the experienced drivers by analyzing their driving model, usually, the experienced drivers will have a lower vacant ratio and enough driving time. And we can get the pick-up points of experience driver at a given time period by using a statistical analysis method. Then, the pick-up points, which are near in space, are clustered. The different period may have different pick-up points. So the pick-up points have spatial and time stamps. Subsequently, we calculate the picking-up rate of each recommendation point. Moreover, a PCD model is used to measure the performance of the candidate route consisting of a sequence of recommendation points. In addition, we find the monotonically increasing property of the PCD and according to the monotone property, we present the PS-Tree data structure with a pruning algorithm to get the optimal route and speed up the search process. Finally, we propose a recommendation algorithm for multiple taxis to assign an appropriate route to each cab, which can minimize the overall potential cruising distance.

IV. ROUTE RECOMMENDATION

A. RECOMMENDING POINT GENERATION

There are two main steps for generating recommending points. The first step is clustering the pickup points in a given time period for all the experienced driver. Second, calculate the pickup rate at this clustered point.

The clusters are based on the driving distance between pick-up points. The driving distance is calculated by using the Google Map's API.

It is reasonable to define experienced drivers who have a high occupancy rate and plenty of driving time. In general, there are three statuses for driving: occupied, cruising and out-of-service. Driver's driving time is the time exclude out-of-service, and the occupancy rate is the ratio of occupancy time to total driving time. It is reasonable to define out-of-service as when two continuous GPS reading is too long. In practice, it cannot be counted as driving time.

Based on the historical pick-up points at a given time period, a programme CLUTO [29] is used to cluster these pick-up points based on driving distance. And the center of the cluster is pick-up points C_i.

The probability at a recommended point is defined as the ratio of the number of taxis, which pick up passengers in the region to the number of all unoccupied taxis entering

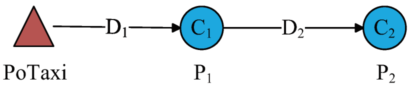


FIGURE 4. An example of a recommended cruising route.

the cluster region. And a cluster region is defined as a circle whose centroid is the cluster center and radius is the average distance of all the point to center.

So the pickup rate at C_i is defined as follow: $P(C_i, t_p) = \frac{|status(cruising \to occupied)|}{|status(cruising)|}$ where $|status(cruising)|$ denotes the number of cruising taxis which passed cluster C_i at time period t_p, and $|status(cruising \to occupied)|$ is the number of these cruising taxis which passed by cluster c at time period tp and changed their state from cruising to occupied.

B. PCD MODEL

The recommendation started with searching for a route with the minimal PCD, which is the optimal route. In this section, we will first show the PCD model. Then we demonstrate the monotonicity of the model function. Then based on the monotonicity, the PS-Tree with the pruning algorithm is proposed to speed up the process of getting the optimal route.

1) THE POTENTIAL CRUISING DISTANCE

Suppose there is a taxi at $PoTaxi$, and follow the route $R = \{PoTaxi, C_1, C_2, \ldots, C_n\}$ at time period tp. The probability of a taxi pick up a passengers at C_i is as follow:

$$P(C_i|R, tp) = \begin{cases} P(C_i, tp) & \text{if } i = 1; \\ P(C_i, tp)\prod_{j=1}^{i-1}(1 - P(C_j, tp)) & \text{if } i > 1. \end{cases}$$

The PCD function is defined as (1), as shown at the bottom of the next page.

By simplify the above formula, we can re-write it as

$$PCD = \frac{\sum_{i=1}^{n}\left[P(C_i|R, tp)D(C_{i-1}, C_i)/P(C_i, tp)\right]}{\sum_{i=1}^{n} P(C_i|R, tp)} \quad (2)$$

For practical use, we can define a cutting length k for the route, which means if cruising route reach k, and not pick up a passenger, the driver will apply a new recommendation.

To explain the PCD function, Fig. 4 shows an example of a recommended cruising route ($PoTaxi \to C_1 \to C_2$). In this example, the corresponding probability is $\{P_1, P_2\}$, and the driving distance is $\{D_1, D_2\}$ respectively. The potential driving distance should be $(P_1D_1 + (1-P_1)P_2(D_1+D_2)/P_2)$ while it may have passengers of $(P_1 + (1 - P_1)P_2)$. Therefore, the potential cruising distance for getting a passenger is $\frac{P_1D_1+(1-P_1)(D_1+D_2)}{P_1+(1-P_1)P_2}$.

2) PS-TREE

To get the optimal route for the target taxi driver, we build the Path-Search-Tree (PS-Tree). PS-Tree is a multiway tree which uses the current position of the target taxi as the root node and each of the other nodes is associated with a recommending point. The path from the root node to a leaf node is a candidate route. In the beginning, for a given length of the route, the PS-Tree is equivalent to a full-connected tree. The child nodes of the root node are all of the recommending points. And if there is, these nodes' child nodes are all the recommending points except for its ancestor nodes until the depth of the tree is the length of the route. However, traversing the tree to get the optimal solution is an exponential problem. To speed up the search process, a pruning algorithm is needed.

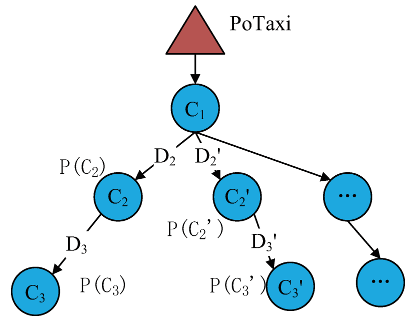


FIGURE 5. Illustration: the route pruning.

To clearly state the monotonicity of the function, we re-write the function as follows: $PCD = \frac{\sum_{i=1}^{n}\prod_{j=1}^{i}\bar{P}(C_{j-1},tp)D(C_{i-1},C_i)}{1-\prod_{i=1}^{n}\bar{P}(C_i,tp)}$

where $\bar{P}(C_i, tp) = 1 - P(C_i, tp)$, and $\bar{P}(C_0, tp) = 1$.

Theorem 1: The PCD function is strictly monotone increasing for each variable $\bar{P}(C_i, tp)$ and $D(C_{i-1}, C_i)$, $(i = 1, 2, \ldots, n)$.

Proof: See Appendix.

In light of the monotonicity of the PCD function, a pruning algorithm is proposed to reduce the search space.

Proposition 1: Let branch paths $R_1 = \{C_i, \ldots, C_k\}$ and $R_2 = \{C'_i, \ldots, C'_k\}$ have the same origin point, that is $C_i = C'_i$, and the same length. The corresponding probability set is $P1 = \{P_i, \ldots, P_k\}$ and $P2 = \{P'_i, \ldots, P'_k\}$ respectively, and the driving distance set is $D1 = \{D_i, \ldots, D_k\}$ and $D2 = \{D'_i, \ldots, D'_k\}$ respectively. We can prune the route R_2 from the PS-Tree if $\forall\, P_j$ in P1 and corresponding P'_j in P2, D_j in D1 and corresponding D'_j in D2, $\overline{P_j} \le \overline{P'_j}$, $D_j \le D'_j$; and $\exists\, P_j$ in P1 and corresponding P'_j in P2, D_j in D1 and corresponding D'_j in D2, there $\overline{P_j} < \overline{P'_j}$ or $D_j < D'_j$.

For example, as shown in Fig. 5, the two paths are $R_1 = \{PoTaxi, C_1, C_2, C_3\}$ and $R_2 = \{PoTaxi, C_1, C'_2, C'_3\}$. If the probability of each pick-up point of R_1 is no less than the corresponding probability of the pick-up point of R_2. And the distance of each road segment (from one pick-up point to another point) of R_1 is not greater than the corresponding distance of the road segment of R_2. Moreover, as Fig. 5 shows, the pruning is irrelevant to the position of a taxi.

Algorithm 1 The Optimal Route

Input: $PoTaxi, tp, k, C_{set}, P_{set}, D_{set}$
Output: The optimal recommending route.

```
1: function GETOPTIMALROUTE( ... )
2:     ShortestCruDist ← +∞;
3:     CandidateRouteSet ← GETCANDROUTESET(k);
4:     for each route in CandidateRouteSet do
5:         PotentialCruDist ← GETPCRUISINGDIST(...);
6:         if ShortestCruDist > PotentialCruDist then
7:             ShortestCruDist ← PotentialCruDist;
8:         end if
9:     end for
10: end function
```

Thus, we can prune the route R_2 from the PS-Tree in advance before starting online recommendation.

C. ROUTE RECOMMENDATION

We first introduce the method for obtaining the optimal PCD route to the target taxi. Once the PCD value of the performance of all the candidate roads is obtained, we can recommend the best one to the taxi given its current location and time. And benefit from the PS-Tree pruning algorithm proposed above, a large number of candidate roads can be eliminated.

Algorithm 1 shows the pseudo-code of the algorithm of getting the optimal route. Given the current location and time of a taxi; we can first get the set of the recommended cluster nodes (C_{set}) of the current time, the set of probability P_{set} and the set of driving distance (D_{set}) at the current time. Then, before starting recommendation, using the PS-Tree pruning algorithm, all of the possible candidate route with length k can be gotten by the function *GetCanRouteSet*() in line 3. The PCD function is calculated for all the route, which is implemented in line 5. Finally, calculate all distance, we can get an optimal route with minimum PCD.

A better recommendation method for multiple taxis in some area is to recommend the current best route one by one. In this way, the effect of the previous recommendation to the current is taken into account. After a passenger was taken at one point, the capacity and picking-up rate of that point would be changed and that change will affect the next recommendation.

After obtaining the set of the probability and the capacity of the pick-up points and the driving distance matrix of these pick-up points during the period. We can get top N candidate routes of length k with the smallest PCD at all position. Assume there are M empty taxies in one area. For each taxi in the area, we assign the current optimal route to the taxi. The potential cruising distance calculated by the model is the driving distance of the taxi. Then we update the capacity and the picking-up rate of each pick-up point along the optimal route. Based on the new capacity and probability of the pick-up points, the currently optimal route will be calculated and updated.

It is uncertain whether the taxi is possible to carry a passenger in one position, but there is a probability to carry a passenger. We update the capacity of the pick-up points of the route by considering the probability of having a passenger. Then the probability updates with the change of the capacity.

Assume there are M taxis at a position *PoTaxi*, and an optimal route $R = \{PoTaxi, C_1, C_2, \ldots, C_n\}$ at time period tp. The corresponding capacity set is $\{V_1, V_2, \ldots, V_n\}$ and the corresponding probability set is $\{P_1, P_2, \ldots, P_n\}$. After we assign one taxi to this route, the new capacity of the pick-up points will be V_i':

$$V_i' = V_i - S_i \tag{3}$$

$$S_i = \begin{cases} P_1, & \text{if } i = 1 \\ (1 - \sum_{1}^{i-1} S_i)P_i, & \text{if } i > 1. \end{cases} \tag{4}$$

In Eq. 3 and Eq. 4, S_i is the number of reduced capacity of the $i-th$ pick-up point. V_i' decreases associated with the probability. The new probability of the pick-up points will be P_i':

$$P_i' = \frac{P_i V_i'}{V_i} \tag{5}$$

where V_i represents the last calculated capacity of the pick-up point. As we can see, after the optimal route assigning to one taxi, the capacity and the picking-up rate of the pick-up points along the optimal route will change. For each recommendation, we will update the capacity and the picking-up rate of the pick-up points to obtain the new optimal route to assign to the next taxi.

Algorithm 2 shows the pseudo-code of the recommendation algorithm for multiple taxis in a different position. In order to reduce the total potential cruising distance and consider the load balancing, we allocate the current optimal route to the coming empty taxi under the circumstance of updating the optimal route in real time. As can be seen, for each position (pos), and for each taxi (t) in the position, we get the current optimal route, and then calculate the driving distance along the optimal route by using the function *GetRecDistance*() based on the *PCD* model we proposed

$$PCD = \frac{\sum_{i=1}^{n-1}(P(C_i|R,tp)\sum_{j=0}^{i-1}D(C_j,C_{j+1})) + P(C_n|R,tp)\sum_{j=0}^{n-1}D(C_j,C_{j+1}) \Big/ P(C_n,tp)}{\sum_{i=1}^{n}P(C_i|R,tp)} \tag{1}$$

Algorithm 2 The Recommendation Algorithm

Input: M_{set}, PoT_{set}, tp, k, C_{set}, P_{set}, D_{set}, $Capacity$.
Output: $RouteList$, $RecDistance$.

1: **function** GETRECOMMENDINGROUTES(...)
2: $Routes \leftarrow$ GETSORTEDROUTE(...);
3: $Optimal_{R}oute \leftarrow Routes(1)(1)$;
4: $RouteList \leftarrow \emptyset$;
5: $RecDistance \leftarrow 0$;
6: **for** each position pos in PoT_{set} **do**
7: **for** each taxi t in $M_{set}(pos)$ **do**
8: $pos.RouteList += Optimal\_Route$;
9: $recDistance = t.GetRecDistance(\ldots)$;
10: **for** each point p in $Optimal\_Route$ **do**
11: $p.Capacity \leftarrow$ GETNEWCAPACITY($Capacity$, P_{set});
12: $p.Probability \leftarrow$ GETNEWPROBABILITY($Capacity$, P_{set});
13: **end for**
14: $RecDistance += recDistance$;
15: $Optimal_{R}oute \leftarrow$ GETOPTIMALROUTE(...);
16: **end for**
17: **end for**
18: **end function**

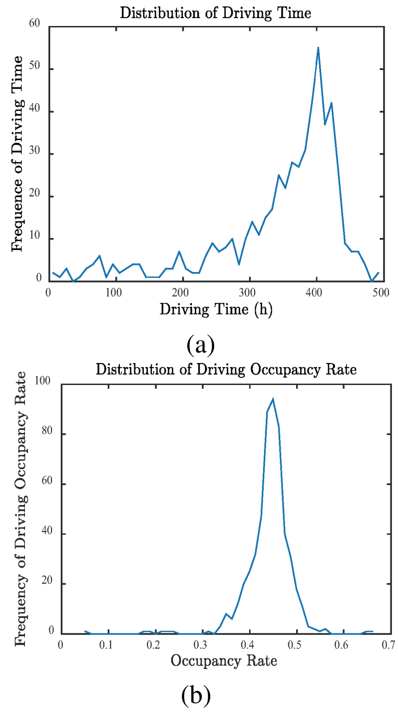


FIGURE 6. Distribution of: (a) Driving time; (b) Driving occupancy rate.

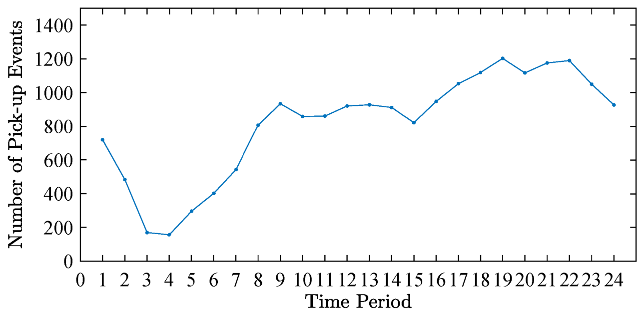


FIGURE 7. The changes in pick-up events over time of experienced drivers. The size of the timeslot is one hour, where 1 stands for 00:00 - 00:59, 2 stands for 01:00 - 01:59, etc.

from line 6 to 9. We update the capacity, picking-up rate of the pick-up points along the optimal route by using the function *GetNewCapacity*() and *GetNewProbability*() from line 10 to 12. Next, the optimal route is updated for the next taxi by using the function *GetOptimalRoute*() in line 15. Finally, we can get the list of routes for each position (*RouteList*) and the total driving distance of all the taxis (*RecDistance*).

V. EXPERIMENTAL RESULTS

A. EXPERIMENTAL SETUP

The dataset we used is a real-world taxi tracer [30], which record more than 500 taxis in a month. Each record is a tuple: (ID, $latitude$, $longitude$, $status$, $timestamp$) The timestamp of our dataset is the GMT time zone, and we translate GMT time to the local time to make the requests and responses more intuitive. For example, Fig. 7 shows the number of pick-up events is low in the period of 3AM to 4AM in San Francisco, which makes sense. In GMT time, however, the lowest number of pick-up events appears about 7AM to 8AM, which is misleading.

The experiments were conducted on a PC with Intel Dual Core i7-8550U CPU and 8G memory with Windows 10 operating system. The algorithms are developed with MATLAB and Python.

Experienced drivers are determined by driving time and occupancy rate. Fig. 6 shows the distributions of the driving time and the driving occupancy rate. Fig. 7 shows the pickup events frequency. And we choose two typical periods, which are 18:00-19:00 and 14:00-15:00. So, we will focus on this two time period in the experiment. All potential pick-up points are clustered into 10 clusters. Table 1 and 2 present the 10 clusters in these two time period, which shows circle region and pickup rate of all clusters. The circle region of the cluster is defined by the center position and radius.

B. EFFICIENCY ANALYSIS

Table 3 lists the results about the number of the candidate routes before and after applying the pruning algorithm for the time periods of $14:00-15:00$ and $18:00-19:00$. It is clear that the number of candidate routes is significantly reduced after pruning.

Table 4 lists the time used to search for the optimal cruising routes with or without applying the pruning algorithm for two time periods: $14:00-15:00$ and $18:00-19:00$. The search time shown here includes all the time for getting the optimal cruising routes. It is evident that the time of using

TABLE 1. Ten clusters of pickup points information during 18:00 - 19:00.

Cluster #	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
size	239	113	214	113	130	135	68	65	55	71
latitude	37.78647	37.80349	37.79091	37.79240	37.76676	37.79456	37.77573	37.77438	37.75038	37.62493
longitude	-122.40942	-122.41193	-122.40027	-122.42260	-122.42574	-122.43721	-122.39663	-122.45873	-122.43327	-122.38711
radius(m)	378.6	584.7	494.3	564.7	889.7	803.7	690.2	1369.6	2109.5	2328.5
$P(C_i)$	0.8795	0.7039	0.8888	0.8713	0.7856	0.7383	0.5831	0.6935	0.8377	0.4419

TABLE 2. 10 clusters of pickup points information during 14:00 - 15:00.

Cluster #	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
size	69	24	42	184	151	59	96	106	47	44
latitude	37.61475	37.70423	37.75358	37.79087	37.78653	37.76744	37.80369	37.79360	37.77130	37.77758
longitude	-122.38618	-122.41852	-122.43296	-122.40164	-122.41253	-122.44328	-122.41476	-122.43607	-122.42185	-122.39720
radius(m)	1964.6	2318.2	2909.6	464.8	524.8	1458.0	653.5	883.6	801.0	649.4
$P(C_i)$	0.4955	0.2967	0.8182	0.7764	0.7974	0.6861	0.6739	0.6852	0.5736	0.5866

TABLE 3. The number of candidate routes before and after pruning.

Length of Route	$14:00-15:00$		$18:00-19:00$	
	Before Pruning	After Pruning	Before Pruning	After Pruning
$k=3$	720=10*9*8	100	720	58
$k=4$	5040=10*9*8*7	509	5040	260
$k=5$	30240=10*9*8*7*6	2326	30240	1562

TABLE 4. Time (in seconds) used to search for the optimal cruising routes with or without pruning.

Length of Route	$14:00-15:00$		$18:00-19:00$	
	w. Pruning	w/o Pruning	w. Pruning	w/o Pruning
$k=3$	0.046728	0.069911	0.044932	0.063406
$k=4$	0.09969	0.424203	0.084591	0.42633
$k=5$	0.531886	2.427247	0.699872	2.41866

pruning is consistently less than those without using pruning. This is in line with our observation of the number of candidate routes.

The pruning algorithm reduces the number of candidate routes to save search time with a significant margin for all different lengths of the optimal cruising route. For example, when k is set to 5 and during $18:00-19:00$, 1,562 routes will be searched with pruning while 30,240 routes to be searched without pruning algorithm. It shows that about 94.8% of the candidate routes can be eliminated. Although we only show the result of these two periods, the trend is similar in other time periods.

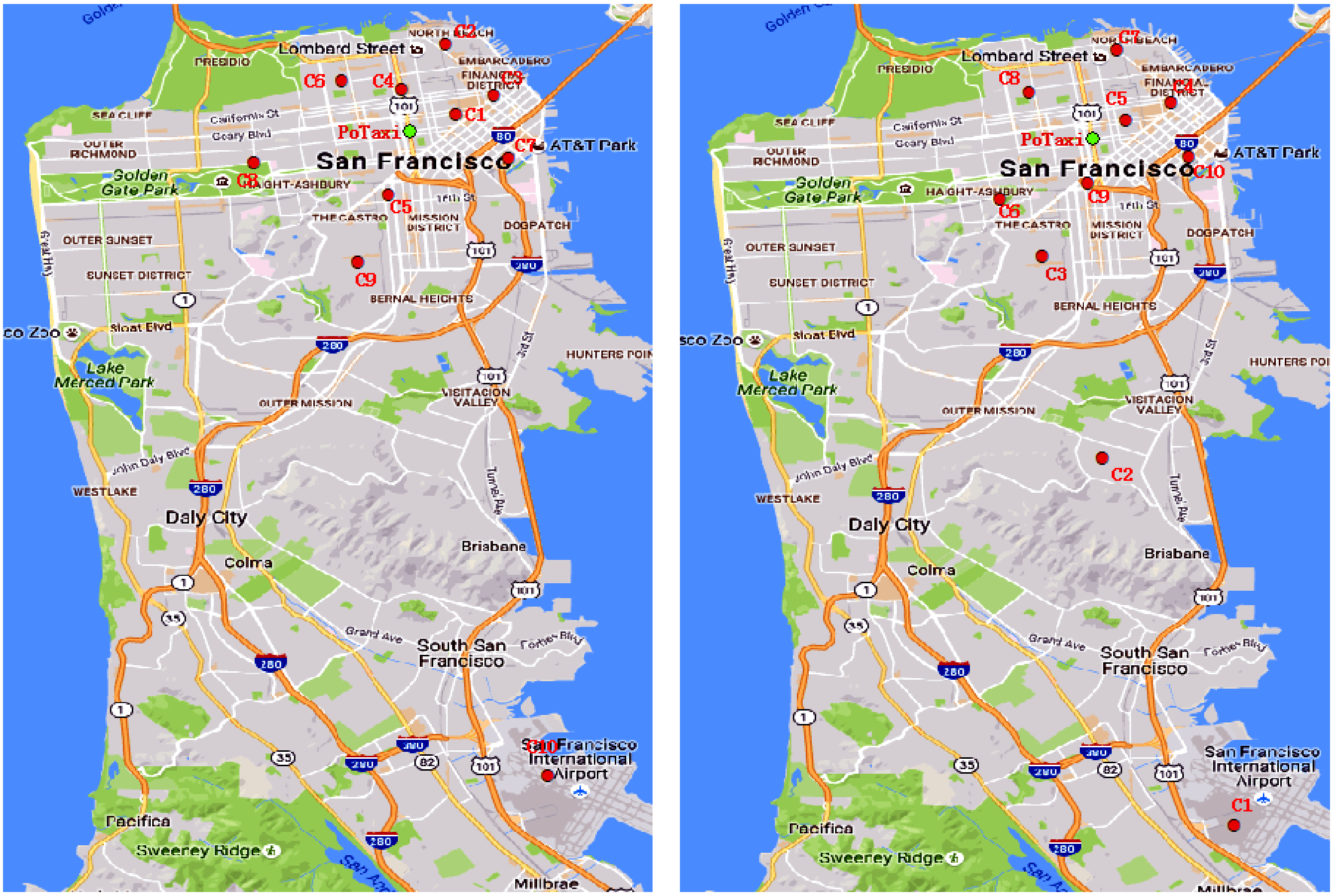
C. MODEL EFFECTIVENESS

This section compares the PCD and the PTD method [10]. Fig. 8 shows a typical recommendation scenery during period $18:00-19:00(a)$ and $14:00-15:00(b)$. The red points represent the cluster centroids of an experienced driver, and the green point is a vacant taxi require a recommendation. Table 5 shows the recommendation result for PCD and PTD with a route limits to 3 section. The result shows that the final PCD is less than PTD, and the recommended route only differs for the last location, and for PCD the distance to the last location is short, there is a higher probability to take a passenger. So it is clear that PCD recommendation is better than PTD recommendation. Table 6 shows very similar result for time period 14:00-15:00.

D. EVALUATION ON RECOMMENDATION

Suppose there are P positions and M taxis at each position in one time period. Based on the recommendation algorithm, we obtain a list of routes for each position. For a taxi at a position, and for every point along the recommended route, we randomly generate a number of 0 or 1 which means having a passenger or not according to pick-up ratio at that point. Subsequently, updating the capacity of each point along the route. If the number is 1, meaning that the taxi takes a passenger at this point, and then the capacity of the point should subtract one, ending the recommendation trip of the current taxi. If the number is 0, meaning that the taxi passes the point without picking up a passenger, and the capacity remains the same, the taxi continues to the next point. And if the capacity of one point becomes zero, the taxi continue to the next point. After knowing whether the taxi is taking a passenger and where the taxi is taking the passenger, the cruising driving distance is calculated by simulation. Through multiple simulations, the expecting cruising driving distance for pickup a passenger is obtained by averaging the results of the simulations.

Here, we evaluate the proposed recommendation method by simulating the average cruising driving distance based on the set of path generated by our recommendation methods and the baseline method. The baseline method uses a Round Robin algorithm to make a recommendation for multiple taxis. The baseline method selects top candidate routes of length k with the smallest potential cruising driving distance for each position and then recommends routes in the circle list to the coming empty taxis in turn. While our method selecting top candidate routes (different from the baseline method) of length k with the smallest potential cruising driving distance for each position and then assign the current optimal route to the coming empty taxi. The average cruising driving distance is the ratio of the total cruising driving distance traveled by



(a) (b)

FIGURE 8. Route recommendation. (a) 18:00-19:00; (b) 14:00-15:00. The red points are the potential pick-up points, and the green one is the location of the taxi, which waiting for the recommendation.

TABLE 5. A typical recommendation during 18:00 - 19:00.

Model	Recommend route $k = 3$	P(C_{i_3})	D(C_{i_2}, C_{i_3})	Distance	C_{i_3}
PCD	$PoTaxi \to C_1 \to C_3 \to C_4$	0.8713	2335	1336	C_4
PTD	$PoTaxi \to C_1 \to C_3 \to C_7$	0.5821	2643	1350	C_7

TABLE 6. A typical recommendation during 14:00 - 15:00.

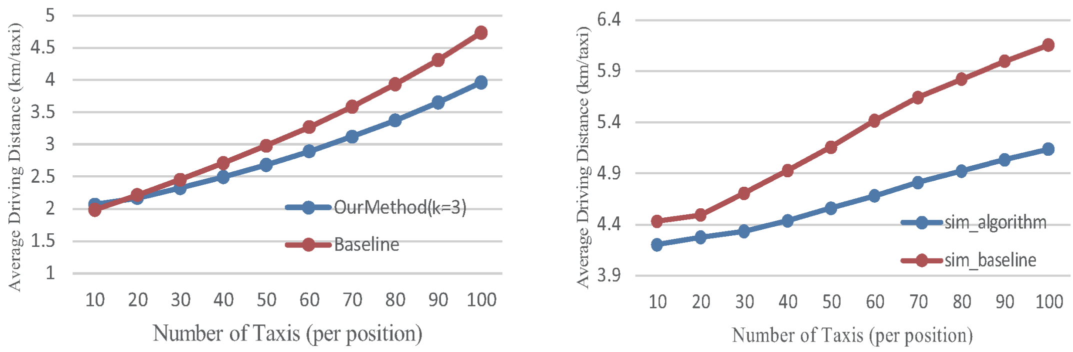
Model	Recommend Route $k = 3$	P(C_{i_3})	D(C_{i_2}, C_{i_3})	Distance	C_{i_3}
PCD	$PoTaxi \to C_5 \to C_4 \to C_7$	0.6739	2105	1324	C_7
PTD	$PoTaxi \to C_5 \to C_4 \to C_{10}$	0.5865	2331	1340	C_{10}

all the taxis in one certain time period to the number of taxis.

First of all, based on our method and the baseline method, we can get the average cruising driving distance of the virtual taxis by these two methods. After getting the list of routes for the taxis of each position by the two algorithms, let these taxis adopt the two recommended route lists for simulation experiments. Finally, through the simulation process to get the simulated results of the average cruising driving distance.

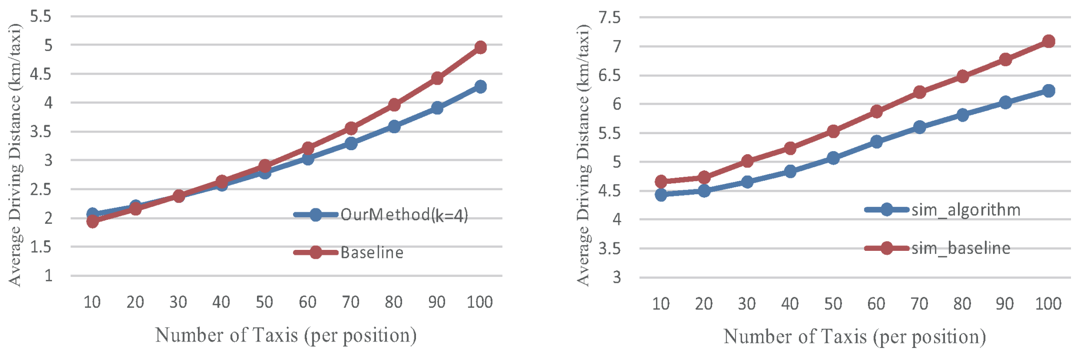
The experiment is carried out by selecting a different number of position and a different number of taxis at each location. For example, assume there are taxis at two position need recommendation service. Then we can choose two different positions and the number of taxis at each location many times to start our experiment. For each time, the list of routes for the taxis of the two position generated by our proposed recommendation method and the baseline method is obtained. Then, let these taxis follow the corresponding recommended route. Averaging driving distance before taking a passenger, we can get the result.

Fig. 9(a) shows the average cruising driving distance based on our method and the baseline method on a different number



(a) (b)

FIGURE 9. **(a) The average cruising distance based on our method and the baseline method on a different number of taxis at 4 positions. (b) The simulation results on average cruising distance based on the path generated by our recommendation methods and the baseline method.**



(a) (b)

FIGURE 10. **(a) The average cruising distance based on our method and the baseline method on a different number of taxis at 4 positions. (b) The simulation results on average cruising distance based on the path generated by our recommendation methods and the baseline method.**

of taxis at 4 location within the time period $18:00-19:00$ when the length of the route $k=3$. Fig. 9(b) shows the corresponding simulation results.

Fig. 10(a) shows the average cruising driving distance based on our method and the baseline method on a different number of taxis at 4 location within the time period $18:00-19:00$ when the length of the route $k=4$. Fig. 10(b) shows the corresponding simulation results.

Note that, in this example, we select top 45 candidate routes for each different position in our method and select top 5 candidate routes for each different location in the baseline method in advance with the purpose of speeding up the recommendation process. It is shown that the proposed method is better than the baseline method, especially when the number of taxis at each location is large.

VI. CONCLUSION

Considering the dynamic change of the pick-up rate caused by a previous recommendation, in this paper, a recommendation algorithm is designed to minimize the global vacant taxi driving distance. The system has the ability to recommend routes for multiple taxis at different locations. The system is based on *PCD* evaluation model [13]. And we proved the monotonicity of PCD. Based on the monotonicity and PS-Tree data structure, a pruning algorithm is proposed to speed up the searching process for getting the optimal cruising route. Moreover, we observe that the picking-up rate will vary with the capacity of the passengers along the route after having a customer and the current recommendation have an impact on the following recommendations. Therefore, we put forward a new recommendation algorithm for multiple taxis to make an effective recommendation. The algorithm takes the change of the pick-up probability into account and considers the impact of the current recommendation on the next. Finally, the simulation results show that the proposed algorithm is more efficient than the baseline method, especially when the number of taxis of each position becomes large.

As a possible future study, evaluation with extensive datasets for an understanding of the practical usability enables the integration to the smart city applications. In addition, we plan to generate service requests at random and determine which taxi responds to a request according to the exact taxi arriving time. Alternatively, an algorithm that handles real-time traffic and requests is worth of investigation. This

$$\lim_{h\to 0^-}\frac{PCD(\bar{P}(C_k,tp)+h)-PCD(\bar{P}(C_k,tp))}{h} = \frac{\sum_{i=k}^{n}\frac{\prod_{j=1}^{i-1}\bar{P}(C_j,tp)}{\bar{P}(C_{k-1},tp)}D(C_{i-1},C_i)+(\sum_{i=1}^{k-1}\prod_{j=1}^{i}\bar{P}(C_{i-1},tp)D(C_{i-1},C_i))(\frac{\prod_{i=1}^{n}\bar{P}(C_{i-1},tp)}{\bar{P}(C_k,tp)})}{(1-\prod_{i=1}^{n}\bar{P}(C_i,tp))(1-\prod_{i=1}^{n}\bar{P}(C_i,tp))} \tag{6}$$

$$\lim_{h\to 0^+}\frac{PCD(\bar{P}(C_k,tp)+h)-PCD(\bar{P}(C_k,tp))}{h} = \frac{\sum_{i=k}^{n}\frac{\prod_{j=1}^{i-1}\bar{P}(C_j,tp)}{\bar{P}(C_{k-1},tp)}D(C_{i-1},C_i)+(\sum_{i=1}^{k-1}\prod_{j=1}^{i}\bar{P}(C_{i-1},tp)D(C_{i-1},C_i))(\frac{\prod_{i=1}^{n}\bar{P}(C_{i-1},tp)}{\bar{P}(C_k,tp)})}{(1-\prod_{i=1}^{n}\bar{P}(C_i,tp))(1-\prod_{i=1}^{n}\bar{P}(C_i,tp))} \tag{7}$$

$$\frac{\sum_{k=i+1}^{n}\frac{\prod_{j=1}^{k}\bar{P}(C_{j-1},tp)}{\bar{P}(C_i,tp)}D(C_{k-1},C_k)+\frac{\prod_{j=1}^{n}\bar{P}(C_j,tp)}{\bar{P}(C_i,tp)}\sum_{k=1}^{i}\prod_{j=1}^{k}P(C_{j-1},tp)D(C_{k-1},C_k)}{(1-\prod_{j=1}^{n}\bar{P}(C_j,tp))(1-\prod_{j=1}^{n}\bar{P}(C_j,tp))} \tag{10}$$

is particularly important for distributed systems with limited computing power in automobiles.

APPENDIX
PROOF OF MONOTONICITY OF THE PCD MODEL

Note that the function PCD is an elementary function for each variable within its domain, therefore, it is continuous for the variables within the domain. We only to prove that the PCD is derivable within the open interval of the domain and the corresponding derivative is greater than or equal to zero.

For each $\bar{P}(C_i, tp)$, $(i = 1, 2, \ldots, n)$, we can calculate the left and the right limit of the function (6) and (7), as shown at the top of this page:

And the same for $D(C_{i-1}, C_i)$, $(i = 1, 2, \ldots, n)$:

$$\lim_{h\to 0^-}\frac{PCD(D(C_{k-1},C_k)+h)-PCD(D(C_{k-1},C_k))}{h} = \prod_{i=1}^{k}\bar{P}(C_{i-1},tp)\Big/\left(1-\prod_{i=1}^{n}\bar{P}(C_i,tp)\right) \tag{8}$$

$$\lim_{h\to 0^+}\frac{PCD(D(C_{k-1},C_k)+h)-PCD(D(C_{k-1},C_k))}{h} = \prod_{i=1}^{k}\bar{P}(C_{i-1},tp)\Big/\left(1-\prod_{i=1}^{n}\bar{P}(C_i,tp)\right) \tag{9}$$

From the above, we can obtain that Eq. 6 is equal to Eq. 7 and Eq. 8 is equal to Eq. 9. So, the function PCD is derivable for each variable $\bar{P}(C_i, tp)$ and $D(C_{i-1}, C_i)$ $(i = 1, 2, \ldots, n)$ within the open interval of the domain. The derivative of the function for $\bar{P}(C_i, tp)$ can be calculated as (10), as shown at the top of this page:

The derivative of the function for $D(C_{i-1}, C_i)$ can be calculated as:

$$\frac{\prod_{j=1}^{i}\bar{P}(C_{j-1},tp)}{1-\prod_{i=1}^{n}\bar{P}(C_i,tp)} \tag{11}$$

We can prove that Eqs. 10 and 11 are greater than or equal to zero. In summary, the potential cruising distance function is strictly monotone increasing for each variable $\bar{P}(C_i, tpr)$ and $D(C_{i-1}, C_i)$, $(i = 1, 2, \ldots, n)$.

REFERENCES

[1] T. Hunter, R. Herring, P. Abbeel, and A. Bayen, "Path and travel time inference from GPS probe vehicle data," *NIPS Analyzing Netw. Learn. Graphs*, vol. 12, no. 1, 2009.

[2] Y. Ye *et al.*, "Geographically weighted regression model for urban traffic black-spot analysis," in *Proc. IGARSS*, Fort Worth, TX, USA, Jul. 2017, pp. 4866–4869.

[3] Y. Zhang, Z.-J. M. Shen, and S. Song, "Parametric search for the bi-attribute concave shortest path problem," *Transp. Res. B, Methodol.*, vol. 94, pp. 150–168, Dec. 2016.

[4] R. Zhang, F. Rossi, and M. Pavone, "Model predictive control of autonomous mobility-on-demand systems," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1382–1389.

[5] P. Chen and Y. Nie, "Bicriterion shortest path problem with a general nonadditive cost," *Procedia-Social Behav. Sci.*, vol. 80, pp. 553–575, Jun. 2013.

[6] X. Kong, F. Xia, J. Wang, A. Rahim, and S. K. Das, "Time-location-relationship combined service recommendation based on taxi trajectory data," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1202–1212, Jun. 2017.

[7] R.-H. Hwang, Y.-L. Hsueh, and Y.-T. Chen, "An effective taxi recommender system based on a spatio-temporal factor analysis model," *Inf. Sci.*, vol. 314, pp. 28–40, Sep. 2015.

[8] W. Yang, X. Wang, S. M. Rahimi, and J. Luo, "Recommending profitable taxi travel routes based on big taxi trajectories data," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Ho Chi Minh City, Vietnam: Springer, 2015, pp. 370–382.

[9] H. Hu, Z. Wu, B. Mao, Y. Zhuang, J. Cao, and J. Pan, "Pick-up tree based route recommendation from taxi trajectories," in *Proc. Int. Conf. Web-Age Inf. Manage.* Harbin, China: Springer, 2012, pp. 471–483.

[10] Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, and M. Pazzani, "An energy-efficient mobile recommender system," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 899–908.

[11] M. Qu, H. Zhu, J. Liu, G. Liu, and H. Xiong, "A cost-effective recommender system for taxi drivers," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 45–54.

[12] M. Elhoseny, A. Shehab, and X. Yuan, "Optimizing robot path in dynamic environments using genetic algorithm and bezier curve," *J. Intell. Fuzzy Syst.*, vol. 33, no. 4, pp. 2305–2316, 2017.

[13] H. Lv, F. Fang, Y. Zhao, Y. Liu, and Z. Luo, "A performance evaluation model for taxi cruising path recommendation system," in *Advances in Knowledge Discovery and Data Mining* (Lecture Notes in Computer Science), vol. 10235. Jeju, South Korea: Springer, 2017.

[14] J. Letchner, J. Krumm, and E. Horvitz, "Trip router with individualized preferences (TRIP): Incorporating personalization into route planning," in *Proc. Nat. Conf. Artif. Intell.*, Cambridge, MA, USA: MIT Press, vol. 21, no. 2, 2006, pp. 1795–1800.

[15] N. Malviya, S. Madden, and A. Bhattacharya, "A continuous query system for dynamic route planning," in *Proc. IEEE 27th Int. Conf. Data Eng. (ICDE)*, Apr. 2011, pp. 792–803.

[16] B. Li *et al.*, "Hunting or waiting? Discovering passenger-finding strategies from a large-scale real-world taxi dataset," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PERCOM Workshops)*, Mar. 2011, pp. 63–68.

[17] J. Yuan, Y. Zheng, L. Zhang, X. Xie, and G. Sun, "Where to find my next passenger," in *Proc. 13th Int. Conf. Ubiquitous Comput.*, 2011, pp. 109–118.

[18] X. Li *et al.*, "Prediction of urban human mobility using large-scale taxi traces and its applications," *Frontiers Comput. Sci.*, vol. 6, no. 1, pp. 111–121, 2012.

[19] A. Awasthi, Y. Lechevallier, M. Parent, and J.-M. Proth, "Rule based prediction of fastest paths on urban networks," in *Proc. IEEE Intell. Transp. Syst.*, Sep. 2005, pp. 978–983.

[20] H. Gonzalez, J. Han, X. Li, M. Myslinska, and J. P. Sondag, "Adaptive fastest path computation on a road network: A traffic mining approach," in *Proc. 33rd Int. Conf. Very Large Data Bases VLDB Endowment*, 2007, pp. 794–805.

[21] F. Miao, S. Han, A. M. Hendawi, M. E. Khalefa, J. A. Stankovic, and G. J. Pappas, "Data-driven distributionally robust vehicle balancing using dynamic region partitions," in *Proc. 8th Int. Conf. Cyber-Phys. Syst.*, 2017, pp. 261–271.

[22] K. T. Seow, N. H. Dang, and D.-H. Lee, "A collaborative multiagent taxi-dispatch system," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 3, pp. 607–616, Jul. 2010.

[23] F. Miao *et al.*, "Taxi dispatch with real-time sensing data in metropolitan areas: A receding horizon control approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 463–478, Apr. 2016.

[24] R. Zhang and M. Pavone, "Control of robotic mobility-on-demand systems: A queueing-theoretical perspective," *Int. J. Robot. Res.*, vol. 35, nos. 1–3, pp. 186–203, Jan. 2016.

[25] S. Ma, Y. Zheng, and O. Wolfson, "T-share: A large-scale dynamic taxi ridesharing service," in *Proc. IEEE 29th Int. Conf. Data Eng. (ICDE)*, Apr. 2013, pp. 410–421.

[26] S. Ma, Y. Zheng, and O. Wolfson, "Real-time city-scale taxi ridesharing," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 7, pp. 1782–1795, Jul. 2015.

[27] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003.

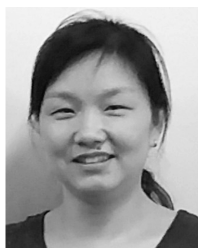
[28] I. Trestian, K. Huguenin, L. Su, and A. Kuzmanovic, "Understanding human movement semantics: A point of interest based approach," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 619–620.

[29] T. Andrea and K. George, "A segment-based approach to clustering multi-topic documents," in *Proc. SIAM Datamining Conf. Text Mining Workshop*, 2008.

[30] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser. (Feb. 2009). *CRAWDAD Dataset EPFL/Mobility (V. 2009-02-24)*. [Online]. Available: https://crawdad.org/epfl/mobility/20090224

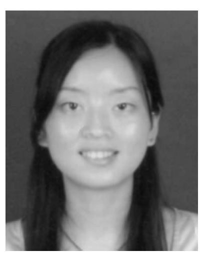
ZHONGWEN LUO received the B.S. degree in mathematics from Wuhan University, China, in 1985, and the M.E. degree in mechanics from the Institute of Mechanics, Academia Sinica, China, in 2009. He was a Lecturer with the Institute of Mechanics, Academia Sinica, from 1990 to 1995. He was with the Institute of Information Engineering, China University of Geosciences, as the Associate Dean, from 2004 to 2017. He has been serving the Faculty of Information Engineering, China University of Geosciences, as an Assistant Professor, an Associate Professor, and a Professor, since 1993, where he has been the Director of the AI Institute since 2010. His research interests include AI, robotics, and high-performance computation and networks.

HUIMIN LV received the B.S. and M.S. degrees in software engineering from the China University of Geosciences, China, in 2015 and 2018, respectively. Her research interests include artificial intelligence, scheduling, GIS applications, and network management.



FANG FANG (M'17) received the B.S. degree in computer science and technology and the Ph.D. degree in management science and engineering from the China University of Geosciences, Wuhan, China, in 1998 and 2012, respectively. She is currently an Associate Professor with the College of Information Engineering, China University of Geosciences. Her research interests include spatial data mining, machine learning, and GIS applications. She is a member of the CCF.

YISHI ZHAO received the M.Sc. (Hons.) and Ph.D. degrees in computer science from Newcastle University in 2006 and 2011, respectively. He is currently a Lecturer with the Faculty of Information Engineering, China University of Geosciences, Wuhan. His research interests lie in performance modeling and performance evaluation, in particular Markov modeling through queuing theory, stochastic process algebra, and discrete event simulation.



YUANYUAN LIU received the B.E. degree from Nanchang University, Nanchang, China, in 2005, the M.E. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2007, and the Ph.D. degree from Central China Normal University. She is currently a Lecturer with the China University of Geosciences, Wuhan. Her research interests include image processing, computer vision, and pattern recognition.

XIUQIAO XIANG received the Ph.D. degree in systems analysis and integration from the Huazhong University of Science and Technology, China, in 2009. He was a Math Post-Doctoral Fellow with the Huazhong University of Science and Technology from 2009 to 2011. After that, he was a Lecturer with the College of Information Engineering, China University of Geosciences, where he has been an Assistant Professor since 2012. His current research interests include computer vision, image processing, compressed sensing, machine learning, and deep leaning.

XIAOHUI YUAN (S'01–M'05–SM'16) received the B.S. degree in electrical engineering from the Hefei University of Technology, China, in 1996, and the Ph.D. degree in computer science from Tulane University in 2004. After his graduation, he was with the National Institutes of Health until 2006. He is currently an Associate Professor with the University of North Texas. His research interests include computer vision, data mining, machine learning, and artificial intelligence. He was a recipient of the Ralph E. Powe Junior Faculty Enhancement Award in 2008 and the Air Force Summer Faculty Fellowship in 2011, 2012, and 2013.