Hindawi
Journal of Advanced Transportation
Volume 2020, Article ID 8674512, 12 pages
https://doi.org/10.1155/2020/8674512

Research Article

Multiagent Reinforcement Learning-Based Taxi Predispatching Model to Balance Taxi Supply and Demand

Yongjian Yang, Xintao Wang, Yuanbo Xu, and Qiuyang Huang

School of Computer Science and Technology, Jilin University, Changchun 130012, China

Correspondence should be addressed to Yuanbo Xu; yuanbox@jlu.edu.cn

Received 10 November 2019; Revised 16 January 2020; Accepted 24 January 2020; Published 19 February 2020

Academic Editor: Francesco Galante

With the improvement of people's living standards, people's demand of traveling by taxi is increasing, but the taxi service system is not perfect yet; taxi drivers usually rely on their operational experience or cruise randomly to find passengers. Without macroguidance, the role of the taxi system cannot be fully utilized. Many scholars have studied taxi behaviors to find better operational strategies for drivers, but their researches rely on local optimization methods to improve the profit of drivers, which will lead to imbalance between supply and demand in the city. To solve this problem, we propose a Multiagent Reinforcement Learning- (MARL-) based taxi predispatching model through analyzing the running data of 13,000 taxis. Different from other methods of scheduling taxis based on the real-time location of orders, our model first predicts the demand for taxis in different regions in the next period and then dispatches taxis in advance to meet the future requirement; thus, the number of taxis needed and available in different regions can be balanced. Besides, in order to reduce computational complexity, we propose several methods to reduce the state space and action space of reinforcement learning. Finally, we compare our method with another taxi dispatching method, and the results show that the proposed method has a significant improvement in vehicle utilization rate and passenger demand satisfaction rate.

1. Introduction

Smart city, an emerging technology, which aims to apply the new generation of information and communication technology to all walks of life in the city, is able to alleviate the "big city disease" [1], coordinate urban development, and improve the running efficiency of the city and the quality of citizens' life [2]. Intelligent transportation [3, 4], as an indispensable part of a smart city, aims at improving the operation efficiency of transportation systems, making full use of transportation resources, and ensuring traffic safety [5]. It plays a vital role in citizens' lives and the operation of the whole city. Nowadays, traffic congestion, frequent accidents, energy waste, air pollution, and other problems commonly exist in cities and they can be well solved by intelligent transportation [6, 7].

With the rapid development of wireless communication technology and the Internet of Things (IoT), collecting the trajectory records of mobile objects becomes simple and fast, which makes intelligent transportation possible [5, 8]. Various devices embedded with GPS are ubiquitous in our lives, such as smartphones [9, 10], private cars [11, 12], and public transport [13]. Location information can be obtained more easily, and a large number of trajectory data are collected every day. Trajectory data has spatial attributes as well as temporal attributes; it becomes the main research object of spatiotemporal data mining technology. The application of trajectory data can not only provide location-based services for users, but also help urban planning and intelligent transportation. Gathering and analyzing these large-scale real-world digital traces have provided us with an unprecedented opportunity to grasp the city dynamics and understand the social and economic patterns better [14–16].

However, the corresponding operation strategy did not develop with the increase of the number of taxis, there are still many shortcomings, such as the difficulty in finding taxis in peak hours, uneven distribution of taxis, and the drivers' refusal of service [17]. Taxi drivers' strategies of seeking passengers are mostly empirical and substantially vary among each other [18, 19], which leads to low service efficiency and low income. Many studies have been devoted to solving these problems [8, 18, 20, 21], but basically from the drivers' point of view, these local optimization methods may lead to starvation in some areas. So it can neither provide guidance for taxi dispatching from a global perspective nor provide better ride experience for passengers. There are also studies devoted to assigning vehicles to each order based on the real-time order locations. However, scheduling based on real-time order status has some drawbacks; for example, if there are few taxis available around a passenger, we have to arrange a taxi according to the shortest distance priority principle to serve this passenger, but the actual distance might be very far. It is not an ideal arrangement neither for the driver nor for the passenger. Vehicles have to travel longer distances, and passengers need to wait longer which makes the whole taxi system inefficient.

To this end, we propose a vehicle prescheduling model from the perspective of the whole city, so that taxi resources can be fully utilized and service quality and passengers' experience can be improved. Through analysis of the historical trajectory data, firstly we identify the characteristics of the population movement patterns and taxi operation rules in cities. Based on these two points, then we count the number of vehicles that can provide services at the current time and predict the amount of taxi demands in the future. According to the predicted results, we can know the quantity of supply and demand in every area of the city. Finally, Multiagent Reinforcement Learning can be used for taxi scheduling, which will eventually balance the global supply and demand and enable more passengers to take taxis in shorter time.

Our major contributions are summarized as follows:

(1) We study the crowd movement patterns in different regions through analyzing the historical taxi trajectory data, which can provide some auxiliary information for vehicle scheduling.

(2) We propose a taxi predispatching model based on Multiagent Reinforcement Learning method, which can balance the number of taxis and requirements in each region.

(3) We propose a divide-and-conquer method to reduce the volume of overlarge state space in MARL, which improves the computational efficiency.

(4) We evaluate the performance of different time series prediction algorithms in predicting future pickup requests through experiments and prove the validity of the proposed model through experimental comparison.

The remainder of this paper is organized as follows. In Section 2, we give a brief review of taxi operation strategy researches and online order matching methods. In Section 3, we provide the definition of the problem; then we introduce the processing pipeline of the article. The data used in this paper, the method of processing the data, and the division of urban areas are introduced in Section 4. In Section 5, we introduce the scheduling method based on Multiagent Reinforcement Learning. The experimental results are shown in Section 6. Finally, we conclude the paper in Section 7.

2. Related Work

Mining taxi trajectory data has been a research hotspot in the smart city [22]; many scholars have studied this issue. Through the analysis of relevant studies, we find that the literature on taxi research mainly focuses on two aspects. One is to analyze the taxi drivers operating strategies and study which strategy can bring higher income to drivers. The other is from the perspective of the overall taxi market, focusing on dispatching and providing guidance for taxis. In this section, we mainly introduce the research results of other scholars from these two perspectives.

Different cities have different characteristics of crowd movement patterns. But in the same city, the income of different drivers is also different because they may adopt different operation strategies. Many scholars have studied which kind of operation strategy taxi drivers should adopt to get higher profits. Rong et al. [18] extract efficient operational strategies through large-scale historical taxi trajectory data and then analyze these strategies through multiple indicators to get some valuable insights and use these strategies to increase drivers' income. Li et al. [14] design a simulation model to test the performance of three different search strategies from two perspectives including passenger waiting time and vacant taxi travel rate. Chen et al. [23] use three indicators including the levels of taxi service, taxi operation, and taxi development to analyze the operation of taxis, so as to improve the management of the taxi industry and promote the sustainable development of the taxi industry.

Some scholars offer advice to taxi drivers by analyzing crowd movement patterns. Based on these patterns, they provide suggestions for taxi drivers and recommend some locations for them. In these locations, there is a greater possibility of receiving passengers, which can reduce the cruising time and thus increase their income. Kong et al. [24] propose a time-location-relationship (TLR) combined service recommendation model to improve drivers' profits according to the characteristics of passengers in different functional regions. The TLR model analyzes the relationship between passengers getting on and off during every period and adopts Gaussian Process Regression (GPR) to predict the amount of passengers and recommends drivers to their nearest region where the demand of taxi is most at the same time. Phithakkitnukoon et al. [25] present a predictive model for the number of vacant taxis in a given area based on time of the day, day of the week, and weather condition. With this knowledge, we can allocate vehicles for requests more quickly. Xiaolong et al. [26] investigate human mobility patterns by analyzing large-scale taxi traces and develop an improved ARIMA method to predict Pickup Quantity (PUQ) of those urban hotspots and then recommend taxi drivers to an optimal hotspot where the taxi driver will spend the least time to pick up the next passenger.

Yuan et al. [27] present a recommender system for both taxi drivers and people expecting to take a taxi, using the knowledge of passengers mobility patterns and taxi drivers picking-up/dropping-off behaviors learned from the GPS trajectories. This recommender system provides taxi drivers with some locations and the routes to these locations and provides people with some locations (within a walking distance) where they can easily find vacant taxis. Golpayegani and Clarke [28] consider the respective preferences of drivers and passengers. They present a multiagent collaborative passenger matching and taxi dispatch model. Passengers and drivers are modeled as autonomous agents having multiple often-conflicting preferences. The attention to the preferences of passengers and drivers in this paper gives us great inspiration. A system should consider the preferences of different users rather than treating them equally. Dimitriou et al. [16] study the taxi trajectory data of New York City. By analyzing the travel time and distance of taxi and the situation of getting on and off in key areas such as airport, they recommend the optimal location for taxis to find passengers.

The above studies are all from the drivers' point of view; the goal is to make more profits for drivers. These studies are local optimization, which are not conducive to the quality of taxi service from the perspective of the whole city. Some other studies focus on how to match available vehicles with requests more reasonably. They use different algorithms to achieve this goal; for instance, Kuemmel et al. [29] leverage a stable marriage assignment algorithm and apply it for dispatching taxis to passengers. The stable marriage algorithm was developed initially for matching men and women according to their preferences in polynomial time. Zheng and Jie [30] also use the stable marriage method. They study the online to offline taxi scheduling problem. In the case of nonsharing taxi dispatches, it uses the stable marriage method and uses three rules to find all possible stable matches. Seow et al. [31] propose a multiagent architecture to match taxis and requests attempting to improve passengers satisfaction more globally. The city is divided into different regions; each region maintains its own available taxi queue and request queue. The system will match the requests and vehicles in each region at regular intervals. Wei et al. [17] studied the impact of service refusal on the balance of supply and demand in the taxi market.

There are also some researchers who use reinforcement learning to achieve their goals. Guériau and Dusparic [32] propose a reinforcement learning-based decentralized approach to vehicle relocation as well as ride request assignment in shared mobility-on-demand systems. Each vehicle autonomously learns its behaviour, including both rebalancing and selecting which requests to serve, based on its local current and observed historical demand. The rebalancing strategies proposed in this paper are very constructive and provide us with a good reference. Li et al. [33, 34] both use MARL to solve the problem of matching vehicles and orders, but the former follows the distributed nature of the peer-to-peer ride-sharing problem and adopt the mean field approximation to simplify the local interactions by taking an average action among neighborhoods. The latter uses an extended version of reinforcement learning: hierarchical reinforcement learning (HRL). It models ride-hailing as a large-scale parallel ranking problem, combines order dispatching with fleet management, and conducts the decision-making process in a hierarchical way.

The existing researches dispatch vehicles in real time according to the location of orders. Due to the imbalance of supply and demand in different regions, some taxis need to travel a long distance to serve passengers, which will prolong the waiting time of passengers and reduce the operational efficiency. If we can know in advance the prospective demand of each region, we can take some measures to deal with this problem. Fortunately, we now have a variety of very mature predictive models, including machine learning models, deep learning models, and various time series models, all of which can achieve high accuracy. Therefore, the prescheduling model proposed in this paper first predicts the future pickup requests by time series predicting model and then dispatches taxis to achieve the balance between supply and demand in each region. After doing so, only a small-scale scheduling is required. The simulation results show that the proposed method can effectively avoid taxi congregation caused by local optimization methods and improve the operating efficiency of taxis.

3. Overview

In this section, we will introduce the problem definition and processing pipeline to have a better understanding of what is stated in this article.

3.1. Problem Definition. Regardless of the size of the city and the number of taxis, the number of available taxis and taxi demands in different areas of a city is unbalanced, especially in rush hours. Therefore, we propose a taxi predispatching model to balance the supply and demand of taxis in different regions and finally improve the utilization rate of taxis, meet more demands, and reduce passenger waiting time.

This paper regards the study area on the map as a two-dimensional plane and then divides it into equal-sized grids. According to the real-time GPS data uploaded by taxis, we can get the location of each taxi and the number of taxis in each grid (supply quantity), which compose the supply matrix S_t (t represents the time). And after forecasting the demand of each grid the demand matrix D_t can be obtained by combining the values of all grids according to their spatial locations. By subtracting the two matrices, we can get the objective matrix, through which we can know the supply and demand situation of the entire area. The problem then turns to how to schedule taxis so that more values in the target matrix are greater than or equal to zero. In this paper, Multiagent Reinforcement Learning is used to let the machine automatically explore the best adjustment scheme to achieve this goal.

3.2. Processing Pipeline. The main processing pipeline of our method is illustrated in Figure 1. It mainly consisted of four parts: data preprocessing, map partitioning, demand forecasting, and taxi dispatching. Data preprocessing is used to remove unnecessary and error information in the GPS data and facilitate later application. Map partitioning divides the city into grids of the same size and then analyses the crowd travel patterns in different grids to provide assistance for taxi scheduling later. The demand forecasting section uses several time series forecasting methods to predict the prospective number of taxi demands in each grid, so that the future demand situation of each region can be grasped in advance. After that, taxi dispatching can be done according to the current taxi distribution and future demand situation.

4. Data Process

Shanghai is one of the most prosperous cities in China. The demand for taxis is very large. Taxi plays an essential role in the urban traffic. It is of great significance to optimize the efficiency of taxi service. This paper uses the GPS positioning data of 13700 taxis in Shanghai from April 1, 2015, to April 30, 2015, to study the taxi demand in Shanghai. Taxis' positions are sampled every 10 seconds, and a piece of data is generated whenever passengers get on or off. In 30 days, about 3 billion pieces of data are generated. The fields in the data and their meanings are shown in Table 1.

4.1. Data Preprocess. Due to the device failure, transmission interference, or storage errors, data may be incorrect. For example, when a taxi driver is after work, he may keep the taximeter on although there is no passenger in the taxi. Taxi state and taxi location are very important for subsequent experiments, so unreasonable data should be corrected or deleted for the purpose of getting more accurate results. To clarify the real vacant and occupied trajectories (trajectories with and without passenger, respectively), the data processing steps are performed as follows.

Step 1. Sort data by time.

Sorting the data of each taxi according to time, the state of taxi should be regularly converted between available and occupied ones. Corresponding to the data, taxi status field should change between 0 and 1. For example, 0011... 1100 or 1100... 0011, from 1 to 0, means receiving passenger and from 0 to 1 means passenger getting off. Combining latitude and longitude, we can know where passengers get on and off.

Step 2. Eliminate errors in state transition.

The state of a vehicle might transform frequently, for example, 00100110001 or 111011011101. Obviously, these situations are unreasonable. It will cause erroneous records of getting on and off many times, which will have an impact on the results. The way to deal with such errors is to limit the shortest time with passengers on board and empty cars. If it is below the time threshold, it will be considered as a wrong conversion. Through statistical analysis of the data, the minimum time of taxi with passengers on board and no load are set to five minutes and one minute, respectively.

Step 3. Correct the wrong location point.

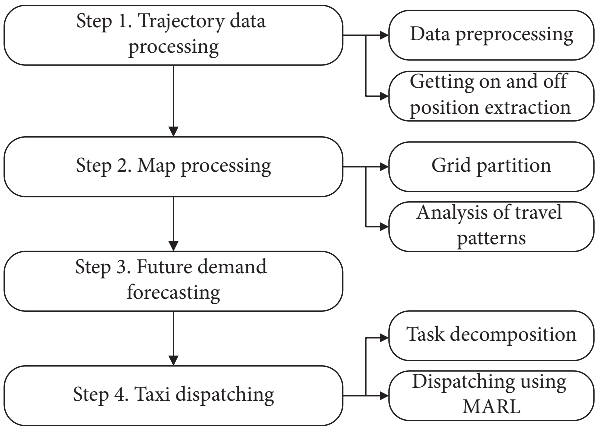

Step 1. Trajectory data processing
Data preprocessing
Getting on and off position extraction
Step 2. Map processing
Grid partition
Analysis of travel patterns
Step 3. Future demand forecasting
Task decomposition
Step 4. Taxi dispatching
Dispatching using MARL

Figure 1: Processing pipeline.

Due to the errors of GPS equipment, weak satellite signals, or transmission errors, the position of some points in the trajectory may be abnormal; that is, the distance between two points exceeds the maximum distance a car can travel over a period of time. In order to deal with this situation, we take the midpoint of the position of the two records (before and after the error record) as the actual location of the point. Since the object of analysis is grid, it is not necessary to get a very precise location.

4.2. Map Description and Process. We mainly study the area between longitude 121.4100°–121.5045° and latitude 30.1940°–31.2750° in Shanghai. This area includes commercial centers, railway stations, residential areas, and many tourist attractions. It is highly representative for analyzing the taxi situation of the whole city. Generally, there are two methods to divide a region. The first one is to divide the region according to the main roads, and the other is to divide the region into the same size grids [35]. The method of dividing by main roads is not easy in choosing the right roads because of various ring roads and viaducts and the nonuniformity of grids' size; therefore it will bring extra difficulty to the future prediction and scheduling. So we choose the second method. The research area is divided into 9 × 9 grids and tabbed from 1 to 81; the size of each grid is 1 km × 1 km. Figure 2 shows the results of partitioning.

4.3. Relationship of Getting On and Getting Off. The latitude and longitude range of each grid can be determined after the meshing is completed. The data uploaded by taxis contains latitude and longitude. So we can match each piece of data to the corresponding grid. Then, according to the time information in the uploaded record, we can get the number of available taxis and taxi demands in each grid.

After sorting the data according to time, the state of each taxi should change regularly between occupancy and idleness in the continuous time series. For example, $1 \longrightarrow 1 \longrightarrow 0 \longrightarrow \cdots \longrightarrow 0 \longrightarrow 1 \longrightarrow 1$ or $0 \longrightarrow 0 \longrightarrow 1 \longrightarrow \cdots \longrightarrow 1 \longrightarrow 0 \longrightarrow 0$. A transition of $1 \longrightarrow 0$ means that the state of taxi has changed from empty state to

Table 1: Taxi data description.

Fields	Description
Car_id	Unique code to identify each vehicle
State	The state of a taxi: 1 means that the taxi is vacant and 0 means that it is occupied
Time	The sample timestamp "YYYY-MM-DD HH:MM:SS"
Longitude	Longitude of the sampling position
Latitude	Latitude of the sampling position
Speed	Instantaneous speed at the sampling time

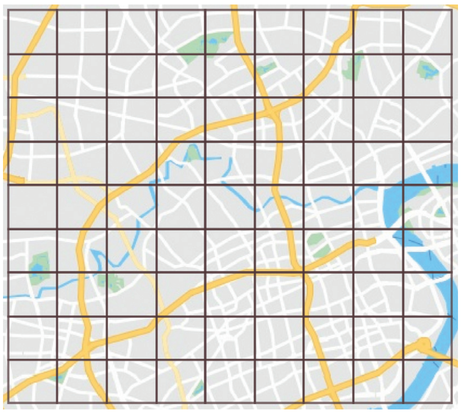


Figure 2: Partition result.

occupied state; that is, a demand is satisfied. We can count the number of transitions over a period of time to get the demand in each grid. Similarly, if the state symbol changes from 0 to 1, it means that a passenger gets off. After the above processing, we can get the quantity of getting on and getting off in each grid during all time periods.

As shown in Figure 3, we show the quantitative relationship between passengers getting on and off in three grids during weekdays and weekends. People in residential areas go out to work in the morning and go home in the evening, so the number of people getting on a taxi in the morning is more than the number of those getting off a taxi and the situation at night is just the opposite. As shown in Figures 3(a) and 3(d), the morning rush hour of working day is 8 o'clock, the evening rush hour is 20 o'clock, and the weekend morning and evening peaks are at 10 a.m. and 22 p.m., respectively. Compared with workdays, the morning and evening rush hours of weekends are later, because people go out later on weekends, and taking part in various entertainment activities at night also makes people go home later.

Commercial areas, for recreation and entertainment, maintain a relatively high number of boarding and disembarking times in comparison to residential districts. As shown in Figure 3(b), a lot of people arrived before noon and the amount of people getting on is much higher than the amount of those getting off after 21 o'clock, because people start going home. Weekends show the same trend as workdays, but the peak traffic is much busier. This is in line with our expectations; there will definitely be more people to entertain when they do not need to go to work.

Compared with residential areas, working areas have the opposite pattern of travel. People arrive at work in the morning and go home in the evening. The get-off peak is at 8-9 o'clock and the boarding rush hour is at 20 o'clock. But the traffic during evening rush hour is weaker than the early rush hour, because there is no hurry to go home from work. Some people may use different modes of transportation to go home, such as subway or bus. Comparing weekends with workdays, the patterns are the same, but the traffic and the specific time of the early peak are much weaker and later, indicating that some people still go to work on weekends, but the number of people is less, and the time is later.

Through the analysis of different functional areas, we could understand the pattern of crowd travel in different functional areas. This information can assist the scheduling process and make it more reasonable, such as dispatching more taxis to working areas during evening rush hour.

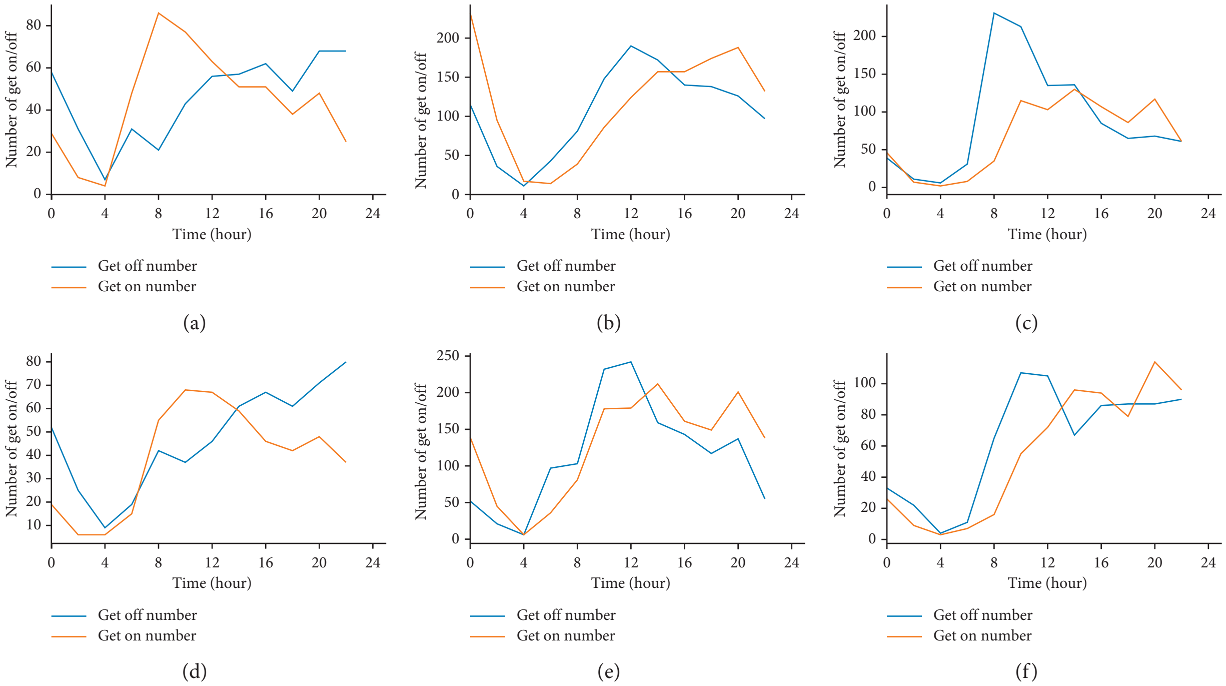
5. Dispatch Model

Through the study of historical data above, we know the supply and demand situation of taxis in different regions and can use different forecasting methods to predict the quantity of taxi demand in the future. With this knowledge, we utilize reinforcement learning method to schedule taxis, so that all regions can achieve balance between supply and demand.

5.1. WoLF-PHC Algorithm. There are some commonly used MARL algorithms, such as Minimax Q-learning, Nash Q-learning, Friend-or-Foe Q-learning (FFQ), and WoLF Policy Hill-Climbing (WoLF-PHC). The first three methods need to maintain Q-function for all agents in the learning process; the space required by the three methods is very large. In order to solve this problem, we expect each agent to maintain the Q-value function only by knowing its own actions. WoLF-PHC is such an algorithm that each agent only saves its own actions to complete the learning task. So we use WoLF-PHC in this paper.

WoLF-PHC combines "Win or Learn Fast" rule with policy hill-climbing algorithm (PHC). WoLF refers to adjusting parameters carefully and slowly when the agent does better than the expected value and speeding up the pace of adjusting parameters when the agent does worse than the expected value [36]. PHC is a single agent learning algorithm in the stable environment. The core of this algorithm is the idea of reinforcement learning, which increases the probability of choosing the action that can get the maximum cumulative expectation [37].

This algorithm defines two strategies: current strategy $h(s,a)$ and average strategy $\widetilde{h}(s,a')$. The current strategy is a probability distribution function with an initial value of



(a)

(b)

(c)

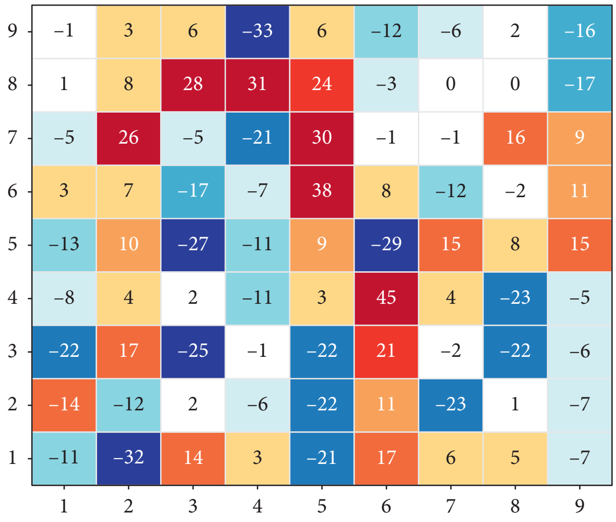
(d)

(e)

(f)

Figure 3: Relationship between getting on and getting off on weekdays and weekends in different regions. (a) Residential area weekday. (b) Commercial area weekday. (c) Working area weekday. (d) Residential area weekend. (e) Commercial area weekend. (f) Working area weekend.

$h(s,a) = (1/|A_i|)$. This probability distribution function will be updated when agent chooses action in the following way. For Q-function, if it is the best action, i.e., $a = \arg\max_a Q(s,a')$, it will increase the probability, while other actions will reduce the probability. WoLF-PHC constantly updates the average strategy and compares it with the current strategy: if the average reward value of the current strategy is greater than that of the average strategy, i.e., $\sum_a h(s,a)Q(s,a) > \sum_a \widetilde{h}(s,a)Q(s,a)$, the agent will be considered as "win." At this time, the average strategy will adopt the rate δ_{win} to update the strategy slowly. Otherwise, the current agent will be considered as "lose," and the larger rate δ_{lose} will be used for faster adaptive learning.



9	−1	3	6	−33	6	−12	−6	2	−16
8	1	8	28	31	24	−3	0	0	−17
7	−5	26	−5	−21	30	−1	−1	16	9
6	3	7	−17	−7	38	8	−12	−2	11
5	−13	10	−27	−11	9	−29	15	8	15
4	−8	4	2	−11	3	45	4	−23	−5
3	−22	17	−25	−1	−22	21	−2	−22	−6
2	−14	−12	2	−6	−22	11	−23	1	−7
1	−11	−32	14	3	−21	17	6	5	−7
	1	2	3	4	5	6	7	8	9

Figure 4: The result of $D - G$.

5.2. Dispatch Process. After forecasting the demand for each grid in the next period, the demand matrix D can be obtained by combining the predicted results of each small grid according to its spatial position. D_{ij} represents the demand of the grid in row i and column j. The supply matrix S can be obtained by counting the number of taxis in each grid at the current time. A new matrix X (as shown in Figure 4) can be obtained by subtracting the demand matrix from the supply matrix, in which the positive value represents the number of available taxis and the negative value represents the unsatisfied demands. Our goal is to minimize the negative number in the matrix with the shortest driving distance.

In order to achieve this goal, we use WoLF-PHC algorithm, which regards each taxi as an agent and uses grid number to represent its spatial position. The spatial position of each taxi constitutes the current state. After a taxi takes action, its position will change, and the state will change accordingly. Each taxi can take five actions at each step, including up, down, left, right, and stay, but it can stay only when a grid needs taxis. If a grid does not need it, it is meaningless to keep it in this grid. When the number of available taxis is larger than the total demand, we should try to satisfy all the demands. In this situation, termination state of the algorithm means that all values in the target matrix are positive; that is, the termination state is reached when all requests are satisfied. Otherwise, the termination state means that there are only negative numbers and zero in the matrix, which means that no extra taxis can be used. If the

```
Require: current vehicle distribution matrix $S$ and predicted demand matrix $D$ for each grid in the next $t$ minutes
Ensure: the dictionary of vehicle exchange between grids
ori_mat = $S - D$: supply matrix subtracts demand matrix to obtain initial difference between supply and demand in each grid
get big_grid_mat by dividing the region into large grids and calculate the difference between supply and demand in each large grid
big_grid_map ⟵ MATRIX_PROCESS (big_grid_mat)
adjust the value of the small grids in each large grid according to the scheduling result big_grid_map and get the new matrix grids_in_each_large_grid
small_grid_map = []
for each grid $grid_i$ in grids_in_each_large_grid do
    map$_i$ ⟵ Matrix_process grid$_i$
    append the map$_i$ to the small_grid_map
end for
return big_grid_map, small_grid_map
```

Algorithm 1: WoLF-PHC-based taxi dispatch algorithm.

```
Require: the matrix to be processed by the dispatching algorithm
Ensure: scheduling map obtained by algorithms
function Matrix_process (mat)
    if mat can be handled by the computing resources at hand then
        processing the mat with the WOLF-PHC-based dispatch algorithm
        return scheduling map obtained by the algorithm
    else
        Divide the matrix mat into two smaller ones mat$_1$ and mat$_2$
        map$_1$ = Matrix_process mat$_1$
        map$_2$ = Matrix_process mat$_2$
        get the result map by merging the map$_1$ and map$_2$
    end if
    return map
end function
```

Algorithm 2: Matrix process.

algorithm reaches the balance state after all agents have taken action, all agents will get a reward of 100 points; otherwise they will get −1 points. All agents take actions according to their Q table until they reach the termination state. For the same matrix, there may be many scheduling methods to achieve balance, but after the algorithm has updated the strategy it will eventually find an optimal way to achieve balance.

The location of all agents represents the state of the environment at a given time. There are 81 grids, so the size of the state space is $|G|^{|C|}$, $|G|$ is the number of grids, and $|C|$ is the number of agents. Each agent can take five actions, so the action space is 5. The Q table size of each agent is $|G|^{|C|} \cdot 5$; $|C|$ could reach thousands, so the state space and Q table will be very large and the computational complexity will be very high. In practice, it will take a long time to calculate the location of each taxi. In order to reduce the computational complexity, we need to make the state space smaller. We can achieve this by reducing the size of $|G|$ and $|C|$.

(i) Reduce the size of $|G|$: we can divide 81 grids into 3×3 large grids, each of which is also composed of 3×3 small grids. In this way, the state space is reduced to 1/9 of the original. After large grids have been adjusted and balanced, small grids will be scheduled.

(ii) Reduce the size of $|C|$: we can divide the matrix into two matrices of the same size by dividing the number of taxis in each grid equally, and the same effect can be achieved by balancing each submatrix. The number of agents in matrix can be reduced by half, and the resulting submatrices can be calculated in parallel, which improves the calculation speed further.

The pseudocodes of the algorithms used in this paper are shown in Algorithms 1 and 2.

Different scheduling algorithms have different goals, such as maximizing the drivers' profit, letting drivers find the next passenger faster, or minimizing the waiting time for passengers. The goal of this paper is to improve the utilization rate of taxis and to meet as many demands as possible with a certain number of available taxis. At the same time, the efficiency of the scheduling algorithm is also considered, which means using less taxis to meet more demands.

Therefore, the objective function of the scheduling model is defined as follows:

$$\mathrm{TAR} = r_{\mathrm{ds}} \cdot r_{\mathrm{tu}} \cdot e_{\mathrm{td}}, \tag{1}$$

$$\begin{aligned} r_{\mathrm{ds}} &= \frac{\text{demand\_satisfied}}{\text{total\_demand}}, \\ r_{\mathrm{tu}} &= \frac{\text{taxi\_utilized}}{\min(\text{total\_demand}, \text{total\_taxi})}, \\ e_{\mathrm{td}} &= \frac{\text{demand\_satisfied}}{\text{taxi\_dispatched}}. \end{aligned} \tag{2}$$

In equation (1), r_{ds} represents the demand satisfaction rate, which is calculated by dividing the total pickup requests by the satisfied demand as shown in equation (2). A good dispatching algorithm should satisfy as many demands as possible, so the higher the demand satisfaction rate is, the better the scheduling result will be. r_{tu} denotes the utilization rate of taxis. The calculation method, as shown in equation (2), equals the number of taxis that are effectively utilized (meaning that the taxi is dispatched and meets a certain demand) divided by the smaller value between the total pickup requests and the total number of taxis. There may be two situations; one shows that the number of taxis is less than the demand, in which case all taxis can be effectively utilized; the other is that the number of taxis is more than the demand, in which case taxis that can be effectively utilized are equal to the total demand at most. Sometimes, after the completion of the scheduling, some demands are not satisfied, but there are still some available taxis, which indicates that the scheduling algorithm is not good, so we hope that the value of r_{tu} is larger. e_{td} represents the efficiency of taxi dispatching. As shown in equation (2), the calculation method is equal to the demands satisfied divided by the number of taxis dispatched, which means how many demands are satisfied by each taxi. The larger the value of e_{td} is, the higher the efficiency of the dispatching algorithm is. Our goal is to adjust the proposed model to maximize the value of the objective function.

6. Experiment

In this section, we first compare the performance of three time series forecasting models under different indicators and then use the best performing model to provide data support for the subsequent scheduling. Then we compare the scheduling method proposed in this paper with another method in many aspects to test the effectiveness of our model.

6.1. Prediction Experiment. In order to have a precise prediction result for different time periods in the future, we divided a day into M time segments, each of which is t-hour length. For different types of cities or different regions of the same city, the change rate of traffic conditions is different, so for prosperous areas we should use a smaller t to respond to rapidly changing demand situations. For remote areas or small cities, traffic conditions are relatively stable; we can set t longer, which can reduce the frequency of calculation and ensure the accuracy of prediction.

In this section, three algorithms ARIMA, LSTM, and FBprophet are evaluated to predict demands. Two indicators, RMSE and MAE, are used to compare the performance of the three methods.

(1) RMSE (root mean square error): it is used to measure the deviation between the predicted values and the true values. It focuses on items with large difference between predicted and real values, and the smaller the value is, the better the algorithm will be. It can be defined as follows:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}}. \tag{3}$$

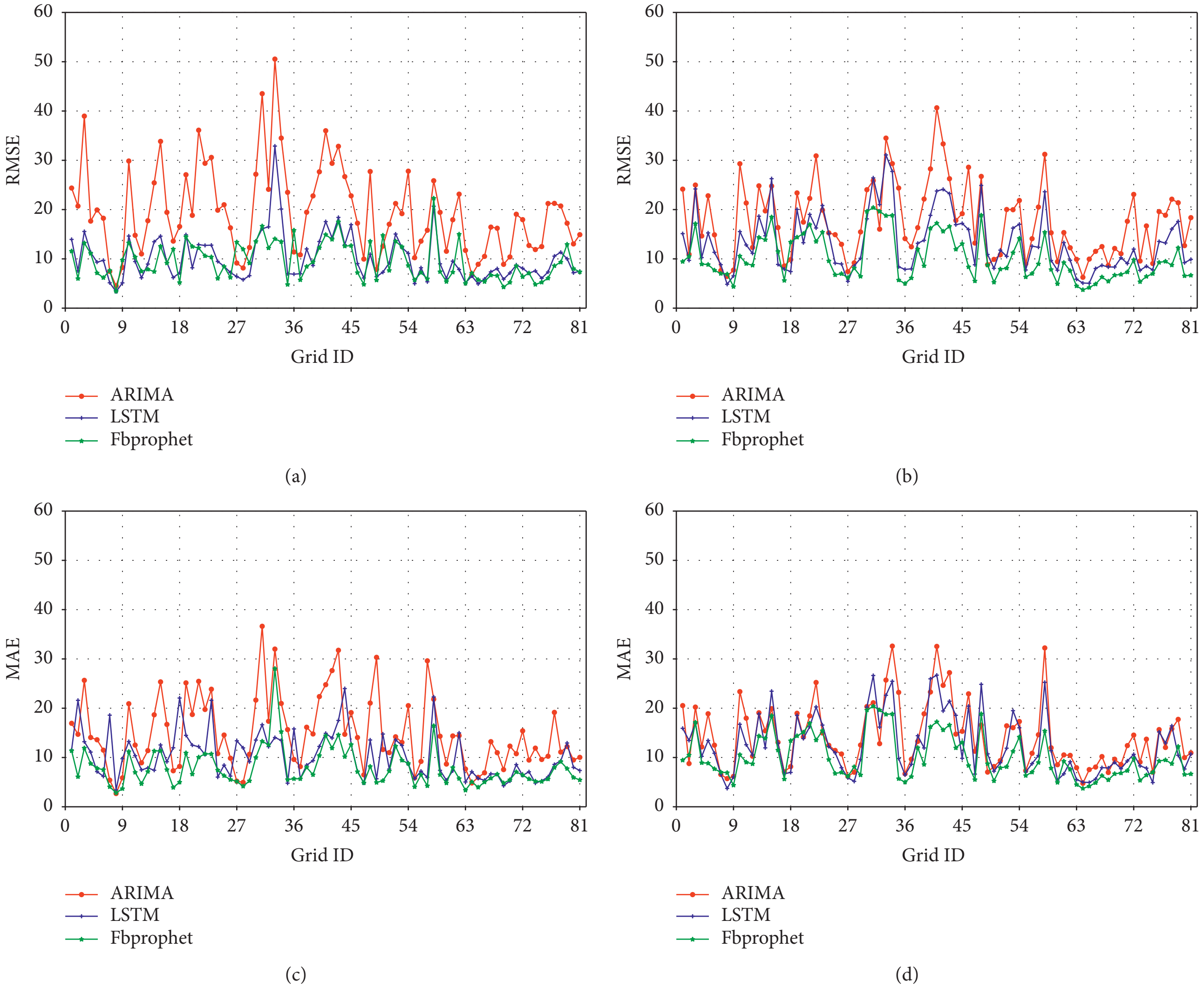
The predicted value and the real values are denoted by $\hat{y}_i$ and y_i, respectively, and the number of measurements is defined as n.

(2) MAE (mean absolute error): it represents the average absolute error between the predicted and observed values. It focuses on the sum of all the differences between predicted and real values. It can be defined as follows:

$$\mathrm{MAE} = \frac{1}{n} \sum_{i=1}^{n} \left| \hat{y}_i - y_i \right|. \tag{4}$$

As shown in Figure 5, FBprophet has the best performance under the two metrics whether it is tested under the condition of weekdays or weekends. This method does not need to adjust parameters. It has good generality to data and the prediction speed is very fast. And it is insensitive to the size of the data; even when forecasting on weekends with less data the accuracy is still high. LSTM's forecast results of working days are similar to FBprophet. It performs worse than FBprophet on weekends, but better than ARIMA. Its disadvantage is that it depends on the quality of network structure design and the setting of various parameters, and the training process of the network will consume a long time. ARIMA, a traditional model, does not perform well in this prediction problem, probably because there are many factors affecting the daily traffic conditions, and the model cannot predict these fluctuations very well. Moreover, this model needs to adjust different autoregressive coefficients p and moving average terms q for different data sets, which is high time cost, so it is not suitable for the prediction of multiple time series. In summary, we decide to use the FBprophet model for forecasting, because faster and higher accurate forecasting can make the scheduling results better.

6.2. Dispatch Experiment. By using the FBprophet model, we can get the taxi demands in each grid in the future. Then we can use the model proposed in this paper to schedule all available taxis in the range. In order to validate the



(a)

(b)

(c)

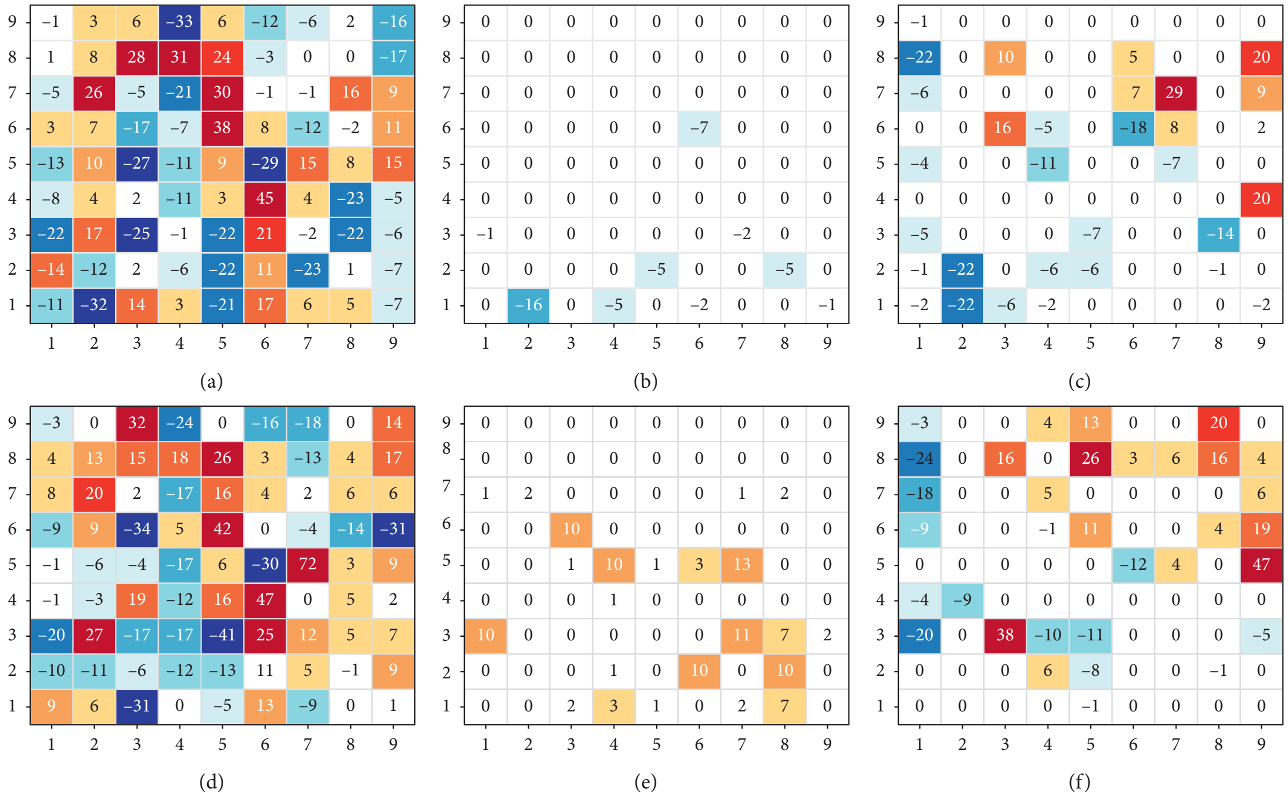
(d)

Figure 5: Comparison of the results of three methods under different indicators in different time periods. (a) RMSE-weekdays. (b) RMSE-weekends. (c) MAE-weekdays. (d) MAE-weekends.

performance of our model, we conducted experiments on different periods of weekdays and weekends and compared it with time-location-relationship (TLR) combined taxi service recommendation model proposed in [24]. The main idea of TLR model is that when a taxi driver needs to find passengers, the model compares the demands in eight grids around the taxi and then recommends the grid of the greatest taxi demands for the taxi as its destination. This scheduling method can easily result in taxis clustering in one area. In this paper, a small improvement is made in the process of implementation. This model will recommend a grid for the taxi, which is selected by the certain possibility from two grids with the most taxi demands. The experimental results are as follows.

In Figure 6, the deeper the red in this grid is, the more available the vehicles there will be, the deeper the blue in this grid is, the more the demands there will be, and the number in the grid represents the specific value. In the scenario shown in Figure 6(a), demands are 44 more than the number of available taxis, and the unsatisfied demands are 527 at the beginning. After our model scheduling, there are 44 unsatisfied demands, the demand satisfaction rate is 91.65%, and the taxi utilization rate is 100%. After TLR model scheduling, the unsatisfied demands are 160, the satisfaction rate is 69.64%, and the taxi utilization rate is 75.9%. In the scenario shown in Figure 6(d), demand is 114 less than the number of available vehicles, and the unsatisfied demand is 450 at the beginning. After our model scheduling, all the demands have been satisfied and the satisfaction rate is 100%. After TLR model scheduling, there are still 106 unsatisfied demands and the satisfaction rate is 76.44%.

According to Figure 6, we can see that the model proposed in this paper performs better in all time periods. During the peak period, 9 a.m. on weekdays, as shown in Figure 6(a), the imbalance between supply and demand is serious, and the number of available taxis is less than the demands. In this case, after our model scheduling, as shown in Figure 6(b), all available vehicles are utilized; in other words, no more taxis can be scheduled to meet the demand; and, for the TLR model, as shown in Figure 6(c), most of the



9	−1	3	6	−33	6	−12	−6	2	−16
8	1	8	28	31	24	−3	0	0	−17
7	−5	26	−5	−21	30	−1	−1	16	9
6	3	7	−17	−7	38	8	−12	−2	11
5	−13	10	−27	−11	9	−29	15	8	15
4	−8	4	2	−11	3	45	4	−23	−5
3	−22	17	−25	−1	−22	21	−2	−22	−6
2	−14	−12	2	−6	−22	11	−23	1	−7
1	−11	−32	14	3	−21	17	6	5	−7
	1	2	3	4	5	6	7	8	9

(a)

9	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	−7	0	0	0
5	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
3	−1	0	0	0	0	0	−2	0	0
2	0	0	0	0	−5	0	0	−5	0
1	0	−16	0	−5	0	−2	0	0	−1
	1	2	3	4	5	6	7	8	9

(b)

9	−1	0	0	0	0	0	0	0	0
8	−22	0	10	0	0	5	0	0	20
7	−6	0	0	0	0	7	29	0	9
6	0	0	16	−5	0	−18	8	0	2
5	−4	0	0	−11	0	0	−7	0	0
4	0	0	0	0	0	0	0	0	20
3	−5	0	0	0	−7	0	0	−14	0
2	−1	−22	0	−6	−6	0	0	−1	0
1	−2	−22	−6	−2	0	0	0	0	−2
	1	2	3	4	5	6	7	8	9

(c)

9	−3	0	32	−24	0	−16	−18	0	14
8	4	13	15	18	26	3	−13	4	17
7	8	20	2	−17	16	4	2	6	6
6	−9	9	−34	5	42	0	−4	−14	−31
5	−1	−6	−4	−17	6	−30	72	3	9
4	−1	−3	19	−12	16	47	0	5	2
3	−20	27	−17	−17	−41	25	12	5	7
2	−10	−11	−6	−12	−13	11	5	−1	9
1	9	6	−31	0	−5	13	−9	0	1
	1	2	3	4	5	6	7	8	9

(d)

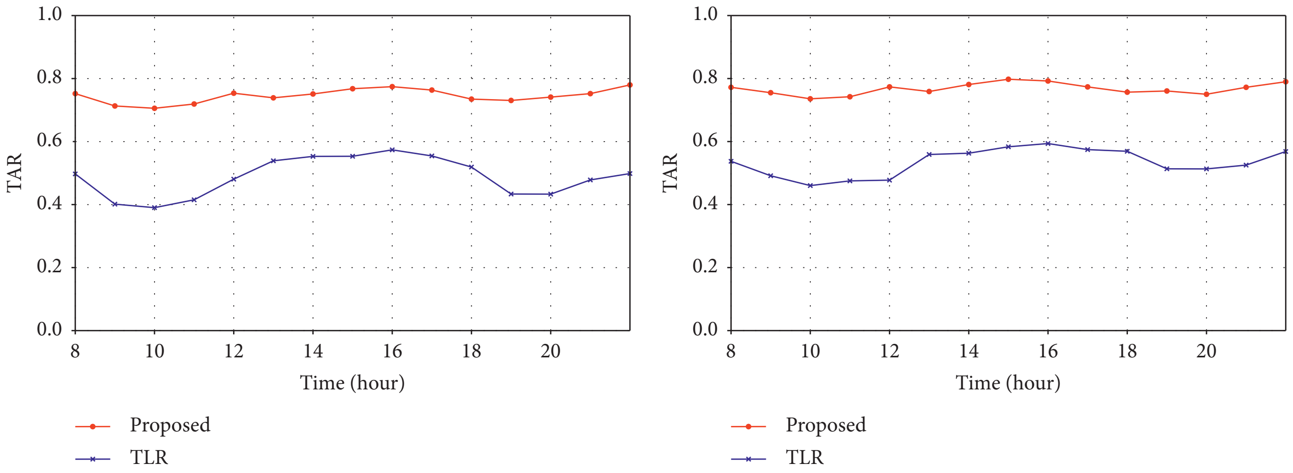
9	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
7	1	2	0	0	0	0	1	2	0
6	0	0	10	0	0	0	0	0	0
5	0	0	1	10	1	3	13	0	0
4	0	0	0	1	0	0	0	0	0
3	10	0	0	0	0	0	11	7	2
2	0	0	0	1	0	10	0	10	0
1	0	0	2	3	1	0	2	7	0
	1	2	3	4	5	6	7	8	9

(e)

9	−3	0	0	4	13	0	0	20	0
8	−24	0	16	0	26	3	6	16	4
7	−18	0	0	5	0	0	0	0	6
6	−9	0	0	−1	11	0	0	4	19
5	0	0	0	0	0	−12	4	0	47
4	−4	−9	0	0	0	0	0	0	0
3	−20	0	38	−10	−11	0	0	0	−5
2	0	0	0	6	−8	0	0	−1	0
1	0	0	0	0	−1	0	0	0	0
	1	2	3	4	5	6	7	8	9

(f)

Figure 6: Dispatch result of the two models. (a) Unscheduled status at 9:00 on weekdays. (b) The result of (a) after the scheduling of model proposed in this paper. (c) The results of (a) after the scheduling of TLR model. (d) Unscheduled status at 21:00 on weekends. (e) The results of (d) after the scheduling of model proposed in this paper. (f) The results of (d) after the scheduling of TLR model.



(a)

(b)

Figure 7: Model performance comparison. (a) The performance on weekdays. (b) The performance on weekends.

requirements are met, but there are still many available taxis leaving unused. At 9 p.m. on weekends, as shown in Figure 6(d), the degree of imbalance is relatively light, and the total number of available taxis is larger than the demands. In this case, after our model scheduling, as shown in Figure 6(e), all the demands are satisfied, and the remaining taxis are evenly distributed. However, the TLR model, as shown in Figure 6(f), cannot satisfy all the demands even when the number of taxis is more than the demands. In addition, it can be seen from Figures 6(c) and 6(f) that the hot zone and the cold zone are separated after the dispatching of the TLR model, which shows that if the cold zone and the hot zone are far away, the taxis in the hot zone cannot be used. This indicates that the contrast model is a local optimization model, and our model is a global optimization model, which can achieve the balance of supply and demand in the global scope.

Figure 7 shows the comparison result of two scheduling models under objective function 1 on weekdays and weekends. The experiment compares the scheduling results of the two models from 8 a.m. to 10 p.m. using the one-month data. It is clear from the graph that the proposed model is better than the comparative model as a whole. And the proposed model is more stable than the comparative model; the results of the comparative model are worse during the morning and evening peak periods than other periods; the reason is that the strategy of adjacent grid scheduling used by the comparative model cannot make full use of taxi resources, especially when many grids need taxis. Compared with weekdays, the objective function values of both methods become higher, and the gap between the two methods becomes smaller at weekends. The reason is that the spatial-temporal distribution of the demand becomes more uniform on weekends and the rush hour in the morning and evening is weaker. According to the above, the model proposed in this paper can make more efficient use of taxi resources and meet the needs of passengers better.

The experiment is carried out on an 8-core machine with an 8G RAM. The number of times in which a target matrix is split varies with the number of agents. But the splitting operation is very fast; the total splitting time does not exceed 0.01 s. Hence the running time is mainly determined by the speed of reinforcement learning algorithm. Reinforcement learning algorithms need some time to explore the optimal strategy. We repeat the experiment 100 times and the average running time of the program is 13.88s.

7. Conclusion

In this paper, we have proposed a MARL-based taxi predispatching model to balance the supply and demand of taxis in different areas of the city. Through the analysis of the historical data, we find that different functional regions have different crowd mobility patterns, and they all have regularity. Then, in order to react to the taxi demand situation in advance, we use three time series forecasting methods to predict the taxi pickup requests of each grid in the future and compare the results of them. Finally, according to the distribution of taxis at the current time, the scheduling model based on the multiagent reinforcement learning is used to dispatch taxis among grids. To reduce the computational complexity of the algorithm, we adopt the divide-and-conquer strategy, dividing the general tasks into subtasks that can be processed by a single machine, and each small task can be paralleled. The final scheduling method is obtained by summing up the results of all subtasks, which greatly improves the computational speed and the real-time performance of taxi scheduling.

In the experimental part, we first compare the prediction results of the three prediction models. The results show that the FBprophet model performs best under the two evaluation metrics, so we finally use the prediction results of FBprophet to approximate the real demand situation in the future. Then we compare the proposed scheduling algorithm with the TLR combined service recommendation method. We can see from the results that the proposed dispatching algorithm has better performance in various scenarios, and the performance is stable under different traffic conditions.

In the future, we will further carry out more fine-grained scheduling; specifically we will study which taxi should be dispatched in each grid, how to choose route for each taxi, and where to find passengers after reaching the designated grid. We will try to solve these problems and further improve the efficiency of taxi service.

Data Availability

The raw data used to support this study have not been made available because of privacy issue.

Conflicts of Interest

The authors claim that there are no conflicts of interest in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundations of China under Grant nos. 61772230 and 61972450, the Natural Science Foundation of China for Young Scholars (no. 61702215), China Postdoctoral Science Foundation (nos. 2017M611322 and 2018T110247), and Changchun Science and Technology Development Project (no. 18DY005).

References

[1] G. U. Sheng-Zu, “Theoretical considerations and strategic choice on the development of smart city,” *China Population Resources Environment*, vol. 208, pp. 94–97, 2012.

[2] P. Neirotti, A. De Marco, A. C. Cagliano, G. Mangano, and F. Scorrano, “Current trends in smart city initiatives: some stylised facts,” *Cities*, vol. 38, pp. 25–36, 2014.

[3] Z. Xiong, H. Sheng, W. G. Rong, and D. E. Cooper, “Intelligent transportation systems for smart cities: a progress review,” *Science China Information Sciences*, vol. 55, no. 12, pp. 2908–2914, 2012.

[4] Y. Hernafi, M. B. Ahmed, and M. Bouhorma, “An approaches’ based on intelligent transportation systems to dissect driver behavior and smart mobility in smart city,” in *Proceedings of the IEEE International Colloquium on Information Science Technology*, Tangier, Morocco, October 2016.

[5] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, “An information framework for creating a smart city through internet of things,” *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 112–121, 2014.

[6] J. Zawieska and J. Pieriegud, “Smart city as a tool for sustainable mobility and transport decarbonisation,” *Transport Policy*, vol. 63, pp. 39–50, 2018.

[7] R. Olszewski, P. Pałka, and A. Turek, “Solving smart city transport problems by designing carpooling gamification schemes with multi-agent systems: the case of the so-called mordor of Warsaw,” *Sensors*, vol. 18, no. 2, p. 141, 2018.

[8] E. Kourti, C. Christodoulou, L. Dimitriou, S. Christodoulou, and C. Antoniou, “Quantifying demand dynamics for

supporting optimal taxi services strategies," *Transportation Research Procedia*, vol. 22, pp. 675–684, 2017.

[9] S. Vhaduri, C. Poellabauer, A. Striegel, O. Lizardo, and D. Hachen, "Discovering places of interest using sensor data from smartphones and wearables," in *Proceedings of the 2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing*, San Francisco, CA, USA, August 2017.

[10] T. M. T. Do and D. Gatica-Perez, "The places of our lives: visiting patterns and automatic labeling from longitudinal smartphone data," *IEEE Transactions on Mobile Computing*, vol. 13, no. 3, pp. 638–648, 2014.

[11] W. Dong, L. Qian, X. Zhu, C. Jie, Y. Huang, and W. Chen, "Understanding travel behavior of private cars via trajectory big data analysis in urban environments," in *Proceedings of the 2017 IEEE 15th International Conference on Dependable, Autonomic and Secure Computing, 15th International Conference on Pervasive Intelligence and Computing, 3rd International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress*, Orlando, FL, USA, November 2017.

[12] D. Wang, J. Fan, Z. Xiao et al., "Stop-and-wait: discover aggregation effect based on private car trajectory data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 1–11.

[13] T. Kyaw, N. N. Oo, and W. Zaw, "Building travel speed estimation model for yangon city from public transport trajectory data," in *Proceedings of the International Conference on Big Data Analysis and Deep Learning Applications*, Miyazaki, Japan, May 2018.

[14] B. Li, D. Zhang, L. Sun et al., "Hunting or waiting? discovering passenger-finding strategies from a large-scale real-world taxi dataset," in *Proceedings of the IEEE International Conference on Pervasive Computing Communications Workshops*, Syracuse, NY, USA, June 2011.

[15] Y. Yu, Z. He, Z. Song, F. Xin, and C. Wang, "Investigation on structural and spatial characteristics of taxi trip trajectory network in Xi'an, China," *Physica A: Statistical Mechanics and Its Applications*, vol. 506, pp. 755–766, 2018.

[16] L. Dimitriou, E. Kourti, C. Christodoulou, and V. Gkania, "Dynamic estimation of optimal dispatching locations for taxi services in mega-cities based on detailed GPS information," *IFAC-PapersOnLine*, vol. 49, no. 3, pp. 197–202, 2016.

[17] D. Wei, C. Yuan, H. Liu, D. Wu, and W. Kumfer, "The impact of service refusal to the supply demand equilibrium in the taxicab market," *Networks and Spatial Economics*, vol. 17, no. 1, pp. 225–253, 2017.

[18] H. Rong, Z. Wang, Z. Hui et al., "Mining efficient taxi operation strategies from large scale geo-location data," *IEEE Access*, vol. 5, pp. 25623–25634, 2017.

[19] C. Kang and K. Qin, "Understanding operation behaviors of taxicabs in cities by matrix factorization," *Computers, Environment and Urban Systems*, vol. 60, pp. 79–88, 2016.

[20] D. Zhang, T. He, S. Lin et al., "Online cruising mile reduction for large-scale taxicab networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 11, pp. 3122–3135, 2015.

[21] D. Zhang, L. Sun, B. Li et al., "Understanding taxi service strategies from taxi gps traces," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 123–135, 2015.

[22] D. Zhang, H. Tian, L. Shan, S. Munir, and J. A. Stankovic, "Dmodel: online taxicab demand model from big sensor data in a roving sensor network," in *Proceedings of the IEEE International Congress on Big Data*, Anchorage, AK, USA, June 2014.

[23] Z. Chen, J. Weng, L. Sui, and Z. Yuan, "Study on the index system of taxi operation monitoring based on multi-source data," in *Proceeding of 2015 International Conference on Computer Science and Intelligent Communication, Atlantis Press*, 2015.

[24] X. Kong, X. Feng, J. Wang, A. Rahim, and S. K. Das, "Time-location-relationship combined service recommendation based on taxi trajectory data," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1202–1212, 2017.

[25] S. Phithakkitnukoon, M. Veloso, C. Bento, A. Biderman, and C. Ratti, "Taxi-aware map: identifying and predicting vacant taxis in the city," in *Proceedings of the First International Joint Conference on Ambient Intelligence (AmI'10)*, Malaga, Spain, November 2010.

[26] L. I. Xiaolong, G. Pan, W. U. Zhaohui et al., "Prediction of urban human mobility using large-scale taxi traces and its applications," *Frontiers of Computer Science in China*, vol. 6, pp. 111–121, 2012.

[27] N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie, "T-finder: a recommender system for finding passengers and vacant taxis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 10, pp. 2390–2403, 2013.

[28] F. Golpayegani and S. Clarke, "Co-ride: collaborative preference-based taxi-sharing and taxi-dispatch," in *Proceedings of the 2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, Volos, Greece, November 2018.

[29] M. Küemmel, F. Busch, and D. Z. Wang, "Taxi dispatching and stable marriage," *Procedia Computer Science*, vol. 83, pp. 163–170, 2016.

[30] H. Zheng and W. Jie, "Online to offline business: urban taxi dispatching with passenger-driver matching stability," in *Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, Atlanta, GA, USA, June 2017.

[31] K. T. Seow, N. H. Dang, and D. H. Lee, "A collaborative multiagent taxi-dispatch system," *IEEE Transactions on Automation Science Engineering*, vol. 7, no. 3, pp. 607–616, 2010.

[32] M. Guériau and I. Dusparic, "SAMoD: shared autonomous mobility-on-demand using decentralized reinforcement learning," in *Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1558–1563, Maui, HI, USA, November 2018.

[33] M. Li, Zhiwei Qin, Y. Jiao et al., "Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning," in *Proceedings of the 2019 World Wide Web Conference*, San Francisco, CA, USA, 2019.

[34] J. Jin, M. Zhou, W. Zhang et al., "Coride: joint order dispatching and fleet management for multi-scale ride-hailing platforms," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, Beijing, China, November 2019.

[35] J. Yuan, Y. Zheng, and X. Xie, "Discovering regions of different functions in a city using human mobility and pois," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Beijing, China, August 2012.

[36] L. Buoniu, R. Babuka, and B. D. Schutter, *Multi-Agent Reinforcement Learning: An Overview*, MDPI, Basel, Switzerland, 2010.

[37] H. M. Schwartz, *Multi-Agent Machine Learning: A Reinforcement Approach*, Wiley, Hoboken, NJ, USA, 2014.