A Novel Taxi Dispatch System for Smart City

Qingnan Zou, Guangtao Xue, Yuan Luo, Jiadi Yu, and Hongzi Zhu

Department of Computer Science & Engineering, Shanghai Jiao Tong University
{redsniper,gt_xue,yuanluo,jdyu,hongzi}@sjtu.edu.cn

Abstract. Taxis as a kind of public transit have been taken by citizens thousands of times every day in urban areas. However, it is economically inefficient for vacant taxis to randomly cruise around to seek for passengers. In this paper, we propose a dynamic taxi dispatch system for smart city which dispatches routes with high probability to encounter passengers for vacant taxis. In the system, a dynamic probabilistic model has been established, which considers the impact of time on passenger appearance and the effect of different vacant taxis traveling route on each other's pick-up probability. Specifically, a novel feedback system has been introduced in the system, which utilizes the information about where taxis pick up passengers to amend system probabilistic model. Moreover, extensive trace-driven simulations based on real digital map of Shanghai and historical data of over 2,000 taxis demonstrate the good performance of our system.

Keywords: taxi dispatch system, passenger probabilistic model, hot spot, feedback mechanism.

1 Introduction

As an important part of public transit, taking taxis are very popular among citizens' choices of traveling in the urban area. For example, according to an investigation in New York City [1], 41% respondents take a taxi every week and 25% of them take a taxi every day. Due to the huge demand of citizens, large amount of taxis are deployed in most metropolises such as Shanghai. Generally, in order to pick up passengers, drivers of vacant taxis often drive their vehicles along random routes or special routes based on their personal experience, which, however, due to the lack of passenger information, often ends up with not encountering any passengers. Thus, taxi drivers acquire decline of income because of the waste of time and energy in cruise. Moreover, the invalid cruise of vacant taxis also contributes to traffic jams in cities.

To help vacant taxi pick up passengers more efficiently and smart, however, is a challenging problem. First, it is difficult to gain the distribution of passengers since passengers can appear at any place and any time in the city. Second, despite knowing possible time and place where passengers may appear, designing a cruise route for a vacant taxi is still not trivial, which somehow can be reduced to a traveling salesman problem [2]. Third, since the number of passengers is limited,

N. Streitz and C. Stephanidis (Eds.): DAPI/HCII 2013, LNCS 8028, pp. 326–335, 2013.

different vacant taxies traveling on road can impact each other. For example, a vacant taxi heading for a place where passengers may appear will fail to meet passengers if another vacant taxi has arrived there earlier and pick up the only one passenger. Recently, some taxi schedule systems have been proposed to help vacant taxis pick up passengers more efficiently. Y. Ge et al. [2] introduced an algorithm for taxi dispatching called LCP, which however ignored neither the dynamics of passenger appearance nor the impact among different vacant taxis. An adaptive taxi dispatching system was proposed by K. Yamamoto [3], which simply takes passenger appearance for a fixed model.

In this paper, a novel taxi dispatch system is proposed to help vacant taxis pick up passengers more efficiently. Generally, the taxi dispatch scheme consists of two technical components. First, a dynamic model of passenger appearance is established in order to predict the appearance of passengers across the whole area. By clustering passenger appearance records from historical data into several clusters, places where passengers appear with high probability-**hot spots**-are found, around which the appearance of passengers within a short time is discovered to obey Poisson Process. In this way, the system can predict where and when passengers appear precisely. Second, an adaptive dispatch algorithm with feedback mechanism is proposed, which considers the impact among different taxis when designing routes for vacant taxis and introduces methods with pruning to search best routes for vacant taxis so that computing complexity declines. In order to enhance the accuracy of the system, a feedback mechanism is introduced, in which taxis report where they succeed to pick up passengers. The system is evaluated through extensive trace-driven simulation and results show that it can achieve performance.

- Discover that the appearance of passengers in a region within a short time obeys Poisson Process by analyzing massive historical records. This phenomenon is key to the system to establish an accurate model of passengers and predict the pick-up probability of vacant taxis precisely.
- Propose an online taxi dispatch system, which considers impact among different taxis while designing routes for vacant taxis and introduces a feedback mechanism to improve the accuracy of pick-up probability predicted by the system.
- Give a good method of pruning to reduce the computing complexity faced when designing routes for vacant taxis. In this way, our system can handle large scale vacant taxis in urban area, especially in metropolises.

The remainder of the paper is organized as follows. In Section 2, the related work in the literature is introduced. In Section 3, we present the problem definition and the system model. The design detail of our system is presented in Section 4. Section 5 describes the evaluation on the system and gives the result. We finally give a conclusion of our work in Section 6.

2 Related Work

As the GPS facilities is common in taxis, taxis dispatching system is attractive nowadays. D. Lee et al. [4] designed a taxi dispatch system based on current demands and traffic conditions, which, however, aims at reducing the waiting time of passengers but not consider much of taxi drivers' profit. S. Phithakkitnukoon et al. [5] proposed a method to predict vacant taxis on road, which, like [4], also only services passengers. The work in [2], [7], [8] and [9] all proposed taxi dispatch systems for taxi drivers so that they could pick up passengers more quickly and efficiently. But, they all missed the dynamics of passenger appearance and the impact among different taxis. J. Yuan et al. [6] introduced a taxi system which considered passenger appearance a dynamic process but ignored the impact among different taxis. An adaptive taxi dispatch system was proposed in [3], which also simplify the appearance of passengers as a fixed model.

Different from those systems, in our system, each design of routes for vacant taxis to pick up passengers is based on a dynamic model of passenger appearance and the impact of different taxis. Furthermore, a feedback mechanism is introduced in our system so that the system can be more accurate and adaptive.

3 System Model

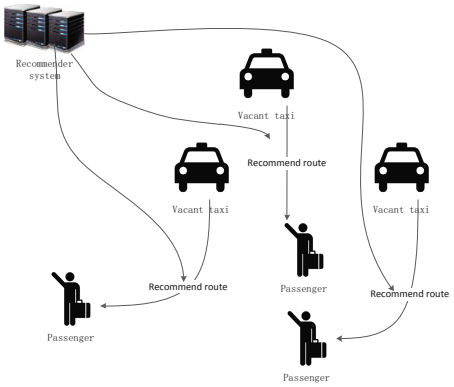


Fig. 1. System model

In this section, the model of our system and the problem solved by the system is presented. Generally, our system can be divided into two parts: a client integrated in taxis and an online dispatch server. In order to simplify the problem, it is assumed that all vacant taxis on road obey dispatching of the system. As shown in Fig. 1, when a taxi is vacant, it sends a request including its location information to the dispatch server. Then the server designs a route for the taxi

according to historical data and routes designed for other vacant taxis, and sends it to the taxi. After that, the taxi will drive along the route until it encounters passengers. If the taxi fails to encounter any passengers until it drives to the end, it will send dispatch request to the server again, which however seldom happens in our system.

3.1 Metrics and Problem Formulation

In order to describe the problem clearly, some definitions are firstly given as follows.

Definition 1. Route. *A route is a limited sequence of consecutive roads.*

Definition 2. Hot spots. *Hot spots are special roads where passengers are more likely to appear than on other roads.*

The pick-up probability can be easily conceived of as the metric, which however is not always in accordance with the taxi drivers' profit. It is because what taxi drivers want is to pick up passengers while drives as short as possible. But high pick-up probability sometimes means long traveling distance. Since pick-up probability is not suit as metric in our system, we introduce the **expected traveling distance** as the metric, which represents the expected traveling distance before a vacant taxi can come across passengers.

Generally, there are several hot spots in urban area. An idea way to dispatch a vacant taxi is to design a route traveling all the hot spots with the lowest expected traveling distance, which however is very difficult. According to [2], even if passenger appearance is assumed to be with fixed probability, the problem can be reduced to a traveling salesman problem. It is more complex if a dynamic passenger appearance model is applied in the system since dynamic model is more complex than the fixed one.

In order to reduce the computing complexity of the problem, a traveling distance limit $\boldsymbol{\sigma}$ is set for designing routes in the system. Therefore, the problem becomes how to design a route through several hot spots with a limited distance for a vacant taxi so that the taxi has the lowest expected traveling distance when traveling along the route.

3.2 Passenger Appearance Model

From historical data, it is discovered that passenger appearance in a region spot within a short time obeys Poisson Process. Fig. 2(a) shows an example of passenger appearance number per minute in Xujiahui area, Shanghai from 14:00 to 17:00 every day during the period from 2007/02/20 to 2007/02/25. It is obvious that the curve in the figure is almost the same as the curve of Poisson distribution, which indicates passenger appearance obeys Poisson Process.

$$Pr[N(t_s - t_r) = k] = \frac{[\lambda(t_s - t_r)]^k}{k!} e^{-\lambda(t_s - t_r)}. \tag{1}$$

Since passenger appearance around a hot spot within a short time obeys Poisson Process, it can be presented by Eq. (1), where t_s is the start time, t_r is the end time, k is the number of passengers appeared during the period and λ is the eigenvalue of Poisson Process. **It should be noted that in the equation, "one passenger" represents a group of passengers taking the same taxi.** For the ith vacant taxi arriving at a hot spot, let

- t_i **denote the arriving time and** t_0 **denote the start time of Poisson Process;**
- A_i **represent that the** ith **taxi succeeds to pick up a passenger;**
- $Pr(A_i)$ **denote the probability of picking up a passenger around the hot spot for the taxi.**

Though the model of passenger appearance is known, it is still difficult to predict the pick-up probability i.e. $Pr(A_i)$. In order to simplify the problem, a function of two variables is introduced:

$$F(i,k) = \sum_{j=0}^{k+1} F(i-1,j) Pr[N(t_i - t_{i-1}) = k - j + 1]. \tag{2}$$

$F(i,k)$ denotes the probability that there are k passengers left around a hot spot after the ith arriving taxi leaves, which means there are $k+1$ passengers around the hot spot before the ith taxi arrives as Eq. (2) presents. It is obvious that Eq. (2) can be calculated iteration and the boundary iteration can be calculated intuitively. With Eq. (2), $Pr(A_i)$ can be easily calculated:

$$\begin{aligned} Pr(A_i) =& 1 - Pr(\overline{A}_i) \\ =& 1 - F(i-1,0) Pr[N(t_i - t_{i-1}) = 0]. \end{aligned} \tag{3}$$

It should be noted that there is some difference on the boundary such as $i = 1$ or $k = 0$. But all these boundary situations can be derivate intuitively, and we do not present the details here due to room limit.

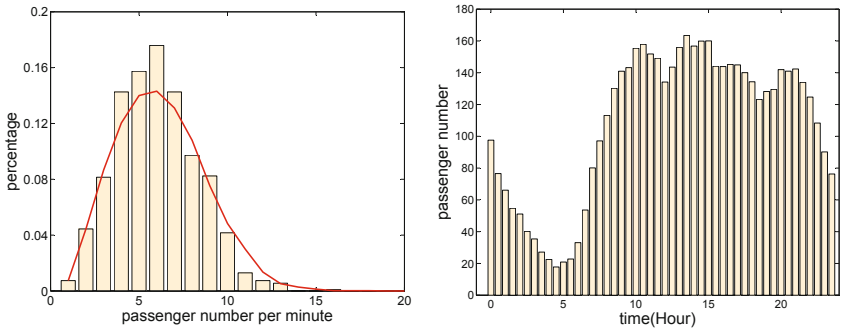
Considering the pick-up probability of the ith arriving taxi under the condition that the $(i-1)th$ taxi does not pick up any passengers i.e. $Pr(A_i|\overline{A}_{i-1})$. Since the $(i-1)th$ taxi fails to pick up any passengers, no passengers are left after the $(i-1)th$ taxi leaves i.e. $F(i-1,0) = 1$. Moreover, according to the features of Poisson Process, the event of passenger appearance from t_{i-1} to t_i is independent from that during other nonoverlapping period. Therefore, the conditional probability is:

$$Pr(A_i|\overline{A}_{i-1}) = 1 - Pr[N(t_i - t_{i-1}) = 0], \tag{4}$$

which can be used to update the passenger model in the system with feedbacks from taxis.

4 Design Details

In this section, the details of our taxi dispatch system will be exhibited. First, the contents about how the system finds hot spots and estimate eigenvalue of



(a) Passenger number per minute (b) Passenger number per 0.5h

Fig. 2. The feature of passenger appearance

Poisson Process at each hot spot according to historical records are presented; second, the dispatch algorithm i.e. how the system assign a route for each vacant taxi is shown; finally, the state buffer and feedback algorithm is demonstrated.

4.1 Analysing Historical Records

As defined in Definition 2, hot spot are special roads where passengers are more likely to appear. An intuitive thought is that there must be more passengers appearing around hot spots than in other places in historical records. Therefore, the system can divide historical passenger records into several clusters according to their clustering property in geographical location. Then, the geographical center of each hot spot can be treated as a hot spot. In our system, the K-means algorithm [11] is applied to cluster historical passenger records. Since the cluster number should be set before applying K-means algorithm, the system must choose a suitable cluster number for a region through test.

For each hot spot, the eigenvalue of Poisson Process should be ascertained in order to build passenger appearance model around it as presented in Eq. (1). However, the eigenvalue at a hot spot varies with time. Fig. 2(b) illustrates average passenger number per half hour across a day with one-month statistics retrieved from historical records in Xujiahui area, Shanghai. From the figure, it can be seen that passenger number varies much across a day due to most people's daily routine. However, the number does not change much during a short period like an hour, which gives the system a way to ascertain the eigenvalue during a short period without much deviation. In the system, a sliding window of which the length is set to be one hour is set to estimate the eigenvalue of a hot spot. For example, when the system needs to know the eigenvalue at time t, it counts passenger number from historical records in the time window $[t-30min, t+30min]$. Then the system uses the Maximum-likelihood Estimation (MLE) to estimate the eigenvalue during that period. Sliding window is better than dividing a day into several fixed periods because the second method is not

accurate on the boundary of each period. Therefore, using sliding window to estimate eigenvalue makes the system more adaptive to time change.

4.2 Dispatch Algorithm

As shown in previous sections, a good route means the expected traveling distance for a vacant taxi to pick up passengers is short. Since the process of predicting pick-up probability shown in section 3.2 has already taken the impact of other taxis into account, what the system should do is to design a route through several hot spots with the shortest expected traveling distance.

Assume that the hot spots traveled by a route are:

$$\{S_1, S_2, ..., S_n\},$$

and the corresponding pick-up probabilities at each hot spot are:

$$\{P_1, P_2, ..., P_n\}.$$

Since the route is divided into n parts by the n hot spots, the corresponding distance of each part is:

$$\{D_1, D_2, ..., D_n\}.$$

In order to simplify the problem, the system uses the minimum distance got by Dijkstra algorithm [12] to represent the distance between two hot spots or the origin and a hot spot. Thus the expected traveling distance of the route is:

$$P_1 D_1 + (\sum_{i=1}^{n} D_i) \times \prod_{j=1}^{n} (1 - P_j) + \sum_{i=1}^{n-1} [P_{i+1} \times (\sum_{j=1}^{i+1} D_i) \times \prod_{k=1}^{i} (1 - P_i)]. \quad (5)$$

Thus, the aim of designing route is to make Eq. (5) minimum.

As presented before, in order to reduce the computing complexity of designing a best route, a distance limit $\boldsymbol{\sigma}$ has been set. In the system, the branch and bound method is applied to searching solution space under the distance limit $\boldsymbol{\sigma}$. With this method, the system can find the solution i.e. the best route quickly. The general steps of dispatch algorithm is presented in Algorithm. 1.

4.3 State Buffer and Feedback Mechanism

As presented before, the system needs the dispatch information, such as time and dispatch route, of previous vacant taxis to dispatch current vacant taxis. In the system, a linked table, which plays a role as state buffer, is established for each hot spot as shown in 1.

However, linked tables are enlarging with time in the system because the number of vacant taxis dispatched by the system is increasing, which brings large overheads in storing and computing. In order to solve this problem, a feedback

Algorithm 1. Dispatch algorithm

Input:
Current GPS location of vacant taxi; Current time;

Output:
A scheduled route;

1: Match GPS location of vacant taxi with corresponding route in the digital map and calculate the distance (of the shortest route) to each hot spot;
2: Build passenger model at each hot spot according to time and use branch and bound method to search for the best route for the vacant taxi under the distance limit σ.
3: Once the best route is got by the last step, the predicted time of the taxi arriving at each hot spot contained int the route is stored into the linked tables of corresponding hot spots.
4: Send out the best route to the vacant taxi.

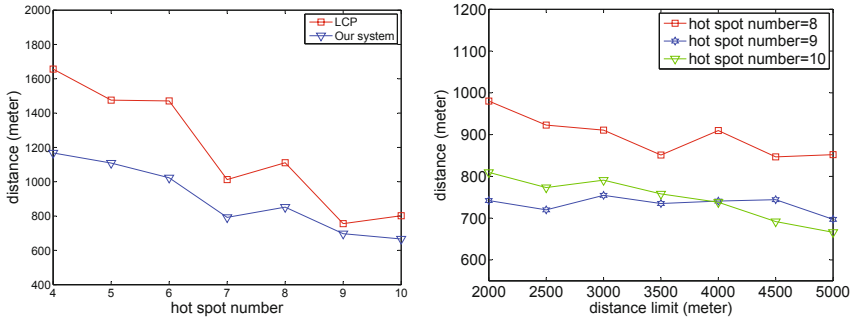
mechanism is introduced in the system. As Eq. (4) denotes, the conditional pick-up probability around a hot spot is independent from previous taxis if the last taxi does not pick up any passengers around the hot spot. Thus the system can remove information in a linked table of a hot spot before the time when a taxi reports it does not pick up passengers around the hot spot. Moreover, the feedback mechanism raises the precision of pick-up probability because it introduces the information in reality to amending the information retrieved from historical records.

5 Evaluation

In this section, we will show how we evaluate the system through trace-driven simulations and how the system performs under the evaluation. As presented before, the evaluation on the system is realized by trace-driven simulations. The simulation program is written by C# language under Visual Studio 2010 environment. The simulation is executed in a computer with I5 2400 CPU, 8 GB memory and 64bit Windows 7 Professional operating system. The training historical data and trace data are all collected around Xujiahui, Shanghai from ShanghaiGrid project [10]. The area where data is collected starts from 121.418409 to 121.463299 in longitude and from 31.181709 to 31.203882 in latitude with a length of 4.3 km and a width of 2.4 km. The data used as historical records is collected from 2007/1/31 to 2007/2/28 and the data used as trace is collected from 2007/3/1 to 2007/3/6.

5.1 The Impact of Hot Spot Number

In this simulation, we set the distance limit 5000 meters and change the number of hot spots from 4 to 10 to see the performance of the system i.e. the average distance traveled by vacant taxis before picking up passengers. The LCP system [2] is also realized in the simulation to be compared to our system.



(a) The impact of hot spot number (b) The impact of distance limit σ

Fig. 3. The result of evaluation

As Fig. 3(a) illustrates, the performance of both systems raises with the increase of hot spots number and our systems always precedes LCP. Though the gap is low when hot spot number is high, our system can reach the same performance with less hot spots, which means low computing overhead.

Since more hot spots means more precise passenger model around each hot spot, the performance of both systems raises with the increase of hot spot number. However, the performances improve when hot spot number is high is not as fast as that when hot spot number is low. This implies that for an area, high hot spot number does not always mean good as high hot spot not only brings performance improve but also high computing overhead.

5.2 The Impact of Distance Limit σ

In this simulation, we change the distance limit from 2000 meters to 5000 meters with 8, 9 and 10 hot spots to see its impact on system performance.

As Fig. 3(b) presents, the system performance improves a little with the increase of distance limit. However, the system improve caused by increase of distance limit is not as obvious as that caused by increase of hot spot number. Moreover, the average traveling distance is much shorter than the distance limit. This phenomenon implies that most vacant taxis pick up passengers at the first several hot spots. Therefore, for the evaluated area, the system can use a relatively short distance limit, which brings both good performance and low computing overhead.

6 Conclusion

This paper proposes a novel dynamic taxi dispatch system to dispatch route for vacant taxis to pick up passengers. In the system, a dynamic passenger appearance model is established according to historical records and an adaptive dispatch algorithm is introduced considering the impact of previous dispatched

vacant taxis. Moreover, the dispatch algorithm is specially designed to reduce computing overhead. Furthermore, a feedback mechanism is introduced in the system to reducing computing and storing overhead and improving dispatch accuracy. Besides, a trace-driven simulation has been conducted to evaluate our system, of which the result shows our system has a good performance.

Acknowledgements. This research is supported by the National High Technology Research and Development Program (2011AA010502) and the National Natural Science Foundation of China (Grant Nos. 61170237).

References

1. Taxi of Tomorrow Survey Results. New York City Taxi and Limousine Commission (2011)
2. Ge, Y., Xiong, H., Tuzhilin, A., Xiao, K.: An energy-efficent mobile recommender system. In: Proc. KDD, pp. 899–908 (2010)
3. Yamamoto, K., Uesugi, K., Watanabe, T.: Adaptive routing of cruising taxis by mutual exchange of pathways. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) KES 2008, Part II. LNCS (LNAI), vol. 5178, pp. 559–566. Springer, Heidelberg (2008)
4. Lee, D., Wang, H., Cheu, R., Teo, S.: Taxi dispatch system based on current demands and real-time traffic conditions. Transportation Research Record: Journal of the Transportation Research Board 1882(-1), 193–200 (2004)
5. Phithakkitnukoon, S., Veloso, M., Bento, C., Biderman, A., Ratti, C.: Taxi-aware map: Identifying and predicting vacant taxis in the city. In: de Ruyter, B., Wichert, R., Keyson, D.V., Markopoulos, P., Streitz, N., Divitini, M., Georgantas, N., Mana Gomez, A. (eds.) AmI 2010. LNCS, vol. 6439, pp. 86–95. Springer, Heidelberg (2010)
6. Yuan, J., Zheng, Y., Zhang, L., Xie, X., Sun, G.: Where to find my next passenger. In: Proc. Ubicomp 2011, pp. 109–118 (2011)
7. Li, B., Zhang, D., Sun, L., Chen, C., Li, S., Qi, G., Yang, Q.: Hunting or waiting? Discovering passenger-finding strategies from a large-scale real-world taxi dataset. In: 2011 IEEE International Conference on Pervasive Computing and Communications Workshop (PERCOM Workshops), pp. 63–68 (2011)
8. Powell, J., Huang, Y., Bastani, F., Ji, M.: Towards reducing taxicab cruising time using spatio-temporal profitability maps. In: Proceedings of the 12th International Symposium on Adavances in Spatial and Temporal Databases, SSTD 2011 (2011)
9. Lee, J., Shin, I., Park, G.: Analysis of the passenger pick-up pattern for taxi location recommendation. In: International Conference on Networked Computing and Adavanced Information Management (2008)
10. Li, M., Wu, M.-Y., Li, Y., Cao, J., Huang, J., Deng, Q., Lin, X., Jiang, C., Tong, W., Gui, Y., Zhou, A., Wu, X., Jiang, S.: ShanghaiGrid: an Information Service Grid. Concurrency and Computation: Practice and Experience 18, 111–135 (2006)
11. Wagstaff, K., Cardie, C., Rogers, S.: SConstrained K-means clustering with background knowledge. In: International Conference on Machine Learning, ICML (2001)
12. Dijkstra, E.W.: A note on two problems in connection with graphs. Numerical Mathematics 1 (1959)