2014 28th International Conference on Advanced Information Networking and Applications Workshops

# Game Theory Based Recommendation Mechanism for Taxi-Sharing

Sheng-Tzong Cheng, Jian-Pan Li and Gwo-Jiun Horng
Department of Computer Science and Information Engineering, National Cheng Kung University
Tainan, Taiwan
e-mail: stcheng@mail.ncku.edu.tw

***Abstract*—This paper presents a recommendation mechanism for taxi-sharing. The first aim of our model is to respectively recommend taxis and passengers for picking up passengers quickly and finding taxis easily. The second purpose is providing taxi-sharing service for passengers who want to save the payment. In our method, we analyze the historical Global Positioning System (GPS) trajectories generated by 10,357 taxis during 110 days and present the service region with time-dependent R-Tree. We formulate the problem of choosing the paths among the taxis in the same region by using non-cooperative game theory, and find out the solution of this game which is known as Nash equilibrium. The results show that our method can find taxis and passengers efficiently. In addition, applying our method can reduce the payment of passengers and increase the taxi revenue by taxi-sharing.**

***Keywords—taxi-sharing; trajectory; recommendation mechanism; non-cooperative game theory***

## I. INTRODUCTION

With the rapid development of city, taxis play a vital role in modern public transportation networks. People always suffer from waiting a long time for a taxicab. In fact, taxi drivers also have similar problems for finding passengers. They have no idea where the passengers are but to cruise on the roads at random. Cruising on the roads blindly wastes the time and gas, and decreases the profit of a taxi driver. Besides, many vacant taxis cruising on the roads incur additional traffic and produce more pollution in the city. Therefore, it's an urgent problem how to improve the utilization of taxis and to save energy. Sometimes passengers are not willing to take the taxicab because of expensive fare unless in an emergency. They would rather take other public transportation system such as bus, train and MRT. Ridesharing is a solution to reduce the energy consumption and the taxi fare of passengers. There are many kinds of collective transport such as car-pooling [1–2], which is a widespread ridesharing way using private vehicles in the United State and Europe.

Using the knowledge of taxi's GPS trajectories and passengers' mobility patterns, we can learn about the places where taxis always pick up passengers. The proposed method works in three phases: cognition, inference and recommendation. We first collect a large number of information concerning taxis and passengers. Second, we analyze the historical trajectories of taxis and represent the service region by a time-dependent R-Tree[3–5]. Game theory [6–8] is the study of mathematical models of conflict and cooperation between intelligent rational decision-makers.

There are many existing works on the taxi service. The development of taxi recommender and ridesharing systems are introduced as follows.

### A. Taxi Recommender System

Taxi dispatch and recommender systems are getting more and more attention from researchers with the development of intelligent transportation systems [9]. The most common method used for getting a taxi is directly hailing one from the side of the street. Although this is a trivial way, customers waiting for a taxi depend on luck. Sometimes people have to wait for a vacant taxi for a long time.

In [3], the authors proposed a system that suggests parking places based on a large number of GPS trajectories generated by taxis. These parking places are locations where taxi drivers wait for passengers. The taxi recommender system devises a probabilistic model to formulate the behavior of taxis, and provides the recommendation services to both taxis and passengers without considering the issue of taxi-sharing.

### B. Taxi Ridesharing System

With the development of society, ridesharing such as car-pooling [1] has become an important mode of mobility. The aim of ridesharing is to reduce the number of vehicles on the road and to share the cost of a trip. Taxi-sharing is the same concept as car-pooling but relies on taxis as the transportation resource. It is more flexible to apply taxi-sharing than car-pooling. However, the cooperation and coordination between taxis presents a real challenge because of the independency and self-employment of taxi drivers.

There are already some existing systems developed for this purpose. In [10] proposes a dynamic (no reservations and immediate assignments) and distributed (no centralized authority) taxi-sharing system that can coordinate between user requests and taxi services. This paper presents the concept of a distributed algorithm for taxis and customers, but does not detail these methods.

In this paper, we present a recommendation mechanism for not only finding taxis and passengers but also for providing a taxi-sharing service. The rest of paper is organized as follows: Section II describes the assumptions made as well as the system model. Section III presents the recommendation phase. Section IV presents the simulation results. Finally, our conclusion is presented in Section V.

978-1-4799-2652-7/14 $31.00 © 2014 IEEE
DOI 10.1109/WAINA.2014.106



IEEE computer society

## II. SYSTEM MODLE

### A. Overview

We assume that taxicabs are equipped with GPS sensors. By some weight sensors or the taxi meter, taxis can record occupancy information and send trajectories with timestamps to the server with a certain frequency. Additionally, all of the taxis and passengers are identified.

As Fig. 1 shows, there are many waiting places which represent popular locations for picking up passengers and waiting for taxis in our environment. On the one hand, taxicabs are cruising on the roads to find passengers and pick them up. On the other hand, passengers are waiting for taxis in our service region.

System architecture is presented in Fig. 2. The server collects historical data including taxis' and passengers' information for analysis. When the taxis and passengers need recommendations, the server replies to them with corresponding information.

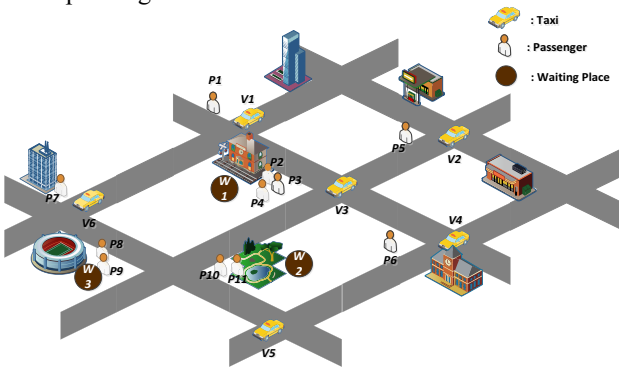


Figure 1. Schematic diagram

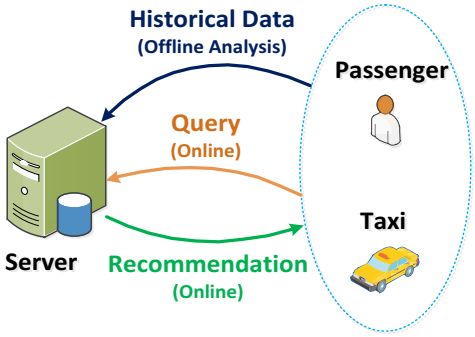


Figure 2. System architecture

The process of our work comprises three phases including the Cognition Phase, Inference Phase and Recommendation Phase. The procedure of our method is shown in Fig. 3 where each block with dashed lines represents each phase in our process.

### B. Cognition Phase

This phase collects the historical information of taxis and passengers, and preprocesses this data for the Inference Phase. There are two modules representing Taxi Trajectories and Movement Logs for gathering data from taxis and passengers respectively.

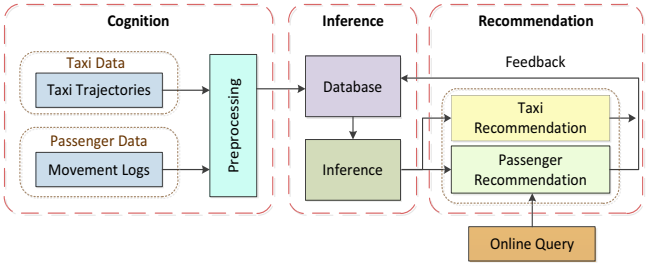


Figure 3. The procedure of our method

### C. Inference Phase

The main objectives in this phase are historical information storage and analysis. This phase consists of two modules, the Database module and Inference module. The database module is responsible for the storage of data from the first phase. Data stored in the Database module are studied by the Inference module. In the Inference module, we figure out waiting places that represent popular locations where passengers are frequently picked up.

We partition the road network and represent the service region in our system by spatial index data structure R-Tree. A point on the road map stands for a GPS trajectory point. The rectangles containing a lot of points represent the regions in an R-Tree. The denser points are in the map, the greater the number of regions (rectangles) generated.

We partition a day into fixed slots (e.g. one hour per slot), and analyze the trajectories to build an R-Tree for each time slot. There are $I$ time slots in a day, and the length of each time slot is $\varepsilon$. The $i_{th}$ R-Tree is represented as $RT^i = \{r_1^i, r_2^i, \ldots, r_j^i\}$, and $r_j^i$ is the region $j$ in R-Tree $RT^i$.

## III. RECOMMENDATION PHASE

In this stage, we provide recommendations to taxis and passengers when they need one. Given the location and time of taxi or passenger, the server finds the proper route or location for the recommendation based on the analysis produced by the Inference module. There are two modules in this phase: the Passenger Recommendation and Taxi Recommendation modules.

### A. Passenger Recommendation

A passenger who needs a recommendation delivers a range query with her location and current time to the server. Accordingly, the server searches the surrounding candidate regions for recommendations in the range of distance threshold $D_p$. The threshold $D_p$ is a walking distance. By calculating the scores of each candidate region, the server selects the region with the highest score for recommendation. The score function S of candidate region j for the mth passenger can be obtained as follows:

$$S^m(j) = \eta_3 \times (\text{the number of children in } R_j) \times \frac{1}{D_j^m} \quad (1)$$

where $\eta_3$ is a normalized factor to adjust the score, $D_j^m$ is the distance between region j and the location of mth passenger. The example is shown as Fig. 4. The children in the candidate region are the trajectory points

contained in the region of the R-Tree constructed previously. The more children in the region, the more possible it is to find a taxi.

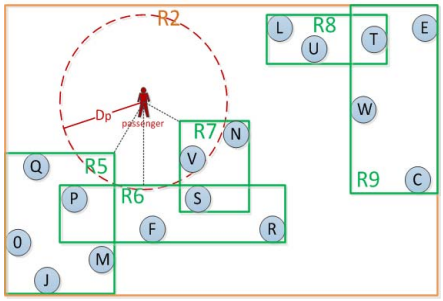

Figure 4. Example of passenger recommendation

## B. Recommendation for One Taxi

For the status of a taxi, we consider four states: cruising ($\mathcal{C}$), parking ($\mathcal{P}$), occupied &shared($\mathcal{O}.\mathcal{S}$) and occupied($\mathcal{O}$). The transition graph is depicted in Fig. 5. At the start, a taxi joins the system; it needs recommendations to find passengers. The server provides some suggestions to the taxi so that the taxi follows the route in the cruising($\mathcal{C}$) state. If the taxi cannot find any passengers along the route, it turns to the parking ($\mathcal{P}$) state and continues to wait. Whenever the taxi picks up a passenger, it changes its state to occupied & shared ($\mathcal{O}.\mathcal{S}$) or occupied($\mathcal{O}$) depending on whether or not the passenger wants to share.

For the taxi recommendation, we perform a range query according to the location and time of the taxi, and retrieve some waiting places. For each waiting place, the server searches the regions of the R-Tree near the taxi and generates the routes. The recommendation routes are constrained by distance threshold$D_t$. If the routes to these waiting places exceed the distance threshold, the server generates the routes based on the corresponding R-Tree. An example of taxi recommendation is shown in Fig. 6.

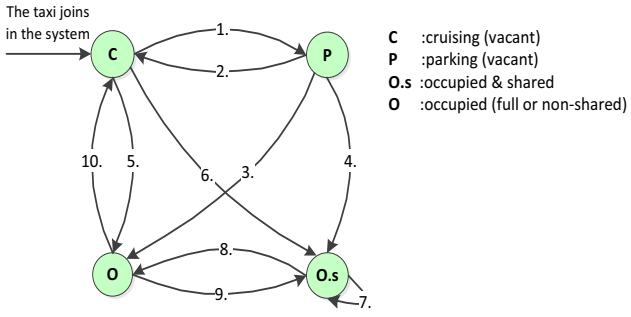

Figure 5.Taxis' Status

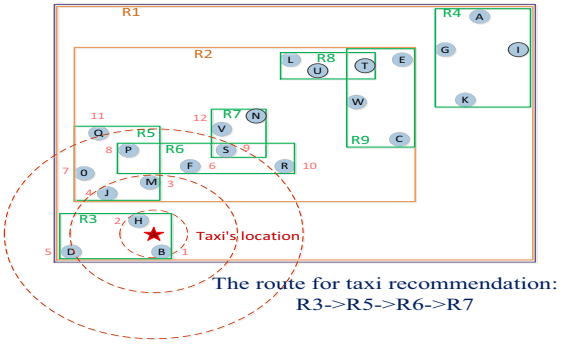

Figure 6. Example of taxi recommendation

## C. Non-cooperative Game Model for Taxi Recommendation

In the previous section, we discussed the recommendations provided for only one taxi. In fact, typically more than one taxi needs a recommendation at the same time. Therefore, in reality we also need to consider a situation in which many taxis need recommendations at the same time. This scenario is illustrated in Fig. 7.

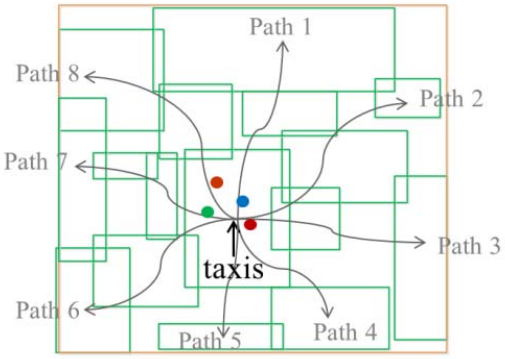

Figure 7. Scenario of recommendation for many taxis

The main objective of the recommendation problem is to maximize the profits of the taxis in the same region by utilizing the concept of equilibrium. A non-cooperative game model can be formulated as follows. The players in this game are the taxis who need route recommendations in the same region at that time. The strategy of each taxi is the choice of route. The payoff for each taxi is the profit (i.e., revenue minus cost) related to probability for picking up a passenger and the cost born by the taxi. The commodity in the scenario is recommendation routes.

Firstly, we assume that there are N taxis in the same region with a request for a recommendation at time t. $Paths_i$for recommendation is considered a strategy of taxi i in the system model. Sdenotes the set of strategies for all taxis in the region (i.e. $\mathbb{S} = \{s_1, \dots s_N\}$).

Based on the assumption described above, we represent the function to calculate the cost that taxis need to bear according to the selection of strategies. This cost function is given by

$$c(\mathbb{S}) = \eta_1 \times \left[\sum_{s_i \in \mathbb{S}} (s_{i.d} + s_{i.t})\right]^{\tau} \quad (2)$$

where $\eta_1$is a normalized factor to adjust the cost, $s_{i.d}$ and $s_{i.t}$ is the distance and travel time for strategy $s_i$, $\tau$ is a non-negative constant, and $\tau \geq 1$(so that the cost function is

convex i.e., there exists an extreme value in this function), and Sdenotes the set of strategies for all taxis in the same region.

The revenue of taxi *i* is

$$r_i = \frac{\mathrm{E}(s_i)}{M(s_i)} = \frac{s_i \times p_{s_i}}{M(s_i)} \quad (3)$$

where $\mathrm{E}(s_i)$ is the expectation (the variable multiplies its corresponding probability) of finding a passenger by choosing the strategy $s_i$, $M(s_i)$ is the number of taxis who choose the strategy $s_i$. Because more than one taxi choose the same route, the probability that a taxi succeeds in picking up a passenger is divided. In other words, the more taxis go the same way, the less likely it will be that a taxi finds a passenger.

As mentioned above, we can present profit by subtracting cost from revenue. Thus, the profit of taxi i can be rewritten as follows:

$$\pi_i(\mathbb{S}) = \eta_2 \times [r_i - s_i \cdot c(\mathbb{S})]$$
$$= \eta_2 \times \left\{ \frac{s_i \times p_{s_i}}{M(s_i)} - s_i \{ \eta_1 \times [\sum_{s_i \in \mathbb{S}} (s_{i.d} + s_{i.t})]^{\tau} \} \right\} \quad (4)$$

where $\eta_2$ is a normalized factor to adjust the profit.

Let $\mathbb{S}_{-i}$ denote the set of strategies adopted by all taxis except taxi i (i.e. $\mathbb{S}_{-i} = \{s_j | j = 1, \dots, N; j \neq i\}$ and $\mathbb{S} = \mathbb{S}_{-i} \cup \{s_i\}$). The optimal recommendation of routes for each taxi depends on the strategies of other taxis. Nash equilibrium is regarded as the solution of the game. The Nash equilibrium of a game is a strategy profile (e.g. lists of strategies in the form of one for each other). The fundamental feature of the Nash equilibrium is that no player can increase his payoff by choosing another strategy according to other players' strategies. This means that no player is willing to change his actions for more profit, given the others' actions.

The Nash equilibrium in this case is obtained by using the best response function, which is the best strategy of one player, given others' strategies. The best response function of taxi I given the strategies of the other taxis $s_j$ where j ≠ i, is defined as follows:

$$\mathcal{BR}_i(\mathbb{S}_{-i}) = argmax_{s_i} \pi_i(\mathbb{S}_{-i} \cup \{s_i\}) \quad (5)$$

The set $\mathbb{S}^* = \{\boldsymbol{s_1^*}, \dots, \boldsymbol{s_N^*}\}$ denotes the **Nash equilibrium** of this game if and only if $s_i^* = \mathrm{BR}_i\left(S_{-i}^*\right)$, ∀i, where $\mathbb{S}_{-i}^*$ denotes the set of the best responses for taxi j for j ≠ i. We can obtain the Nash equilibrium by solving the following equation:

$$\frac{\partial \pi_i(\mathbb{S})}{\partial s_i} = \eta_2 \left\{ \frac{p_{s_i}}{M(s_i)} - \{ \eta_1 [\sum_{s_i \in \mathbb{S}} (s_{i.d} + s_{i.t})]^{\tau} \} - \eta_1 s_i \tau [\sum_{s_i \in \mathbb{S}} (s_{i.d} + s_{i.t})]^{\tau - 1} \right\} = 0 \quad (6)$$

Therefore, the optimization problem with the objective can be defined as follows:

$$\text{Minimize} \sum_{i=1}^{N} |s_i - \mathcal{BR}_i(\mathbb{S}_{-i})| \quad (7)$$

i.e. minimize the sum of the difference between decision variable $s_i$ and the corresponding best response function. The algorithm reaches the Nash equilibrium while the minimum value of the objective function is zero. As a result, the server determines the route recommendation for each taxi in the same region at an arbitrary time.

### *D. Recommendation for Taxi-sharing*

Imagine that passengers $c_1$ and $c_2$ are in taxi v at time t, and a new passenger $c_3$ has sent a taxi-sharing query to the server. Taxi v needs to determine whether passenger $c_3$ is appropriate for taxi-sharing. Note that a trip of the taxi is defined as the duration of time between picking up the first passenger and becoming vacant again. The definition of a trip is depicted in Fig. 8. Besides the original location and destination, every passenger needs to attach information about taxi-sharing, such as the time limit for detouring, the expected value of fare saved and the degree of urgency. The candidate passengers of taxi v for taxi-sharing are filtered through the area in the form of a rectangle. The height of the rectangle is the distance from the current location to the destination of the taxi. The width is a distance threshold $D_s$. The bounded area is illustrated in Fig. 9.

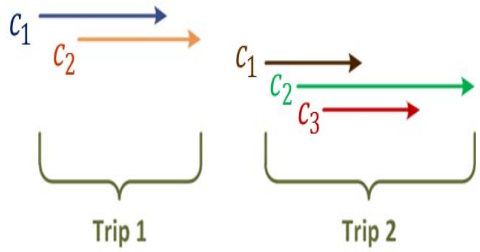

Figure 8. Definition of a trip

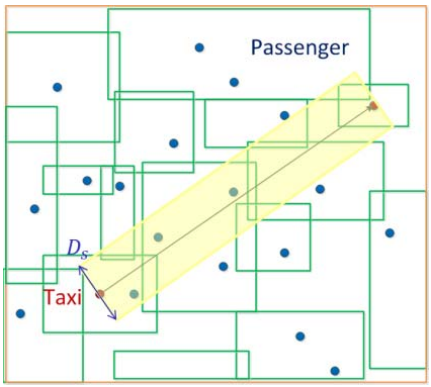

Figure 9. The bounded area of candidate passengers for taxi-sharing

After obtaining the candidate passengers, we propose four rules to check whether a candidate passenger is suitable for ridesharing with on-board passengers in the taxi. Three of these rules are derived from the aspect of passengers, and the last rule is related to the taxi itself.

For each passenger $c_i$ in the taxi at time t, the taxi needs to check the following three rules. The first rule states that the cost of detouring must be less than the profit for taxi-sharing and is described as follows:

**Rule 1**: $\mathrm{Cost_d} \leq \mathrm{Profit_s}$ (8)

If $\mathrm{Cost_d}$ is greater than $\mathrm{Profit_s}$, then the new passenger is not appropriate. The second rule showing the limitation of time for detouring of each passenger can be represented as follows:

**Rule 2**: $$t_d \le t_w(t) \quad (9)$$

$$, t_w(t) = T_w \times (1 - \frac{(t-t_p)}{t_o}) \quad (10)$$

If $t_d$ is greater than $t_w(t)$, then the new passenger is not appropriate.

We consider the payment saved for taxi-sharing in the third rule. Each passenger sets the expected value of payment saved. The benefit from carrying the new passenger needs to fulfill these expectations so that taxi-sharing can achieve its goal of saving money. The third rule is

**Rule 3**: $$f_p < 0 \; and \; |f_p| \ge F_e \quad (11)$$

If $f_p$ is positive or the value of $|f_p|$ is less than $F_e$, then the new passenger is not appropriate. Moreover, we consider the profit of not only passengers but also taxis. A taxi can set the expectation of profit for providing a taxi-sharing service, which is denoted as $P_e$. The new candidate passenger has to allow the taxi to attain more revenue, which is not less than $P_e$. For taxi $v$ at time $t$, the last rule which needs to be checked is

**Rule 4**: $$p_p \ge P_e \quad (12)$$

After all of the rules are satisfied, we can compute the fitness degree of the candidate passengers. The fitness value can be obtained as follows:

$$F_{v[t]}(c_3) = \eta_4 \times p_p \times \sum_{i=1}^{3}(|c_{i.f_p}| \times \frac{1}{c_{i.t_d}}) \quad (13)$$

where $\eta_4$ is a normalized factor to adjust the fitness value, and $p_p$ refers to the profit of taxi v$c_{i.f_p}$ and $c_{i.t_d}$ are the profit and the delay time for the detour required by a passenger. The taxi chooses the passenger by the highest fitness value for taxi-sharing.

If two taxis want to pick up the same passenger for taxi-sharing at the same time, then we can compare the fitness value F of the passenger among these taxis first and the award value A of the taxis if necessary. Once a taxi accepts a passenger for sharing, the award value A increases by 1. The higher award value a taxi has, the more opportunities it has to get a passenger for sharing.

### E. Pricing Scheme

The section considers that payment should be fairly shared by each passenger in the taxi. The previous passengers need to be able to afford the extra payment incurred by the detour. Therefore, the following passengers have to spend more money than the previous one on the distance with taxi-sharing. For the travel distance for taxi-sharing ($d_s$), the payment of passengers $c_1$, $c_2$, $c_3$ is proportional to $\alpha:\beta:\gamma$, where $0 < \alpha < \beta < \gamma$. For example, there are two passengers $c_1$ and $c_2$ for sharingunder the constraint $(\alpha+\beta)r_3 + \alpha(r_4 + r_5) < (\alpha+\beta)r_2 + \beta r_6$ as shown in Fig. 10.

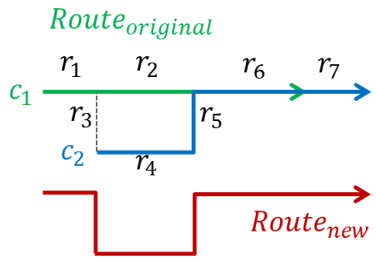


Figure 4. Example of pricing scheme for two passengers

The distance for sharing is $d_s = r_4 + r_5 + r_6$. The fare of passengers before sharing is:

$$\begin{cases} c_{1.f_b} = r_1 + r_2 + r_6 \\ c_{2.f_b} = r_4 + r_5 + r_6 + r_7 \end{cases} \quad (14)$$

, and the fare of passengers after sharing is:

$$\begin{cases} c_{1.f_a} = r_1 + r_3 + \frac{\alpha}{\alpha+\beta} \times (r_4 + r_5 + r_6) \\ c_{2.f_a} = \frac{\beta}{\alpha+\beta} \times (r_4 + r_5 + r_6) + r_7 \end{cases} \quad (15)$$

Thus, we can prove that:

$$\begin{cases} c_{1.f_a} < c_{1.f_b} \\ c_{2.f_a} < c_{2.f_b} \end{cases} \quad (16)$$

## IV. Simulation and Results

### A. Overview

We performed the experiment using the dataset of T-Drive trajectory data sample. Road networks: the dataset using the road network of Beijing, which contains 106,579 road nodes and 141,380 road segments. Trajectories: the dataset contains the GPS trajectory records for over 10,357 taxis during the period of Feb. 2 to Feb. 8 in the year 2010.

Due to the high density of trajectories, we retrieved the center region of the T-Drive trajectory data sample as the service region for the experiment. Fig 5 visualizes the density distribution of the GPS points in the dataset and the service region.

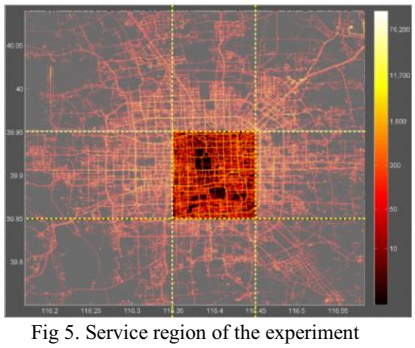

Fig 5. Service region of the experiment

### B. Performance Evaluation

In our study, a taxi needs a recommendation to find its first passenger at the beginning of the day. Then the taxis with one passenger in the car continue searching for proper

passengers for taxi-sharing. A taxi can only pick up one passenger at a time. After picking up passengers, the taxi decides on a route based on the destinations of all the passengers. Adopting the arbitrary shortest path algorithm as the driving policy is the best decision. The distance of and travel time for the route can be computed using the tables constructed based on the time-dependent R-Trees.

We compare our work with the traditional method, which finds passengers and taxis without recommendation. In the traditional method taxis cruise randomly and only pick up one passenger in a trip. We simulated the method with and without recommendation from different aspects.

Fig. 12 shows the average waiting time of taxis with and without recommendations respectively. Each point represents the on average waiting time during 24 hours. With a fixed number of taxis, the average number of taxis in our environment is 1376. The x-axis is the average number of passengers. After the number of passengers increases to 1200, the waiting time for taxis with recommendations diminishes sharply, and is almost one half of the waiting time for taxis without recommendations. Our work can therefore provide a 25% waiting time reduction on average. Fig. 13 shows the average service time of passengers in conditions with and without recommendations respectively. The service time of passengers with recommendations is longer than those without recommendations because of the additional time spent on detours for taxi-sharing.

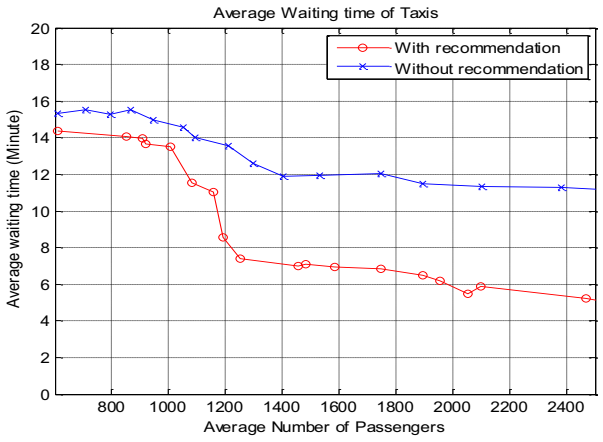


Figure 6. Average waiting time of taxis

## V. CONCLUSION

In this paper, we have proposed a taxi-sharing recommendation mechanism for both taxis and passengers, which combines a non-cooperative game model to solve the competition among taxis in need of route recommendations. Based on the historical information of taxis and passengers, we built time-dependent R-Trees to discover popular locations so that a server can suggest routes and locations using these R-Trees.

We are still in the process of refining the time-dependent R-Trees and parameters in the proposed model. After the simulation, we believe that there are a lot of issues that can be considered; such as an R-Tree with map-matching, the behavior of taxi drivers and the correlations between taxi-sharing passengers. It may be beneficial for the model to take these factors into consideration in the future.

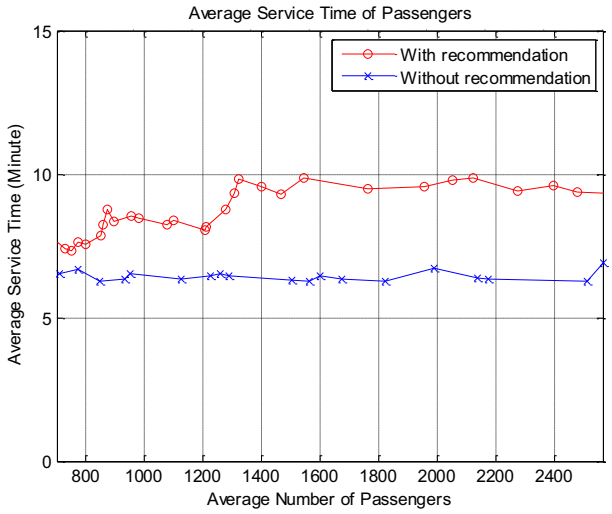


Figure 7. Average service time of passengers

## REFERENCES


[1] Shangyao Yan, Chun-Ying Chen, and Sheng-Chieh Chang. A Car Pooling Model and Solution Method With Stochastic Vehicle Travel Times. *IEEE Transactions on Intelligent Transportation Systems*, 2013; pp. 1–15.

[2] George Dimitrakopoulos, Panagiotis Demestichas, and Vera Koutra. Intelligent Management Functionality for Improving Transportation Efficiency by Means of the Car Pooling Concept. *IEEE Transactions on Intelligent Transportation Systems*, 2012; pp. 424–436.

[3] Antonin Guttman. R-trees a dynamic index structure for spatial searching. *Proceedings of the ACM SIGMOD international conference on Management of data*, 1984; pp. 47–57.

[4] Tuukka Haapasalo, Ibrahim Jaluta, Seppo Sippu, and Eljas Soisalon-Soininen. On the Recovery of R-Trees. *IEEE Transactions on Knowledge and Data Engineering*, 2013; pp. 145–157.

[5] Ben Kao, Sau Dan Lee, Foris K.F. Lee, David Wai-lok Cheung, and Wai-Shing Ho. Clustering Uncertain Data Using Voronoi Diagrams and R-Tree Index. *IEEE Transactions on Knowledge and Data Engineering*, 2010; pp. 1219–1233.

[6] Zhanshan Sam Ma and Axel W. Krings. Dynamic Hybrid Fault Modeling and Extended Evolutionary Game Theory for Reliability, Survivability and Fault Tolerance Analyses. *IEEE Transactions on Reliability*, 2011; pp. 180–196.

[7] Liqiang Zhao, Jie Zhang, and Hailin Zhang. Using Incompletely Cooperative Game Theory in Wireless Mesh Networks. *IEEE Network*, 2008; pp. 39–44.

[8] Eiichi Umehara and Toshizumi Ohta. Using Game Theory to Investigate Risk Information Disclosure by Government Agencies and Satisfying the Public—The Role of the Guardian Agent. *IEEE Transactions on Humans*, 2009; pp. 321–330.

[9] Nicholas Jing Yuan, Yu Zheng, Liuhang Zhang, and Xing Xie. T-Finder: A Recommender System for Finding Passengers and Vacant Taxis. *IEEE Transactions on Knowledge and Data Engineering*, 2013; pp. 2390–2403.

[10] Pedro M. d'Orey, Ricardo Fernandes and Michel Ferreira. Empirical Evaluation of Dynamic and Distribution Taxi-Sharing System. *IEEE International Conference on Intelligent Transportation Systems*, 2012; pp. 140–146.