Vehicle Dispatching Technology Based on Geographic Grid Division

Zhengyi Lei¹,² and Wenjing Jiang¹

¹School of Cybersecurity, Chengdu University of Information Technology, 24 Xuefu Rd, Shuangliu Qu, Chengdu 610225, China.
Email: leizhengyi0927@163.com

Abstract. With the increasing popularity of online ride platforms, more and more historical data can be used in the field of intelligent transportation. However, the taxi behavior of users is difficult to predict and has a great relationship with time and geographical location, which makes the supply and demand of vehicles show an imbalance. Reasonable vehicle scheduling can not only increase driver's income, improve service quality, but also make effective use of traffic resources. In this paper, a grid-based scheduling model is proposed. By gridding maps, the computational complexity of Donger is reduced to output vehicle scheduling actions quickly. The problem of vehicle supply and demand imbalance can be effectively solved by correcting model parameters based on return loss function. In this paper, the model is optimized from three aspects: loss function, learning rate and grid scale. The goal is to maximize the sum of orders. Compared with the traditional machine learning model, the model proposed in this paper can have deeper features and thus obtain higher prediction accuracy. The experimental results on a large number of real historical taxi order datasets show that the proposed model has high prediction accuracy.

1. Introduction
DiDi and Uber online ride sharing platform has gradually changed people's travel mode. By using mobile network and GPS to realize scheduling, we can effectively use the idle vehicles without order service in the short term, so as to meet more order requirements. One of the main challenges of the mission is to address the imbalance between supply and demand of vehicles. In big cities, millions of orders are processed every day, but there are too few vehicles available in some areas to serve current orders quickly. On the other hand, some areas require a small number of passengers, resulting in vehicles can’t be used effectively. If we can send vehicles in advance to areas with large orders, more orders can be served at the same time. despite the large amount of historical data, managing network car-sharing in an efficient manner remains a challenge as vehicle scheduling will affect future vehicle distribution. traditional machine learning methods are difficult to capture and model real-time dynamic changes. With the development of deep learning, deep reinforcement learning has achieved great success in modeling difficult decision-making problems in the past. This paper presents a method to optimize vehicle scheduling by grid division. Our work is based on the following aspects:

- Feasibility of model setting. Rational modeling of service areas, drivers, and orders can be optimized for interaction between platforms, passengers, and drivers. In this paper, the vehicle in the unified grid is modeled as a unified class target, and its scheduling is unified.
- Setting of grid scale. By gridding the service area, the computational complexity is reduced to output the speed of the scheduling action. Grid division of different sizes has a great impact on the performance of the model, and a reasonable grid scale can effectively carry out vehicle scheduling.

In order to overcome the shortcomings of the existing vehicle demand prediction methods, we
propose an order prediction model based on grid division. The main contributions include: (1) analyzing the influence of geographical location on orders and extracting key features from them; (2) dividing service maps into multiple hexagonal thus reducing computational complexity; (3) optimizing parameter update operations from three aspects: loss function, grid size and learning rate.

2. Related Work

Intelligent Transportation System. Advances in artificial intelligence and traffic data mining make it possible to solve more challenging traffic problems. More and more researches apply reinforcement learning algorithms to complex traffic management problems. And reinforcement learning also shows superior performance in this field. One trend is to mine the rules of people daily activities in massive traffic trajectory data (GPS) [1]. GPS datasets in big data systems provide rich contextual information that can efficiently implement advanced functions such as navigation, tracking and security in urban computing systems. In [2], GPS is used to predict road congestion so that the driver can avoid the congested road in advance [3], which reduces the congestion of the road segment and allows the driver to reach the destination in a shorter time [4]. In addition, Wang [5] propose a coupled matrix and tensor factorization model named TCE R, which combines GPS detection data and social media data to predict urban traffic congestion, and to more accurately complete the sparse traffic congestion matrix by collaboratively factorizing it with other matrices and tensors formed by other data. In practical applications, there are performance gaps between different vehicle-mounted GPS devices, and some GPS trajectories are sparse and incomplete. This makes the application of related algorithms more complicated.

3. Problem Formulation

The main problem solved in this paper is to balance the supply and demand difference of vehicles through scheduling management, by increasing available vehicles in high demand areas. Considering the delay of vehicle scheduling, the day is divided into 144 time periods (10 minutes), and the vehicle performs one more scheduling action in one cycle to ensure that the target vehicle can reach the target area in the next time period. The scheduling action is constrained to only schedule the target vehicle to adjacent areas.

This paper formulates the problem based on reinforcement learning [6]. It is defined by the tuple \( G = (N; S; A; P; R; r) \), where \( N \); \( S \); \( A \); \( P \); \( R \); \( r \) are the number of agents, state set, joint action space, transition probability function, Reward function and discount factor respectively. They are defined as follows:

- **Agent** [7]: Modeling a vehicle as an agent is convenient for model scheduling, and vehicles locating in the same grid are treated as the same kind of agent to reduce computational complexity.

- **State**: It include grid id, time, current available vehicles and orders of grids, and future available vehicles and orders of grids, the state will directly determine the output of the action, the execution of the action will also change the state of the grid.

- **Action**: The action is output by the strategy according to the current state of the agent. An action can only dispatch the agent to the adjacent grids or keep it stay in the current grid. Based on the hexagonal grid, the action is defined as \( a_t^i \), \( t \) is the time to execute the action, \( i \) is the id of the agent, and the value of \( a_t^i \) is an integer from 0 to 6. 0 means that the agent \( i \) stays in the current grid at time \( t \), and 1 to 6 means that the agent \( i \) goes to the corresponding adjacent grid at time \( t \). As shown below. The action set of all agents at time \( t \) is represented as \( a_t = \{a_t^i\}^N \), and \( N \) is the total number of agents at time \( t \).

- **Reward function**: Every agent has a reward function, and agents with the same state have the same reward function. The reward for the agent \( i \) to perform the action \( a_t^i \) is defined as the average return of all agents that reach the target grid at time \( t+1 \), so as to avoid dispatching too many vehicles to areas with higher vehicles demand, thus maximizing GMV does not make the return of a single agent too small.
4. Feature Analysis
As shown in figure 1. There are significant differences in the order's form one grid area to another. Daily activities in different areas also affect the demand for vehicles. The demand for vehicles is also higher in areas with high human activity. Different sizes of grid divisions also change the amount of orders within each grid.

![Figure 1. Heat-map for vehicle demand in each grid in Haikou.](image)

5. Order Forecasting Model
After the features are selected, they are constructed as a two-dimensional feature matrix and input to the prediction model, and historical data are used to evaluate the vehicle demand prediction model to verify its performance.

The grid partitioning-based order prediction forecasting scheduling model proposed in this paper contains the following working components, as shown in figure 2.

In this paper, we use reinforcement learning to build an order prediction model. The difference from traditional prediction models is that the objective is to maximize long-term revenue, and the model performs vehicle scheduling with the goal of maximizing the total order revenue for a day.

Q-learning, a commonly used off-policy algorithm, uses the greedy policy \( (s) = \text{argmax}_a Q(s, a) \). Many reinforcement learning includes action value functions, which represent the expected rewards that can be obtained by executing action \( a_t \) in state \( s_t \) according to the strategy \( \pi \).

\[
Q^\pi(s_t, a_t) = E_{\delta_t} [R_t | s_t, a_t]
\]

(1)

Through the recursive relationship of Bellman equation, the expected reward of an action can be expressed as:

\[
Q^\pi(s_t, a_t) = E_{\delta_t} [r(s_t, a_t) + \gamma E_{\delta_{t+1}} [Q^\pi(s_{t+1}, a_{t+1})]]
\]

(2)

We consider function approximators parameterized by \( \theta \), which we Optimize by minimizing the loss:

\[
L(\theta) = E_{s_t, a_t, r_t, s_{t+1}, \mu_t} [Q(s_t, a_t | \theta) - y_t]^2
\]

(3)

\[
y_t = r(s_t, a_t) + \gamma Q(s_{t+1}, \mu(s_{t+1}) | \theta)
\]

(4)

Where \( \beta \) is policy generated by the historical data for training network, \( \rho^\beta \) is discounted state visitation distribution for a policy \( \beta \).

For the problem of continuous action space, it is impossible to directly use Q-learning to calculate the value of each action. Actor-Critic is a method that combines Q-learning and policy gradients. The continuous action output that Q-learning is not good at is given to Actor composed of policy gradients, and Q-learning is used as a Critic to evaluate the output actions of Actor.

Using the action value function to represent the gradient of the state value function \( \nu_a(s) \):

\[
\n\]
\[ \nabla V^\pi(s) = \sum_{x} \sum_{k=0} P(s \rightarrow x, k, \pi) \sum_{a} \nabla \pi(a \mid x) Q^\pi(s, a) \]  

(5)

Where \((s \rightarrow x, k, \pi)\) is the probability of transition from state \(s\) to state \(x\) after \(k\) steps under the policy \(\pi\). The policy gradient is:

\[ \nabla J(\theta) = \sum_{s} \mu(s) \sum_{a} \nabla \pi(a \mid s) Q^\pi(s, a) \]  

(6)

Where \(\mu\) is the distribution under \(\pi\).

The critic \(Q(s; a)\) is learned using the Bellman equation as in Q-learning. Actor \(p(s \mid \theta^p)\) is learned by Equation (9) based on expected return.

\[ \nabla_{\theta^p} J = E_{s \sim \mu} \left[ \nabla_{a} Q(s, a \mid \theta^p) \sum_{s' \sim h_{ik}; a \rightarrow p(s')} \nabla_{\theta^p} p(s | \theta^p) \mid s = s_i \right] \]  

(7)

Figure 2. Working mechanism of the dispatching model.

6. Datasets

We use Data source in Didi Chuxing GAIA Initiative [8]. The data set is described in table 1.

| id | price | Longitude  | Latitude  | Arriving time |
|----|-------|------------|-----------|---------------|
| 1  | 22    | 110.3071   | 20.0032   | 19:20:00      |
| 1  | 24    | 110.3444   | 19.9833   | 19:32:40      |
| 3  | 12    | 110.3377   | 20.039    | 19:55:18      |

7. Experiments

We conducted experiments based on different indicators. Such as value of order(income of drivers one day), order response rate(the proportion of the number of orders served to the total number of orders).
7.1. Grid Scale
As shown in figure 3, We perform vehicle dispatching, order allocation and order servicing based on different grid scales and count the total value of all served orders in a day.

![Figure 3. Value of order for different grid scales.](image)

7.2. Costing of Dispatching
Vehicles perform dispatch actions that require additional fuel costs, which we count as part of the order revenue. A large number of mobilization actions reduces order revenue, thus constraining the model to output only efficiently dispatched actions. As shown in figure 4.

![Figure 4. Value of order for fee of dispatching.](image)

8. References
[1] Jiang B, Dayong T, Yuanyan T, and Dacheng T. 2019. Trajectory Data Classification: A Review. ACM Trans. Intell. Syst. Technol. 10, 4 (2019), 33:1–33:34.
[2] Hao Chen, Senzhang Wang, Zengde Deng, Xiaoming Zhang, and Zhoujun Li. 2019. FGST: Fine-Grained Spatial-Temporal Based Regression for Stationless Bike Traffic Prediction. In Advances in Knowledge Discovery and Data Mining - 23rd Pacific-Asia Conference, PAKDD 2019, Macau, China, April 14-17, 2019, Proceedings, Part I. 265–279.
[3] Bowen Du, Hao Peng, Senzhang Wang, Md. Zakirul Alam Bhuiyan, Lihong Wang, Qiran Gong, Lin Liu, and Jing Li. 2020. Deep Irregular Convolutional Residual LSTM for Urban Traffic Passenger Flows Prediction. IEEE Trans. Intell. Transp. Syst. 21, 3 (2020), 972–985.
[4] Jakob N. Foerster, Nantas Nardelli, Gregory Farquhar, Triantafyllos Afouras, Philip H. S. Torr, Pushmeet Kohli, and Shimon Whiteson. 2017. Stabilising Experience Replay for Deep Multi-Agent Reinforcement Learning. In Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. 1146–1155.
[5] He He and Jordan L. Boyd-Graber. 2016. Opponent Modeling in Deep Reinforcement Learning. In Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016. 1804–1813.

[6] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z. Leibo, David Silver, and Koray Kavukcuoglu. 2017. Reinforcement Learning with Unsupervised Auxiliary Tasks. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings.

[7] Mohammad Al-Zinati and Rym Wenkstern. 2019. Agent-Environment Interactions in Large-Scale MultiAgent Based Simulation Systems. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS ’19, Montreal, QC, Canada, May 13-17, 2019. 763–771.

[8] https://gaia.didichuxing.com